H.264 Streaming Client on DM64x

Arvind Raman, Kismat Singh, Manisha Agrawal Mohan,
Neelakanth Shigihalli, Sriram Sethuraman, B.S. Supreeth

Ittiam Systems (Pvt.) Ltd.,
Bangalore, India

Presented by: Sriram Sethuraman
TI DSP Fest, 3 December 2003, Bangalore

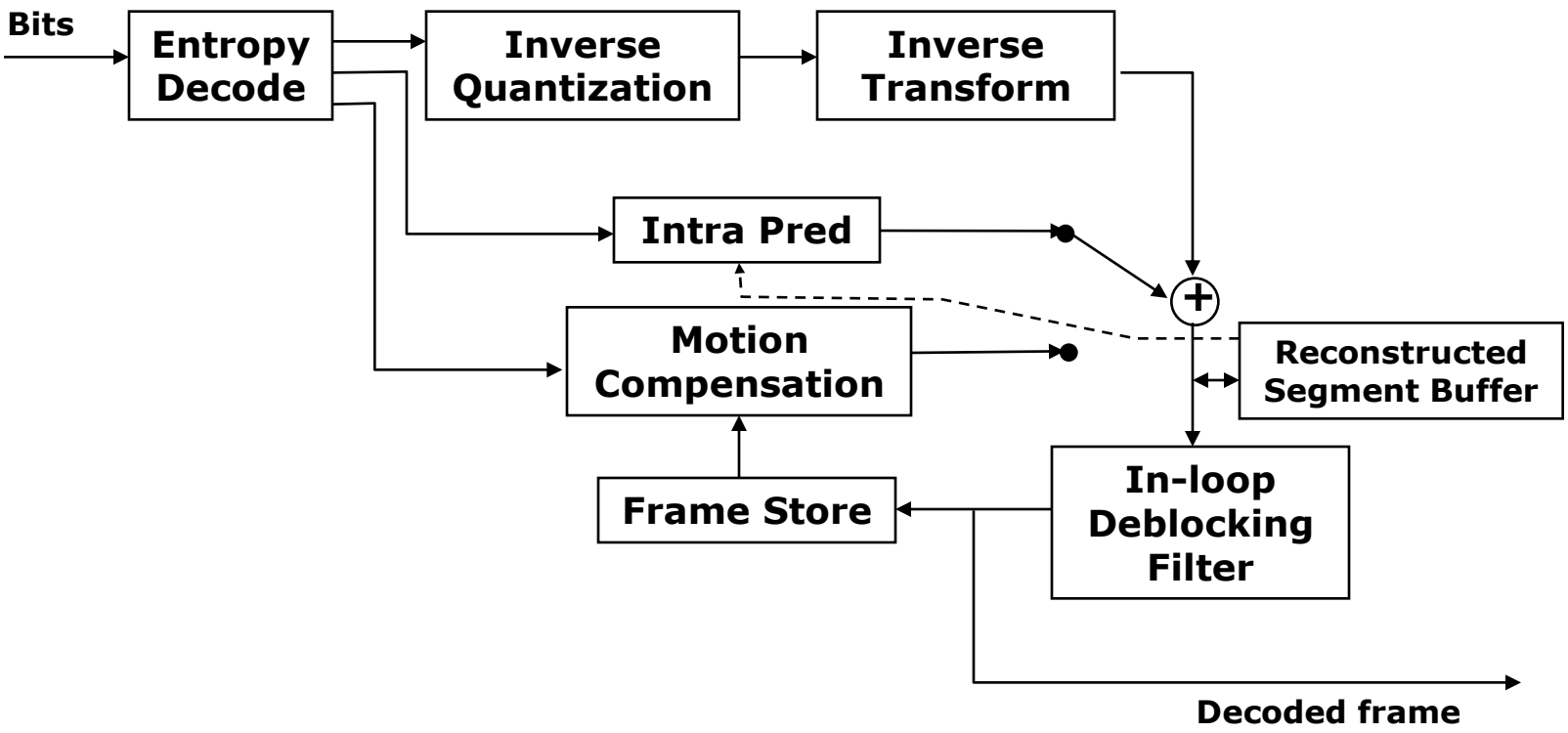
Outline of Presentation

- Overview of H.264
- Application Scenario
- TI DM642 – Capabilities/Features
- End-to-End Setup
- Task Scheduling
- Design Methodology
  - Low-level optimization Strategies
  - High-level multithreading between DSP and DMA
- Conclusions

