Approximate implementation of DCT unit and analyzing its effect on the JPEG encoder unit

Amir Yazdanbakhsh

University of Wisconsin-Madison
Electrical and Computer Engineering

yazdanbakhsh@wisc.edu

May 10, 2013
Do we always need correct data?

- There are many applications which do not need **strict correctness**
- The existing **flexibility in output correctness** requirement in these applications can be utilized to improve design objectives such as power, area, and delay
- Multimedia applications are one of the domains in which the **approximate computing** can be utilized to reduce the cost of hardware design
JPEG encoding

- JPEG encoding is a widely used lossy compression in digital photography

Figure: Sample baseline JPEG encode data flow [eetimes.com]
# JPEG encoding profiling

<table>
<thead>
<tr>
<th>Procedure name</th>
<th>Number of calls</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>emit_byte</td>
<td>625</td>
<td>Write next output bytes</td>
</tr>
<tr>
<td><strong>forward_DCT</strong></td>
<td>585</td>
<td>Perform forward DCT on one or more blocks of a component</td>
</tr>
<tr>
<td>jpeg_fdct_islow</td>
<td>551</td>
<td>Perform the forward DCT on one block of sample</td>
</tr>
<tr>
<td>jpeg_fdct_16x16</td>
<td>300</td>
<td>Perform the forward DCT on a 16x16 sample block</td>
</tr>
<tr>
<td>jcopy_sample_rows</td>
<td>258</td>
<td>Copy some rows of samples from one place to another.</td>
</tr>
<tr>
<td>fullsize_downsample</td>
<td>225</td>
<td>Downsample pixel values of a single component.</td>
</tr>
</tbody>
</table>

**Table:** The name of JPEG encoder procedures and their number of calls for one image conversion.

Amir Yazdanbakhsh

Approximate implementation of DCT for JPEG encoder unit
Figure: DCT architecture for low frequency components (DC component).
DCT Architecture (Approximate Adders)

Figure: DCT architecture for high frequency components.
Kogge Stone Adder

Figure: 16-bit Kogge Stone Adder
Kogge Stone Adder Equations

Generate and Propagate Generator

\[ \text{Generate} = A \text{ and } B \]
\[ \text{Propagate} = A \text{ or } B \]
### Kogge Stone Adder Equations

#### Generate and Propagate Generator

\[
\begin{align*}
\text{Generate} & = A \text{ and } B \\
\text{Propagate} & = A \text{ or } B
\end{align*}
\]

#### Carry Generator

\[
\begin{align*}
\text{Generate} & = G_{i1} \text{ or } (G_{i2} \text{ and } P_{i1}) \\
\text{Propagate} & = P_{i1} \text{ and } P_{i2}
\end{align*}
\]
### Kogge Stone Adder Equations

#### Generate and Propagate Generator

$$
\text{Generate} = A \text{ and } B \\
\text{Propagate} = A \text{ or } B
$$

#### Carry Generator

$$
\text{Generate} = G_{i1} \text{ or } (G_{i2} \text{ and } P_{i1}) \\
\text{Propagate} = P_{i1} \text{ and } P_{i2}
$$

#### Sum Calculation

$$
\text{Sum}_0 = \text{Cin} \text{ xor } P0_i \\
\text{Sum}_i = (G_{\text{last level}_{i-1}} \text{ or } (P_{\text{last level}_{i-1}} \text{ and } \text{Cin})) \text{ xor } P0_i
$$
Correct Carry Generator

Figure: Karnaugh map for generate in "% Carry Generator" module.
Approximate Carry Generator

Figure: Karnaugh map for approximate generate in "Carry Generator" module.
What is done in this project

- Re-implement the forward DCT of JPEG encoder unit in Verilog
- Implement the approximate Kogge Stone adder in Verilog and incorporate it into the DCT unit (for the first time)
- Synthesis the DCT unit to evaluate the design parameters such as area, power, and delay
- Evaluate the accuracy of JPEG encoder with approximate DCT unit and compare it with exact implementation
- Implement the MATLAB code to build the approximate JPEG photo
Synthesis results

<table>
<thead>
<tr>
<th></th>
<th>Area ($\mu m^2$)</th>
<th>Delay (ns)</th>
<th>Dynamic Power ($\mu W$)</th>
<th>Leakage Power ($\mu W$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>19663.812006</td>
<td>1.25</td>
<td>1709.1</td>
<td>501.5362</td>
</tr>
<tr>
<td>Approximate</td>
<td>12018.323225</td>
<td>1.1</td>
<td>632.2219</td>
<td>293.2546</td>
</tr>
</tbody>
</table>

**Table:** Comparison of design parameters between correct and approximate DCT implementation

- Design Compiler with 32nm Standard Cell Library
- Power numbers are obtained with Design Compiler without real simulation
Quality comparison

Figure: JPEG photo with correct Kogge Stone adder
Quality comparison

Figure: JPEG photo with approximate Kogge Stone adder
In memory of the celebrated Persian poet Hafez
Approximate implementation of DCT for JPEG encoder unit

Amir Yazdanbakhsh