SoCs for Portable Video Applications: Architecture level Considerations

Mahesh Mehendale
m-mehendale@ti.com

EDP 2007
Agenda

- Video processing requirements of portable entertainment applications
- Characterizing “variability” in digital video processing
- Low power design techniques and their applicability in the context of Digital Video Sub-system and the SoC
- EDA challenges and Opportunities
Video Processing Chain

- **Capture**: Acquisition of original video content including A/D and sampling
- **Process**: Content is encoded, transcoded, transrated and/or analyzed
- **Deliver**: Content is transported via private or public networks
- **Receive**: Received content is stored, decoded and/or transcoded
- **View**: Content is accessible through a viewing mechanism
Personal Video Entertainment

- Portable Video Recorder
- Portable TV (DVB-T, DVB-H)
- Portable Media Player
- Digital Camcorder
- Portable Navigation
- Video phone
- Web terminal
Portable Media Player – video interfaces

- CCD/CMOS Camera
- DVB-T
- DVB-H
- NTSC/PAL/HDMI Decoder
- Analog Video
- WiFi
- USB
- TV Display
- LCD Display
- HDD
- MMC/SD
- NAND FLASH
Customer care-about

- Multi-standard, multi-format video processing

- Cost
- Power/Energy
Video formats

<table>
<thead>
<tr>
<th>Format</th>
<th>Pixels/frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>QCIF</td>
<td>144x176</td>
</tr>
<tr>
<td>QVGA</td>
<td>240x320</td>
</tr>
<tr>
<td>CIF</td>
<td>288x352</td>
</tr>
<tr>
<td>VGA</td>
<td>480x640</td>
</tr>
<tr>
<td>D1</td>
<td>480x720</td>
</tr>
<tr>
<td>WVGA</td>
<td>480x852</td>
</tr>
<tr>
<td>720P</td>
<td>720x1280</td>
</tr>
<tr>
<td>1080P</td>
<td>1080x1920</td>
</tr>
</tbody>
</table>
DV Engine Solution Space

Scalable, Configurable Multi-format, Multi-standard Digital Video Accelerator

Custom HWAs Dedicated per function

Area

Pwr

VLIW DSP

RISC μP
H.264 Encoder

- **Fn** (current)
- **F'n-1** (reference)
- **F'n** (constructed)

**Steps:**
1. **Motion Estimation**
2. **Motion Compensation**
3. **Choose Intra Predict**
4. **Intra Prediction**
5. **Transform, Inverse Transformation**
6. **Reorder**
7. **Entropy Encode**
8. **Vectors, Headers**
9. **Loop Filter**

- **Transform Scaling/Quantization**
- **Scaling, Inverse Transformation**
H.264 Decoder

- F'n-1 (reference)
- F'n (constructed)
- Loop Filter
- Motion Compensation
- Intra Prediction
- Entropy Decode
- Reorder
- Vectors, Headers
- Scaling, Inverse Transformation
- T^{-1}
- Q^{-1}
Driving Area Efficiency

- Leverage Decoder û Encoder functionality overlap

- Programmable HWAs for similar compute functions but with different parameters (such as number of taps), and/or different coefficients
  - DCT/IDCT, 8X8 vs 4x4
  - Quantization, scaling
  - Variable length coding
  - Interpolation (half pixel, quarter pixel)
  - Filtering

- Hardware-Software partition to meet the desired performance and programmability requirements with minimal area
Data driven variability in Decoding

- Fetch input bit stream
- Entropy decoder and reorder
- Inverse quantization, IDCT
  Fetch reference frame data for MC
- Motion compensation and add residues
- Deblocking Filtering

Data rate, System traffic

Data per frame

#MVs, I/B/P frames, System traffic

#MV, MV resolution, I/B/P frames

Boundary Strength
Low Power Design – across all levels
Component level support

**Silicon IP**

<table>
<thead>
<tr>
<th>Technology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retention SRAM and logic</td>
<td>SRAM and logic retention cells support dynamic power switching without state loss, lowering voltage and reducing leakage.</td>
</tr>
<tr>
<td>Dual-threshold voltages</td>
<td>Higher threshold for lower leakage and lower threshold for higher performance.</td>
</tr>
<tr>
<td>Power management cell library</td>
<td>Switching, isolation and level shifters support multiple domains in SOC implementations.</td>
</tr>
<tr>
<td>Process and temperature sensor</td>
<td>Adapts voltage dynamically in response to silicon processes and temperature variations.</td>
</tr>
<tr>
<td>Design flow support</td>
<td>Complete, nonintrusive support for easily integrating SmartReflex technologies.</td>
</tr>
</tbody>
</table>
## SoC level Power Management Strategies