H.264 - Introduction

- Started in 1998 as H.26L
  - Goal to achieve 50% reduction in bitrate
- Joint ITU-T and ISO/IEC standard
  - ISO/IEC 14496-10 (aka MPEG-4 Part 10)
- Status
  - Now officially balloted as an international standard
  - Via licensing and MPEG LA have formed patent pools
- Addresses diverse applications
  - Video telephony
  - Streaming
  - Storage (contender for the new DVD format)
  - Digital Cinema

H.264 Decoder



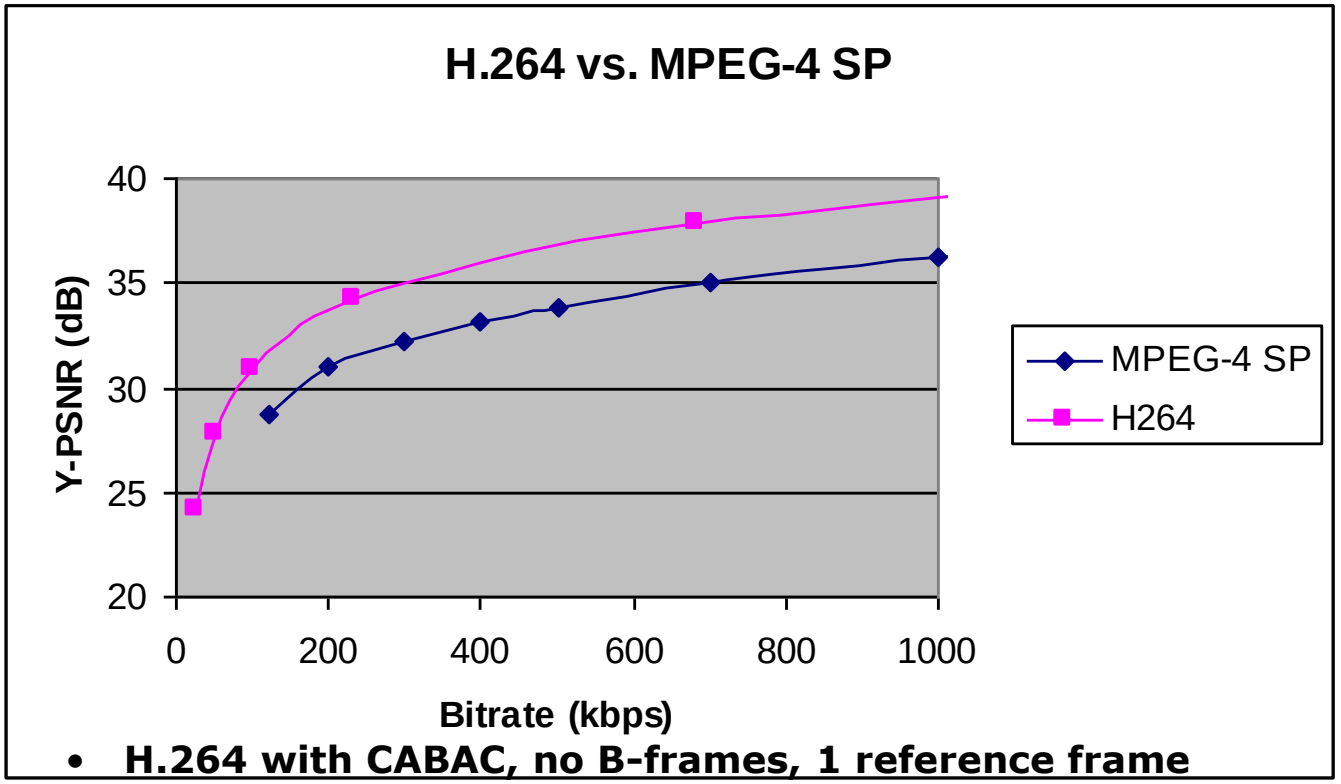
H.264 - Features

- 4x4 block sizes
- Improved intra prediction
  - Compression efficiency on par with JPEG-2000
- Enhanced temporal prediction
  - Improved quarter pixel motion compensation
  - Variable block size motion compensation
  - Multiple reference frames
  - Weighted prediction
- Efficient entropy coding
  - Context based VLC (CAVLC)
  - Context based binary arithmetic coding (CABAC)
- Interlaced support
  - Field pictures, MB level adaptive field/frame
- Integer transform
  - No mismatch between encoder and decoder
- Low complexity, wider range quantization
- Superior in-loop deblocking filter

