Implementing Embedded Streaming Media: 10 Secrets of Success

Jeff Bier
Berkeley Design Technology, Inc. (BDTI)
bier@BDTI.com
www.BDTI.com
© 2001 BDTI

OUTLINE

• Introduction
• Applications
• Formats and Standards for Streaming Audio and Video
• Hardware Considerations
• Software Considerations
• Additional Considerations
• Conclusions
INTRODUCTION

Challenges of Implementing Streaming Media Products

- Evolving technologies
  - Chips, communications standards
  - Compression formats, rights management
- Competitive market
  - Many players, big and small
  - Overlap between similar applications
- Many complex design considerations
  - Quality and feature selection
  - Cost and time-to-market constraints
Characteristics of (Strict) Streaming Media

- Media carried in packets
- Packets may arrive out of order
- Packets may not arrive at all!
- Network or some intermediary *not designed to carry data reliably in real-time*
- Starts playing before the entire audio/video clip is downloaded

Typical Big Picture (Expanded)
APPLICATIONS

Internet Applications

- Audio via Internet becoming ubiquitous
- Video via Internet gaining popularity
- Communications products incorporating streaming media
  - E.g., Nokia 9210 Communicator
- By 2003, 50% of Internet access may be via non-PC devices (CEMA)
- Streaming audio may displace traditional radio
Set-Top Boxes

- Devices providing interface to cable, other services
- New applications emerging
- Today, categories overlap:
  - Home theater functions
  - Direct broadcast satellite
  - Internet terminals
  - Digital recorder (TiVo)
  - Interactive TV
  - Music

Game Consoles

- Game consoles:
  - Stand-alone units
  - Display via TV set
  - Fast CPU
  - Graphics co-processors
  - Storage options
- Consoles & PCs require similar audio functions
- Consoles now support DVD playback
- Broadband communications ports will enable streaming media applications in future consoles
Other Consumer Applications

- DAB – Digital Audio Broadcast
- DBS – Direct Broadcast Satellite
- Digital camcorders
- Home theatre
- Home audio
- Car audio
- In-flight audio/video
- Kitchen appliances (!)

FORMATS AND STANDARDS FOR STREAMING AUDIO AND VIDEO
Secret for Success #1:

Select appropriate algorithm(s)

Selecting an Algorithm

- Compatibility with existing content
- Audio/video quality
- Bitrates supported
  - Match network/broadcast bandwidth?
- Resource requirements
  - CPU cycles, memory use
- Cost considerations
  - Licensing fees, royalties
  - Development effort
- May want to support multiple formats
Video Quality

- Display parameters
  - Frame resolution (pixels per frame)
  - Color resolution (# of possible colors)
  - Frame rate (frames per second)
- Visible compression artifacts
  - “Blocking” artifacts
  - Gibbs effect: blurring/shimmer around objects
  - “Ringing” artifacts
- Viewing tests are important

Audio Quality

- Speech quality
  - Is speech intelligible?
  - Can speaker be identified?
  - Is speech natural?
- Music / streaming media quality
  - “CD-quality”: 16 bits, 44.1 kHz
  - Misused term
- Listening tests are important
MPEG Family

- Moving Pictures Experts Group
- Moving pictures + associated audio
- MPEG-1, MPEG-2, MPEG-4
  - MPEG-2 is the most popular video compression technique today
- Ongoing standardization effort (MPEG-7)

MPEG-4 Media Objects

- SCENE
- 2-D BACKGROUND
- MAN
- STOP SIGN
- VAN
- BODY
- ANIMATE FACE
- SOUND
- IMAGE
- SPEECH

www.BDTI.com
Other Video Algorithms

- **Ligos Indeo**
  - Scalable bitstreams:
    - Low-quality preview
    - High-quality, high bandwidth version

- **RealNetworks RealVideo 8**
  - Bitrates from 20 kbps to 0.5 Mbps
  - Frame rate is adjusted according to content and target bandwidth

- **MJPEG, Cinepak**

Audio Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPEG-1</td>
<td>1992</td>
</tr>
<tr>
<td>MPEG-2</td>
<td>1994</td>
</tr>
<tr>
<td>AAC</td>
<td>1997</td>
</tr>
<tr>
<td>MPEG-4</td>
<td>1999</td>
</tr>
<tr>
<td>MPEG-7</td>
<td>2001</td>
</tr>
<tr>
<td>MPEG-21</td>
<td>20xx</td>
</tr>
<tr>
<td>ATRAC (Sony)</td>
<td>1992</td>
</tr>
<tr>
<td>PAC (Lucent)</td>
<td>1992</td>
</tr>
<tr>
<td>AC-3 (Dolby)</td>
<td>1995</td>
</tr>
<tr>
<td>TwinVQ (NTT)</td>
<td>1995</td>
</tr>
<tr>
<td>Coherent Acoustics (DTS)</td>
<td>1996</td>
</tr>
<tr>
<td>MLP (Meridian)</td>
<td>1997</td>
</tr>
<tr>
<td>G2 (RealNetworks)</td>
<td>1998</td>
</tr>
<tr>
<td>WMA (Microsoft)</td>
<td>1999</td>
</tr>
<tr>
<td>Qdesign</td>
<td>1999</td>
</tr>
</tbody>
</table>
Audio Algorithms