<table>
<thead>
<tr>
<th>Technology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive Voltage Scaling (AVS)</td>
<td>Maintains high performance while minimizing voltage based on silicon process and temperature.</td>
</tr>
<tr>
<td>Dynamic Power Switching (DPS)</td>
<td>Dynamically switches between power modes based on system activity to reduce leakage power.</td>
</tr>
<tr>
<td>Dynamic Voltage and Frequency Scaling (DVFS)</td>
<td>Dynamically adjusts voltage and frequency to adapt to the performance required.</td>
</tr>
<tr>
<td>Multiple Domains (Voltage/Power/Clock)</td>
<td>Enables distinct physical domains for granular power/performance management by software.</td>
</tr>
<tr>
<td>Static Leakage Management (SLM)</td>
<td>Maintains lowest static power mode compatible with required system responsiveness to reduce leakage power.</td>
</tr>
</tbody>
</table>
Power optimal MHz-Vcc Operating Point

- Lower Vcc helps both dynamic and leakage power

- If Vcc is lowered while keeping MHz same – can result in area increase – impacting cost and negating any power gain

- At architecture level – MHz/Vcc for a given technology drives the degree of parallelism and pipelining

- The choice of target format for power optimization impacts area efficiency – for example, an implementation which gives lowest power for 720P resolution is likely to be different (higher area) than the implementation which gives lowest power for D1
**Power Reduction - at Application/Video stream level**

- If it’s decode function – turn off (clock gate/power down) encode functionality (e.g. Motion Estimation)

- For the standard and the profile to be processed, turn off hardware supporting all other standards and profiles (e.g. if MPEG4, turn off CABAC engine in entropy decoder)

- Dynamic frequency and voltage scaling – set the DV engine frequency and voltage operating points – depending on the resolution being supported – D1 at 30fps requires ~2.66 times lesser compute than 720P at 30fps.
Power Reduction @ Frame level

- Turn off un-used hardware depending on I vs P vs B frame
- Turn off un-used hardware depending on Interlaced vs Progressive content
Power Reduction @ MB Level

- Turn off individual hardware accelerators as soon as the computation for the current Macro-block is done (due to variability, the pipeline cannot be fully balanced).

- During motion-compensation the compute requirements vary depending on 1 motion vector vs 4 motion vectors per macro block, they also vary depending on motion vector resolution in terms of pixel vs half pixel vs quarter pixel.

- Turn off deblocking filter, if boundary strength is 0 or there is significant change (gradient) across block boundary in the original image.
Dynamic vs Leakage Power Scaling with Resolution

720P | D1 | CIF

- Dynamic
- Leakage
Power Reduction for CIF

- Compute requirements significantly lower, voltage scaling is limited by Vcc-min.
- Running the engine at lower frequency without lowering the voltage – does not help save energy

Multiple approaches:

1. Significant cycle overhead in completely switching off the engine and switching it back on – does not help at macro-block level, marginal gain at frame level, but done over a group of frames can give power reduction

2. Power down the engine but save the state using retention flops and putting memories in the retention mode – area overhead

3. Design the engine as a “bit slice” and switch off one half while processing CIF – has software implications.
DVFS – applicability at SoC level?

- Audio does not scale with resolution

- Any system function which demands real-time response in a narrow time window

- Modules in the video output processing chain which are tied to the resolution of the display device as against resolution of the video being processed

- Implies multiple voltage domains- can have system level cost implications from PMU standpoint
Managing data bandwidth

- Increasing resolution – implies scaling the IO bandwidth accordingly – but may not be feasible, practical – DDR speed limitations, SDRAM limitation, power, area impact etc.

- Need architecture level solution to address this bottleneck
  - On-chip buffers
  - On the fly computation
  - Improving efficiency of 2D transfers
  - SDRAM data organization
  - Algorithmic solutions?

- At lower resolution, can minimize SDRAM power by powering down unused banks
**EDA Challenges and Opportunities**

- System level power estimation/modelling
- Power management – synthesis and verification
- Physical design challenges
  - Automated clock gating
  - Physical design aware low power synthesis
  - Multi-Vt optimization
  - Timing closure at multiple corners (with DVFS – need to sweep Vmin and Vmax range)
- Building a configurable IP generator – supporting both run-time as well as compile-time scalability (e.g. building a MPEG4 Decode only engine optimized for power and area, with no software change)
Summary

- Portable video entertainment market needs a multi-format, multi-standard digital video engine with HD capability at low cost and low power.

- Highlighted the “variability” in the digital video processing needs including data driven variability.

- Discussed the entire spectrum of power management techniques and its applicability to the power minimization of the DV engine.

- Highlighted a few system level considerations and their architectural implications.

- Finally, presented EDA challenges and opportunity.
THANK YOU