Profiles

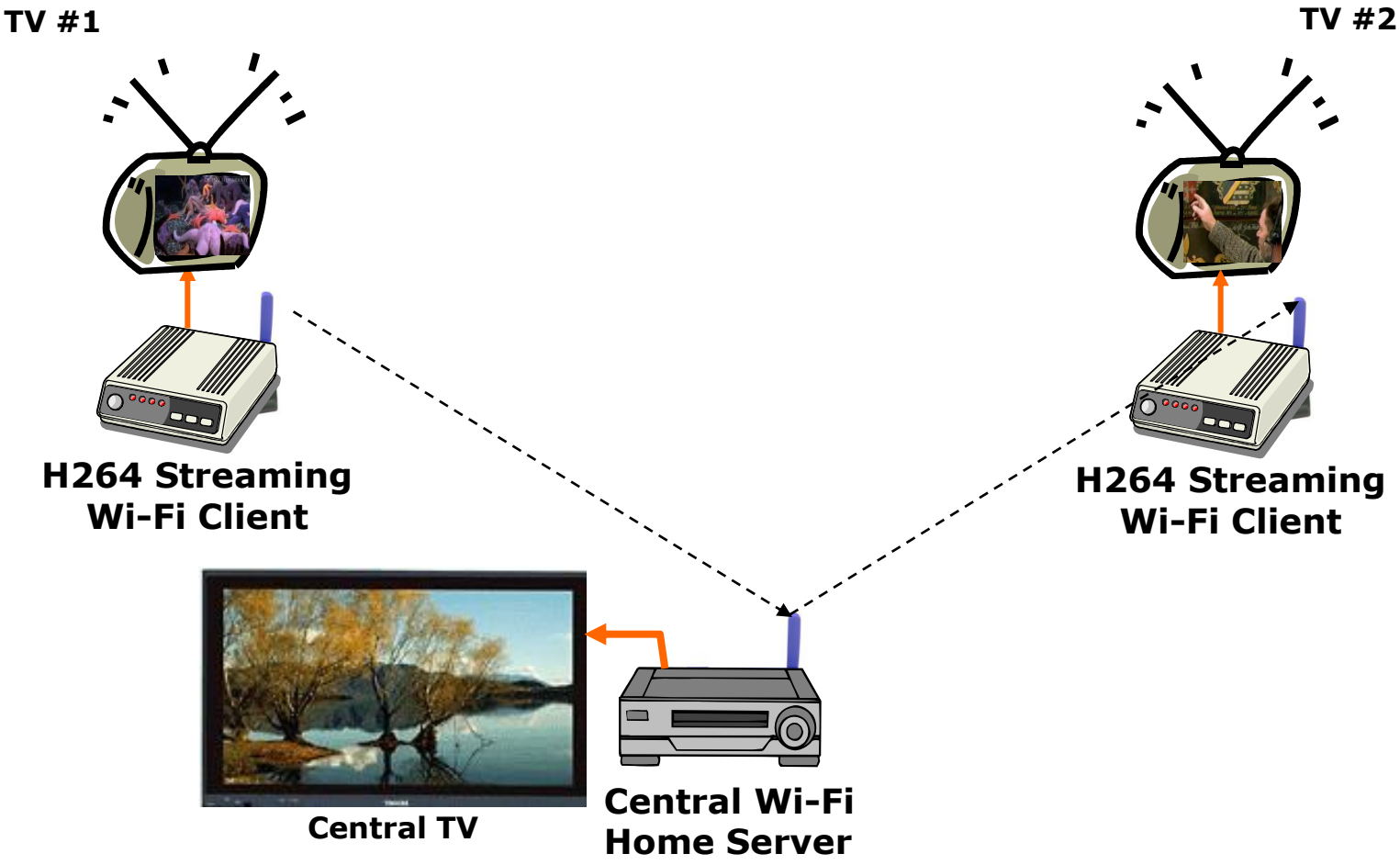
- Baseline Profile
  - Targets low-end devices
- Main Profile
  - Targets entertainment quality
  - Supports I, P, B slices; interlaced; CABAC
- Extended Profile
  - For streaming support; extra error resilience
- Here we target the Main Profile

H.264 Advantage



- 50% bit-rate reduction compared to MPEG-4
- 67% bit-rate reduction compared to MPEG-2

Application Scenario



Application Scenario (Contd.)

