A Dual Voltage Dual Frequency Low Power VLSI Chip for Image Watermarking

Saraju P. Mohanty
Dept. of CSE, University of North Texas
smohanty@cs.unt.edu
http://www.cs.unt.edu/~smohanty/
Outline of the Talk

- Introduction
- Why Low Power?
- Related Works
- Watermarking Algorithms
- Proposed Architecture
- Prototype Chip Implementation
- Conclusions
Why Low-Power?

Major Motivation
Extending battery life for portable applications
Why Low-Power? .......

Battery lifetime

Cooling and energy costs

Environmental concerns

System reliability
Power Consumption in CMOS Circuits

Leakage Current: Reverse biased current in the parasitic diode and subthreshold current due to charge inversion existing at gate below $V_T$.

Standby Current: Continuous DC current from $V_{dd}$ to ground

Short-Circuit Current: DC current from $V_{dd}$ to ground during output transition

Capacitive Current: Flows to charge discharge capacitive loads.
Dynamic Power Consumption

Let, $C_L = \text{load capacitor}, \ V_{dd} = \text{supply voltage}, \ N = \text{average number of transitions/clock cycle} = E(sw) = \alpha$ and $f = \text{clock frequency}$. The dynamic power consumption for CMOS:

$$P_{\text{dynamic}} = \frac{1}{2} C_L V_{dd}^2 N f$$

• Veendrick Observation: In a well designed circuit, short-circuit power dissipation is less than 20% of the dynamic power dissipation.

• Sylvester and Kaul: At larger switching activity the static power is negligible compared to the dynamic power.

We focus on dynamic power reduction !!
Dynamic Power Reduction

- Reduce Supply Voltage ($V_{dd}$): delay increases; performance degradation
- Reduce Clock Frequency ($f$): only power saving no energy savings; performance degradation
- Reduce Switching Activity ($N$ or $E_{sw}$): no switching no power loss !!! Not fully under designers control. Switching activity depends on the logic function and correlations are difficult to handle.
- Reduce Physical Capacitance: done by reducing device size reduces the current drive of the transistor making the circuit slow
Our Approach?

Adjust the frequency and supply voltage in a co-coordinated manner to reduce dynamic power while maintaining performance.
Digital Watermarking?

It is mine

No, it is mine

Multimedia Object

Whose is it this? How to know? What’s the solution of this ownership problem?

Solution: “WATERMARKING”
Digital Watermarking

Digital watermarking is a process for embedding data (watermark) into a multimedia object for its copyright protection and authentication.

Types

• Visible and Invisible
• Spatial/DCT/ Wavelet
• Robust and Fragile
An Watermarked Image (from IBM)
Another Example .......
Watermarking: General Framework

- Encoder: Inserts the watermark into the host image
- Decoder: Decodes or extracts the watermark from image
- Comparator: Verifies if extracted watermark matches with the inserted one
Why Hardware Implementation?

Hardware implementations of watermarking algorithms necessary for various reasons:

– Easy integration with multimedia hardware, such as digital camera, camcorder, etc.
– Low power
– High performance
– Reliable
– Real time applications
## Previous Work
(Hardware based Watermarking)

<table>
<thead>
<tr>
<th>Work</th>
<th>Type</th>
<th>Target Object</th>
<th>Domain</th>
<th>Technology</th>
<th>Chip Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strycker, 2000</td>
<td>Invisible Robust</td>
<td>Video</td>
<td>Spatial</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Tsai and Lu 2001</td>
<td>Invisible Robust</td>
<td>Video</td>
<td>DCT</td>
<td>0.35µ</td>
<td>62.8 mW</td>
</tr>
<tr>
<td>Mathai, 2003</td>
<td>Invisible Robust</td>
<td>Image</td>
<td>Wavelet</td>
<td>0.18µ</td>
<td>NA</td>
</tr>
<tr>
<td>Garimella, 2003</td>
<td>Invisible Fragile</td>
<td>Image</td>
<td>Spatial</td>
<td>0.13µ</td>
<td>37.6 µW</td>
</tr>
</tbody>
</table>
Previous Work: Summary

- Many software implementations of watermarking algorithms.
- Only few hardware implementations.
- Just one hardware implementation in frequency domain which can insert only invisible watermark.
- All other implementations in spatial domain.
Highlights of our Designed Chip

- DCT domain Implementation
- First to insert both visible and / or invisible watermark
- First Low Power Design for watermarking using dual voltage and dual frequency
- Uses Pipelined / Parallelization for better performance
Watermarking through JPEG Encoder

Encoder model

Input Image

DCT

Watermark Insertion Module

Watermark

Quantizer

Quantization Table

Entropy Encoder

Output Image
Watermarking Through Digital Still Camera

Input Image Sensor → A/D Converter → DSP Processor

Watermarking Processor

Watermarking Datapath → Watermarking Controller

Controller and Interface

Memory (Flash, SDRAM)
Invisible Algorithm Implemented

1. Divide the original image into blocks.
2. Calculate the DCT coefficients of all the image blocks.
3. Generate random numbers to use as watermark.
4. Consider the three largest AC-DCT coefficients of an image block for watermark insertion.