- MPEG-1/2 (MP3 = MPEG-1/2, Layer 3)
- MPEG-2 AAC (Advanced Audio Coding)
  - 8 - 96 kHz sample rate, up to 48 channels
- MPEG-4
  - Uses different compression methods for different types of audio signals
- RealNetworks RealAudio 8
  - Frames interleaved across several transmission packets
  - RealNetworks’ algorithms + ATRAC
- Sony ATRAC, Dolby AC-3, Microsoft WMA

Secret for Success #2:

In-depth understanding of algorithm(s) is necessary
HARDWARE
CONSIDERATIONS

Selecting a Processor
…Or Evaluating an Existing One

- Performance Considerations
- Cost Considerations
- Development Considerations
### Processor Categories

- **Custom ASICs**
- **ASSPs**
- **DSP Processors**
- **Media Processors**
- **Embedded RISC CPUs**
- **PC CPUs**

Arrows indicate the spectrum from **Fixed Function** to **Fully Programmable**

**Generality**

### Arithmetic Formats

<table>
<thead>
<tr>
<th></th>
<th>Fixed point</th>
<th>Floating point</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cost</strong></td>
<td>Cheap</td>
<td>Expensive</td>
</tr>
<tr>
<td><strong>Ease of use</strong></td>
<td>Tricky</td>
<td>Easy</td>
</tr>
<tr>
<td><strong>Dynamic range</strong></td>
<td>Same as precision</td>
<td>Set by exponent: 1500 dB for single-precision IEEE</td>
</tr>
<tr>
<td><strong>Precision</strong></td>
<td>16 bit: 1 part in 64K 24 bit: 1 part in 16 M</td>
<td>Equal to mantissa precision (24 bit for IEEE signal precision)</td>
</tr>
</tbody>
</table>
Secret for Success #3:

Match the processor to the algorithm(s)

Performance Considerations

- Architectural features
  - DSP arithmetic operations
  - Data bandwidth, DSP addressing modes
  - Cache size
  - Bit-field manipulation
  - Control operation efficiency
  - I/O efficiency (e.g., interrupt handling)
- Numeric fidelity
  - Data type(s)
  - Saturation, rounding, scaling, block floating-point
- Power consumption
Resource Requirements

- Video requirements depend on:
  - Image size(s) supported by application
  - Desired frame rate
  - Encoding practices

- Real-time MPEG-2 video decode:
  - Example stream: DVD
    - 720x480 pixels, 30 fps
  - On a VLIW media processor:
    - ~80% of a 166 MHz TriMedia TM32 core

- Memory requirements vary from 100s of kbytes to several Mbytes

---

Resource Requirements

- Real-time MP3 decode example:
  - On a 24-bit DSP:
    - ~20 MIPS on a Motorola DSP56307
    - ~56 Kbytes total program + data memory

- Real-time Real G2 decode example:
  - On an embedded CPU:
    - ~27 MIPS on an NEC VR5432
    - ~48 Kbytes total program + data memory
Resource Requirements

- Don’t forget other functions:
  - Player application
  - Sample rate conversion, color space conversion
  - Tone controls
  - Rights management, I/O, ...

Secret for Success #4:

Integration lowers costs and simplifies hardware design
Hardware Integration

- Chip cost vs. system cost
- Hardware system components
  - Memory
    - On-chip memory
    - Specialized external memory interfaces
  - I/O
    - Appropriate interfaces (e.g., I²S)
    - On-chip peripherals
- Off-the-shelf device or custom SoC?

SOFTWARE CONSIDERATIONS
Secret for Success #5:

No use reinventing the wheel: utilize available software modules
Operating Systems

- Provide real-time scheduling, task switching, inter-task communication, file system, (maybe) network stack
- Off-shelf candidates
  - Wind River VxWorks (set-top boxes)
  - Symbian EPOC (wireless)
  - Palm PalmOS (PDAs)
  - Microsoft WinCE (PDAs)
  - iObjects Dadio (portable players)
  - Embedded Linux (set-top boxes)

I/O Management Software

- Management of DAC, USB port, etc.
  - Interrupt service routines (ISRs)
  - DMA management
  - Buffering

- Network stack
  - IP, TCP, UDP, RTSP, RTP, ...