- Wireless enabled Home Media Server
  - Error-free transmission bandwidth
    - Decreases with increasing distance between connection points
  - MPEG-2 today requires ~4Mbps, which results in poor video quality due to lost packets
    - QoS is being worked on from the WLAN side as well
  - H.264 requires only ~1.4 Mbps
    - Enables entertainment quality
- Issue is
  - A low-cost H.264 Streaming Client

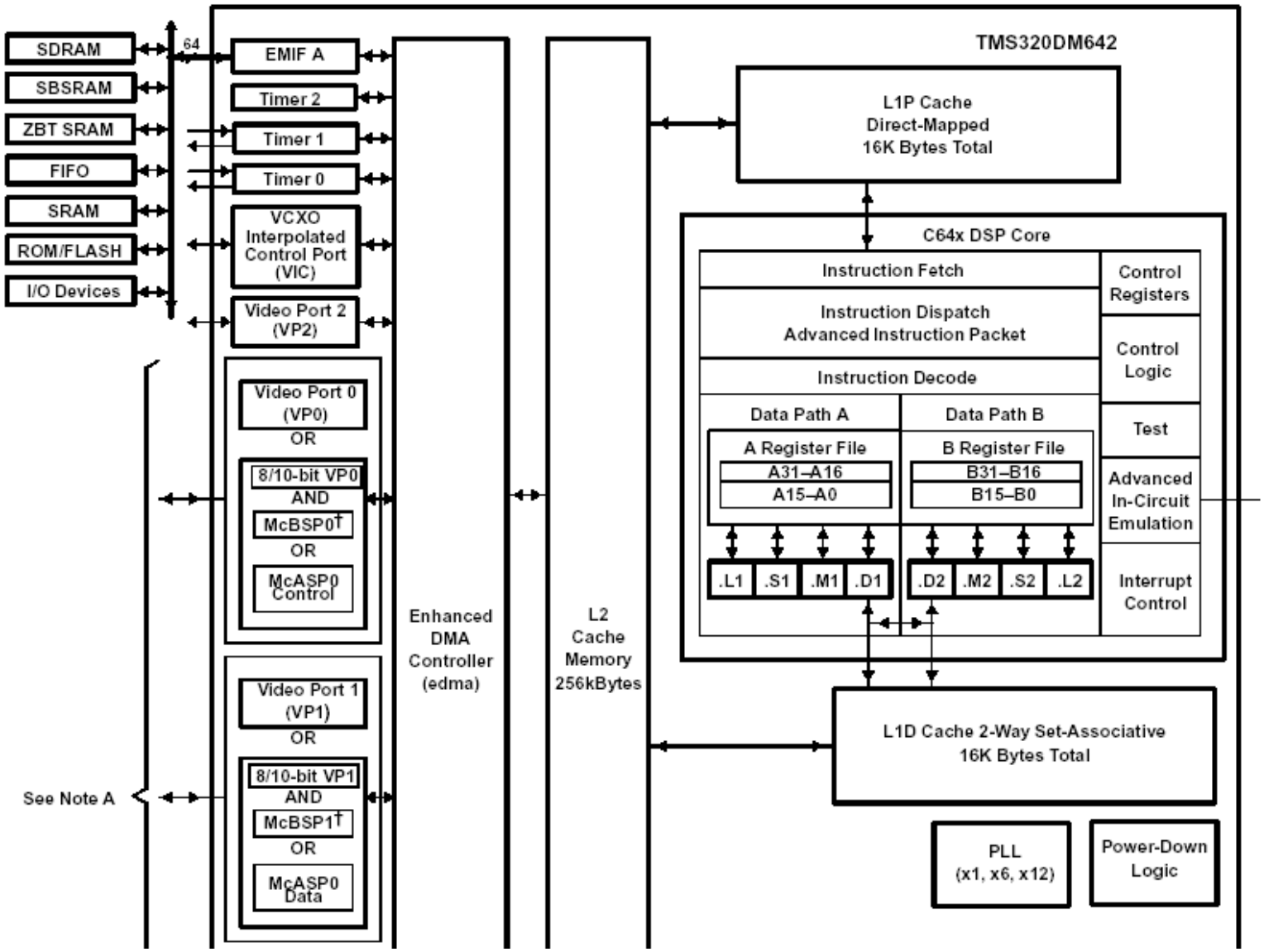
Challenges

- Computational complexity is significantly higher
  - While transform and quantization are simpler, the following add a lot of complexity
    - Deblocking filter in the loop
      - Adaptive; lots of conditional code
        - » Computing boundary strength; signal-based thresholds
      - Multiple filters (3 to 5 taps; variable coefficients)
    - Half-sample interpolation with a 6-tap filter
      - rather than bilinear interpolation
    - Intra prediction
      - Planar mode (takes ~10 operations per pixel)
      - Others (~5 operations per pixel)
    - Context-adaptive entropy coding
      - Multiple VLC tables or arithmetic coder contexts
      - Renormalization complexity in arithmetic decoding

Challenges

- Control flow limitations
  - Intra (4x4) prediction requires reconstructed samples
