Algorithmic Transforms for Efficient Energy Scalable Computation

Amit Sinha, Alice Wang and Anantha Chandrakasan
Massachusetts Institute of Technology
Outline

- Motivation
- Energy Scalability Notion
- Examples
  - Filtering
  - Image decoding
  - Detection
- Energy Scalable Video Application
- Dynamic Voltage Scaling
Energy scalability: Tradeoff quality for energy
Maximize quality for given energy availability

Energy Quality ($E-Q$) graph maximally concave
**Simple Example**

\[ y = f(x) = 1 + k_1 x + k_2 x^2 + \cdots + k_N x^N \]

<table>
<thead>
<tr>
<th>Original</th>
<th>Scalable</th>
</tr>
</thead>
</table>
| \( x_{\text{powi}} = 0.0; \)  
\( y = 1.0; \)  
for(\( i=1; \ i<N; \ i++ \)) {  
  \( x_{\text{powi}} *= x; \)  
  \( y += x_{\text{powi}}*k[i]; \)  
} | if(\( x > 1.0 \)) {  
  \( x_{\text{powi}} = \text{pow}(x,N); \)  
  \( y = k[N]*x_{\text{powi}}+1; \)  
  for(\( i=N-1; \ i>0; \ i-- \)) {  
    \( x_{\text{powi}} /= x; \)  
    \( y += x_{\text{powi}}*k[i]; \)  
  }  
} else { //original algo  
} |

- Incremental refinement
- Most-significant-first approach
Scalable Filtering

\[ y[n] = \sum_{k=0}^{N-1} h[k] x[n-k] \]

Original

Transformed

Re-order Index

Sorted Coeffs
Scalable Filtering E-Q

- 128 Tap FIR filtering on speech data
- StrongARM measurements
  - 5.12µJ per sample
  - 0.21µJ per sample overhead (4%)
Scalable Image Decoding: IDCT

\[ x[i, j] = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} c[u]c[v]X[u, v] \cos \left( \frac{(2i+1)u\pi}{16} \right) \cos \left( \frac{(2j+1)v\pi}{16} \right) \]

- Scalability at the cost of slightly more operations
Most Significant DCT Coefficients

- Most energy concentrated in lower coefficients
- Accumulate lower frequency components first
Incremental Refinement: Scalable IDCT

Scalable [FM-IDCT]

Non-Scalable [Chen]
Detection using Beamforming
- Higher SNR and robustness
- Data aggregation $\Rightarrow$ Lesser transmission overhead

Scalable sensor networks
- Adapt to time varying resources
- Graceful $E-Q$ degradation
Unscalable Beamforming

- Use preset order \(<1,2,3,4,5,6>\>
- Determine quality as source moves from A \(\rightarrow\) B
- E-Q performance depends on source location
Scalable Beamforming

- Initial pre-processing improves performance
  - Most significant first transform
  - Determine signals with large SNR
  - Use quick-sort algorithm to obtain order
- Requires low overhead transformation
Dynamic Voltage Scaling

Just-in-time processing:
- Variable power supply & frequency
- Quadratic savings [Gutnik97]
Energy scalable Java based video decoder

- Performs just-in-time computation
- Allows energy-accuracy-throughput tradeoffs
- Uses energy aware software models
Conclusions and Future Work

- **Energy-Quality scalability important**
  - Time varying resources
  - Want highest possible quality for given energy availability

- **Incrementally refining algorithms required**
  - Algorithmic transformations
  - Most significant computations first
  - Nominal preprocessing overhead

- **Dynamic Voltage Scaling gives more savings**

- **OS hints, JIT compilation for energy scalability?**