- Possible sources:
  - OS vendor
  - Processor vendor
  - Third parties
Player Software

- Responsible for
  - GUI
  - File management (if stored files available)
  - Play, stop, pause, fast-forward, rewind, …
  - Error detection, correction
- Makes calls to decoder, encoder
- Maintains synchronization of audio and video
- Communicates with network

Secret for Success #6:

Create a usable and complete software development environment
Development Considerations

- **Software**
  - Components, modules, applications

- **Architecture**
  - Complexity, data type(s)
  - Compatibility

- **Tools**
  - Compiler
    - Robustness, efficiency
  - Debugger, IDE, development boards, OS
  - Version control

- **Support**
  - From vendor, third parties, consultants

DSP Software Development

- **Not like other kinds of SW development. Why?**
  - Resource-hungry, complex algorithms
  - Severe cost limitations
  - Numeric fidelity
  - Hard real-time constraints
  - Time-to-market constraints

- **Optimization is essential**
Where to Start?

- Standard specifications
- Reference implementation
- Optimized implementation(s)
  - From algorithm vendor
  - From chip vendor
  - From third party developers
- Published papers
  - Often describe optimizations, pitfalls, etc.
- Independent software developers
  - May have valuable experience, expertise, and methodology

Secret for Success #7:

Watch out for outdated or erroneous code, specifications, and documentation
Some Pitfalls to Avoid

- Be wary of publicly available source code
  - May be outdated and/or lack features
  - Audio/video quality may be low
- Be wary of “reference” code
  - May be extremely inefficient
  - May be based on floating-point math
- Be wary of the published spec
  - May be outdated or incomplete
- Be sure to get all errata sheets and updates for spec (and for chip)

Secret for Success #8:

Focus optimization effort where it will be most effective
Software Optimization

- **Divide and conquer**
  - Profile of algorithm execution by function
  - Estimate optimization gain per function
  - Estimate optimization effort per function

- **Optimization techniques**
  - Algorithm transformation/modification
  - Processor-independent software optimization
  - Processor-specific optimization

Optimization Techniques

**Algorithm Transformations**

- Re-arrange block diagram
  - E.g., down-mix in frequency domain

- Coupling channel
  - E.g., re-calculate vs. store in memory

- Truncate where you can

- Recast or factor iMDCT

- Recast Huffman coding
  - Binary search tree?
  - ROM lookup tables?
Optimization Techniques
Processor-Independent Optimization

- **Strength reduction**
  - Avoid costly operations:
    
    ```
    int i, k, x[N];
    ...
    for (i=0; i<N; i++)
    x[i] /= k;
    ```
    
    ```
    int i, k, x[N], oneoverk;
    ...
    oneoverk = (1<<12)/k;
    for (i=0; i<N; i++)
    x[i] = (x[i] * oneoverk)>>12;
    ```

- **Function in-lining**
- **Recycle otherwise idle buffers**

Optimization Techniques
Processor-Specific Optimization

- **Code optimizations**
  - Loop unrolling
  - Change memory map
  - Use specialized instructions
    - ‘C54xx instruction to count 1s, 0s
    - Tricks with bit counter

- **Hardware optimizations**
  - Customize instructions
  - Accelerators and co-processors
Secret for Success #9:

Plan out the testing of the implementation in advance
Testing

- Presents technical challenges
  - Vast amounts of data
  - Development platform limitations
- Audio/video quality
  - Objective measures, subjective tests
  - Varies with type of content
- Modes
  - Sample rates, frame sizes, compression
- Real-time
  - Data-dependent execution time
  - Dynamic processor features

Secret for Success #10:

Give the software “room to grow”
Future-proofing

- Standards & algorithms are evolving
  - New algorithms tend to consume more CPU power and memory than older ones
- Security technology still under development
- Products may need to be field-upgradeable
  - Must support new software downloads
  - Must provide sufficient CPU power and memory for future algorithms

CONCLUSIONS
Conclusions

- Streaming media applications promise to revolutionize communication and entertainment
- Key technologies exist today
  - Broadband connections
  - Algorithms and protocols
  - Inexpensive microprocessors
  - Accessible content & server networks

Conclusions

- Streaming media product design and implementation are extremely challenging
  - Hardware challenges
    - Processor selection
    - Cost limitations
  - Software challenges
    - Demanding algorithms
    - Optimization
    - Testing
  - Audio/video quality requirements
  - Time-to-market
Resources

- BDTI
  - www.BDTI.com
  - *Digital Audio: Applications, Algorithms, and Implementation*
  - *Buyer's Guide to DSP Processors*
- MicroDesign Resources
  - www.MDRonline.com
  - *Microprocessor Report*
  - *Embedded Processor Watch*
- Forward Concepts
  - www.fwdconcepts.com
  - *The Convergence of Audio*
  - *Beyond MP3*