  - Deblocking cannot be performed immediately following the decoding of a macroblock
  - From motion vector decode to motion compensation, up to 96 distinct blocks of data have to be fetched for a single MB
    - Keeping the DSP non-idle
  - Slice can contain an arbitrary number of MBs
- Memory bandwidth problems
  - MVs for every 4x4 subblock from up to 15 ref. frames;
    - with 6-tap filter, requires 9x9 samples, and
    - with bi-directional prediction,
    - works out to ~10 frames for each frame
- Code size (cache misses)

TI TMS320DM642



Reproduced from SPRS200A , © Texas Instruments

TI C64x DSP Core

- VLIW architecture (VelociTI)
  - 256-bit instructions (8 32-bit)
  - Dual data paths
    - 4 functional units (L, M, S, D) each
      - L, S – arithmetic/logical; M – multiply/shift; D – Data load/store
    - 32 32-bit registers each
    - Cross path access to registers
  - Can load and store up to 64-bits
    - Non-aligned load/stores are supported
  - All instructions can be conditionally executed
  - Packed 8-bit and 16-bit operations
    - E.g. AVGU4, DOTPU4, ADD2, MIN2, MAX2, etc.
  - Instructions to pack and unpack
- Transfer crossbar with 4 queues of varying priority
- 64-channel enhanced DMA (TCInt, Chaining, linking)
- 2-level cache (16 kB I and D – L1; 256 kB L2)