Visible Algorithm Implemented

1. Divide Original and watermark image into blocks.
2. Calculate DCT coefficients of all the blocks.
3. Find the edge blocks in the original image.
4. Find the local and global statistics of original image using DC-DCT and AC-DCT coefficients.
5. The mean of DC-DCT coefficients and mean and the variance of AC-DCT coefficients are useful.
6. Calculate the Scaling and embedding factors.
7. Add the original image DCT coefficients and the watermark DCT coefficients block by block.

The Proposed Architecture

Invisible Watermarking

Visible Watermarking

Original Image

Watermark Image

DCT Module

Edge Detection Module

Perceptual Analyzer Module

Scaling and Embedding Factor Module

Random Number Generator Module

Invisible Insertion Module

Visible Insertion Module

Watermarked Image
Highlights of the Proposed Architecture

- Hierarchical architecture.
- Decentralized controller scheme.
- Parallelism and Pipelining exploited.
- Dual Voltage and dual frequency mode operation
Modules in Proposed Architecture

- DCT Module: Calculates the DCT coefficients.
- Edge Detection Module: Determines edge blocks.
- Perceptual Analyzer Module: Determines perceptually significant regions using original image statistics.
- Scaling and Embedding Factor Module: Determines the scaling and embedding factors for visible watermark insertion.
- Watermark Insertion Module: Inserts the watermark
- Random Number Generator Module: Generates random numbers.
Modules in Proposed Architecture

From controller

Input Image

Buffers (constants)

DCT_X

DCT_Y

Decoder Flip-Flop

Latch

AC-DCT of a block

Accumulator

Comparator

Threshold

Mean

Max

17

Edge Detector

Edge or Nonedge Block

DC DCT

AC DCT

DCT Module

Edge Detection Module
Modules in Proposed Architecture

- Perceptual Analyzer Module
  - DC Mean
  - AC Mean
  - AC Variance

- Scaling and Embedding Factor Module
  - Alpha-Beta Module
    - Scaling Module
    - Scaling Module

见识了 DC 和 AC 对应的 DCT 处理，以及模块间的嵌入和嵌入因子 

Dept. of Comp. Sc. & Engineering
S. P. Mohanty
Modules in Proposed Architecture

Pseudorandom numbers generated using LFSR.
Modules in more Detail: DCT Module

DCT module implements the following set of equations.

\[
\begin{align*}
x_{00} &= ((in_{00} \times c_{00}) + (in_{01} \times c_{01}) + (in_{02} \times c_{02}) + (in_{03} \times c_{03})) \\
x_{10} &= ((in_{10} \times c_{00}) + (in_{11} \times c_{01}) + (in_{12} \times c_{02}) + (in_{13} \times c_{03})) \\
x_{20} &= ((in_{20} \times c_{00}) + (in_{21} \times c_{01}) + (in_{22} \times c_{02}) + (in_{23} \times c_{03})) \\
x_{30} &= ((in_{30} \times c_{00}) + (in_{31} \times c_{01}) + (in_{32} \times c_{02}) + (in_{33} \times c_{03})) \\
y_{00} &= ((x_{00} \times c_{00}) + (x_{10} \times c_{01}) + (x_{20} \times c_{02}) + (x_{30} \times c_{03})) \\
y_{01} &= ((x_{00} \times c_{10}) + (x_{10} \times c_{11}) + (x_{20} \times c_{12}) + (x_{30} \times c_{13})) \\
y_{02} &= ((x_{00} \times c_{20}) + (x_{10} \times c_{21}) + (x_{20} \times c_{22}) + (x_{30} \times c_{23})) \\
y_{03} &= ((x_{00} \times c_{30}) + (x_{10} \times c_{31}) + (x_{20} \times c_{32}) + (x_{30} \times c_{33}))
\end{align*}
\]
DCT module implemented as arrays of multipliers and adders.
Scaling and Embedding Factor, Visible Insertion

Dept. of Comp. Sc. & Engineering

University of North Texas

S. P. Mohanty
Modules in more Detail: Invisible Insertion

- Insertion module implemented with a multiplier and an adder.