TI DM642 Features

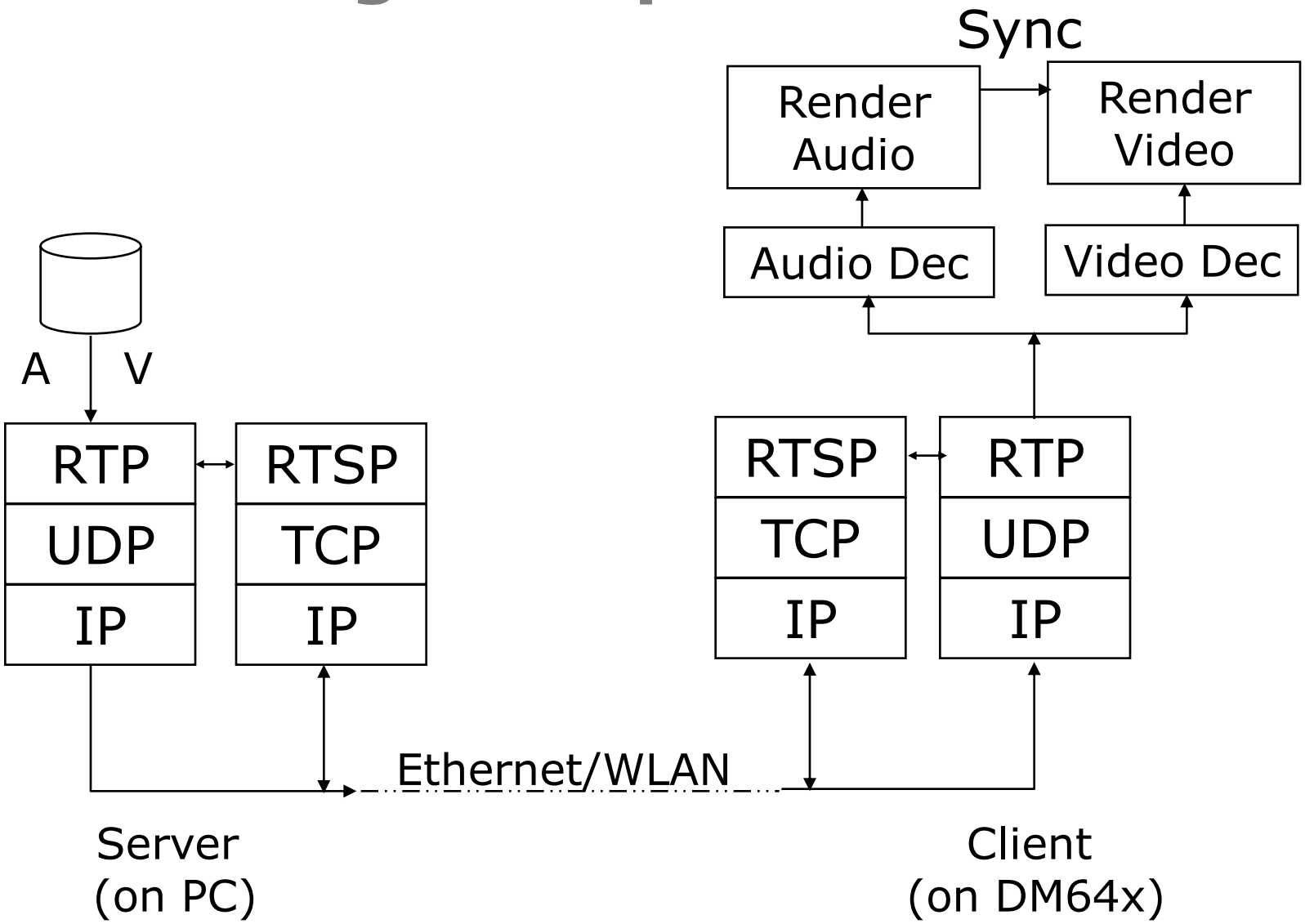
- 2 External Memory Interfaces
  - 1 64-bit (EMIFA); 1 16-bit (EMIFB)
- 3 video ports
  - Can be configured as video input, video output, or transport stream interface
- Multi-channel Audio Serial Port (Data/Control)
- 2 Multi-channel Buffered Serial Ports
- Ethernet MAC
- PCI interface
- Host Post Interface
- I2C Interface

TI DM642 EVM Features

- Composite/S-video input/output
- Line In/Out, Mic In
- 10/100 Mbps Ethernet Controller
- 32 MB SDRAM + 4 MB Flash
- PCI interface (boot option + HPI)
- EMIF Daughter Card Interface
- Video port Daughter Card Interface

… All interfaces to run a streaming A/V client

Streaming Set-up



Task Scheduling

- TI provides on the software framework side,
  - DSP BIOS
  - Networking Development Kit (NDK) with TCP/IP stack
    - Network Scheduler (Event/Timer based processing)
  - Drivers for video display and audio playback
    - Run as separate tasks managing the buffers and EDMA
- Scheduling concerns
  - UDP packets are not lost due to scheduling conflicts
  - Audio/Video synchronization is maintained
  - Buffers do not overflow
  - Achieved by
    - Suitably setting relative priorities of network scheduler, receive threads, and processing threads
    - Using semaphores to trigger a consumer task

Design Methodology

- Develop optimized assembly code for the leaf modules
  - Provides estimates on code size and MCPS
  - Ensure that leaf modules have a clean interface
    - To avoid having to re-write ASM modules later
  - Optimization guidelines
    - Analyze whether module is I/O limited or process limited
    - Maximize the utilization by packing as many slots as possible
      - Reduce slots by using appropriate SIMD instructions
      - Evaluate alternate ways of reducing the # of execution packets
      - Balance the load between data-paths A and B in a clean way to reduce cross-path accesses
    - Explore unrolling possibilities
      - balance code-size, register pressure and speedup
    - Minimize stalls due to branches and delay slots
    - Use conditional execution to avoid shallow branches
    - Avoid memory bank stalls

Exploiting Parallelism

- Algorithmic Parallelism
  - Multiple pixels can be worked on without any read-after-write dependencies
- Packed data operations
  - Using the packed data operations, multiple operations can be performed in one slot
    - E.g. AVGU4 - If pixel data is in bytes, compute four $(p_{i,j} + p_{i,j+1} + 1) >> 1$ operations in one instruction
    - E.g. DOTPU4 – $\Sigma w_{i*} p_i$ (4 8x8 multiplications and 3 adds)
  - Instructions that come in handy
    - AVGU4, DOTP4, DOTP2, SHRMB, SHLMB, UNPCK, PACK, SPACK, MIN2, MAX2, MPY4, SUBABS4, …
- Instruction level Parallelism
  - All operations at an instruction level that are independent and hence can be scheduled in parallel
    - Used to fill all the 8 slots in an execution packet
    - Only catch is the register pressure

Design Methodology (Contd.)

- Partition the decoding tasks such that
  - I-cache thrashing is minimized
  - Suitable pipelining can be established to minimize the cycles spent by the DSP waiting for data; (Keep pipeline granularity flexible)
    - Goal is to maximally utilize DSP and DMA
  - Required buffering due to above pipelining can be easily accommodated in the ISRAM
    - Re-use of scratch buffers minimizes L1 D-cache misses
  - Pick suitable groupings from syntax decoding, motion compensation, texture decoding, deblocking, and 4:2:2 conversion
  - Partitioned groups can be pipelined at MB, groups of MBs, or row of MBs
    - Larger groups can leverage statistical properties
- Verify the pipeline design
  - Use memcpys as placeholders for DMAs with suitable sync points in software (to verify completion of DMA)
  - Select the type of triggering for the various DMA channels
  - Implement the DMA and replace the memcpys
- Adjust code placements and buffer placements in ISRAM

EDMA considerations

- Use QDMA whenever possible for short bursts of transfer
- Use chaining whenever possible to automate triggering
- Optimize the triggering under an ISR to minimize context saving overheads
- Balance the load across the four priority queues and according to the criticality of the data
- Plan the buffer structures according to the restriction that 2D to 2D transfer is possible only if both source and destination strides are equal
- Ensure that DMAed area is not L2 cached
- Do not hard-code channels; use CSL functions

Performance

- In real-time on a DM642@600MHz
  - H.264 Main Profile Decoder
    - D-1 resolution at ~2 Mbps
  - MPEG-4 AAC-LC Decoder
    - Stereo at 44.1 KHz at 64kbps/ch
  - RTSP/RTP streaming client
  - Video/Audio rendering with synchronization
- Speed-ups on leaf modules up to 40x compared to –O3 compiled C code
  - Typical speed-ups in the range of 10-16x

Conclusions

- Real-time streaming client enabled on DM642
- Exploiting the VLIW architecture and packed-data instructions
  - Counters the high computational complexity
- Suitable task partition structure and multithreaded pipeline between DSP and DMA mitigates
  - Control flow limitations
  - Memory bandwidth issues
  - I-cache misses due to increased code size
- Further Challenges
  - Moving to a lower cost processor
    - such as DM640/641

Ittiam

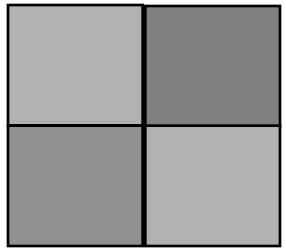
END

Segmented Motion Compensation

16x16

8x16

16x8



8x8

8x8

4x8

8x4

4x4

- Total of 1 - 16 motion vectors per MB

Ittiam H.264 Decoder on DM642

- Main Profile Decoder
  - Supports
    - CAVLC
    - CABAC
    - Segmented motion compensation
    - Multiple reference frames
    - Intra prediction
    - In-loop deblocking filter
  - Targets
    - Entertainment quality video
    - Applications such as PVRs and STBs
- A/V Synchronized Player Demo
  - Platform: TI DM642 EVM
  - File Format: AVI
  - Audio Decoder: MPEG-4 AAC-LC (64kbps/ch, stereo)
  - Video Bit-rate: 1.4 Mbps (VBR)
- Streaming Client Demo
  - RTSP/RTP based streaming