\[ C_{I_{Wk}} = C_{I_k} + \alpha r_{ik} \]
Pipeline and Parallelism

stage 1

DCT X

Forwarding Logic

stage 2

Edge Detection Submodule 1

Perceptual Analyzer Submodule 1

Perceptual Analyzer Submodule 3

Invisible Insertion Module

DC

3 largest AC coefficients

AC

stage 3

Edge Detection Submodule 2

Forwarding Logic

Perceptual Analyzer Submodule 2

Watermarked DCT coefficient
Dual Voltage and Dual Frequency

- Lower Voltage
  - DCT_X
  - DCT_Y
- Slower Clock
- Normal Voltage
  - Edge Detection Module
  - Perceptual Analyzer Module
  - Scaling and Embedding Factor Module
  - Visible Watermark Insertion
  - Invisible Watermark Insertion
- Normal Clock

Dept. of Comp. Sc. & Engineering

S. P. Mohanty
Dual Voltage: Level Converters

• Level converters required to step up the low voltage to high voltage.
• Traditional level converter: Differential Cascode Voltage Switch (DCVS).
• In this work: Single Supply Level Converters – faster, better power consumption, needs single voltage supply only.

Layout and Schematic of SSLV
## Prototype Chip Implementation: Tools Used

<table>
<thead>
<tr>
<th>Tools</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cadence NClaunch</td>
<td>VHDL simulator</td>
</tr>
<tr>
<td>Synopsys Design Analyzer</td>
<td>Verilog netlist generation</td>
</tr>
<tr>
<td>Cadence Silicon Ensemble</td>
<td>Layout, Placement and routing</td>
</tr>
<tr>
<td>Cadence Virtuose tool</td>
<td>Layout Editing</td>
</tr>
<tr>
<td>Cadence Abstract Generator</td>
<td>Abstract generation</td>
</tr>
<tr>
<td>Synopsys Nanosim</td>
<td>Power and delay calculations</td>
</tr>
</tbody>
</table>

**Standard Cell** Design Style adopted. Standard Cells obtained from Virginia Tech. Technology: TSMC 0.25 µm
Prototype Chip Implementation: Design Flow

- VHDL
- Structural Netlist Generation
- Placement and Routing
- Layout
- Abstract Generation
- Technology file
- Design Constraints
- Design Exchange Format (DEF)
- DRC and Extraction Rules
- Layout Exchange Format (LEF)
- Redo Design
- Normal Flow
Design Flow Example: VHDL Code

entity edm3 is
port (clk, reset, vdd1, enable, wss1 : in std_logic;
     AnMax, An : in std_logic_vector(16 downto 0);
     edge_block, done, write_edm3 : out std_logic;
     countout : out std_logic_vector(7 downto 0));
end entity edm3;

architecture behav of edm3 is

component counter8 is
port (clk : in std_logic;
      reset, vdd1, enable : in std_logic;
      q : inout std_logic_vector(7 downto 0));
end component counter8;

signal An_max, AnMax_by_2 : std_logic_vector(16 downto 0);
signal count_out : std_logic_vector(7 downto 0);
signal At_a, B, Bt, count, edgeblock, en_count, write, proces,
temedgeblock : std_logic;

begin

COUNTER: counter8 port map (clk=>clk, reset=>reset, enable=>write, vdd1=>vdd1,
q=>count_out);

countout<=count_out;

counting: process(count_out) is
begin
if (count_out="11111111") then
  count<="1";
else
  count<="0";
end if;
end process;


Design Flow Example: Synthesized Verilog Netlist
Design Flow Example: Placement and Routing
Design Flow Example: Layout
Design Flow Example: Abstract Generation
Overall Prototype Chip: Layout
Prototype Chip: Pin diagram

Low Power Chip for Image Watermarking

- vdd1
- vdd2
- Original Image
- Watermark Image
- alpha
- I/V'
- enable
- reset
- clk1
- clk2

Watermarked Image
- done
- busy
Prototype Chip: Statistics

- **Technology**: TSMC 0.25 µ
- **Total Area**: 16.2 sq mm
- **Dual Clocks**: 280 MHz and 70 MHz
- **Dual Voltages**: 2.5V and 1.5V
- **No. of Transistors**: 1.4 million
- **Power (dual voltage and frequency)**: 0.3 mW
- **Chip (single voltage and frequency)**: 1.9 mW
Conclusion and Future Work

• Dual Voltage, Dual frequency watermarking chip was developed.
• Invisible / Visible insertion
• Pipelined and Parallelized architecture for performance.
• Frequency domain implementation for real time audio and video watermarking.
• Real time watermark extraction.
• Need more robust watermarking algorithms.