Embedding Low Cost Optimal Watermark During High Level Synthesis for Reusable IP Core Protection

Anirban Sengupta, Saumya Bhadauria
Computer Science and Engineering
Indian Institute of Technology, Indore, India
Email: asengupt@iiti.ac.in

Saraju P. Mohanty
Computer Science and Engineering
University of North Texas, USA
Email: saraju.mohanty@unt.edu
Outline of this Presentation

- Introduction
- Proposed methodology
- Proposed particle-swarm based approach for optimal watermark generation
- Proposed method for signature detection
- Properties of watermark generated
- Experimental results
Intellectual Property (IP) Core …

- Consumer Electronics is realized as SoC for low-power, low-cost and high performance requirements.
- Consumer Electronics SoC design challenges include:
  - Lower Cost, Lower Design Cost, and Shorter Time-to-Market

- IP cores based system design is used to meet the challenges
- IP cores (often supplied by third party vendors)
  - Maximize design productivity, minimize design time
An IP Core is a reusable unit of logic, block, component, cell, or layout design that is developed for licensing to multiple vendors to use as building blocks in different system designs.

3-Times of IP Cores

- Hard Cores
- Firm Cores
- Soft Cores
IP Core based Design and Manufacturing

- Due to globalization of design supply chain, possibility of intervention and attacks on IP cores is on the rise
  → mandates protection of IP cores from piracy/counterfeiting even at early stage of design flow
IP Core – Selected Issues/Challenges

Ownership Abuse

Multiple Usage with 1 license, Sub-licensing

Piracy by fraudulent means or reverse engineering

Malicious design modifications

Trojans

Accurately Identify Ownership

Protection

IP core is really doing what it supposed to do

Trust
Selected Solutions for IP Protection

Proposed Approach of the current paper:
- Optimization done for embedding cost
- Optimization done for hardware area
- Multi-variable signature approach
One Solution of IP - Watermarking

- Watermarking has been widespread use in other disciplines: currency, bank checks, multimedia content, etc. It is a natural thinking that watermarking can be deployed for hardware/software IP protection.

- This paper presents a technique for generating low cost watermarking solution during HLS based on multi-variable signature encoding for protection of reusable IP cores.

- Embedding a robust watermark at a high abstraction level (such as behavioral) can serve as a line of defense against:
  - Attacks
  - Nullifying false claim of ownership
  - Protecting the value of a usable IP core
Watermarking for Hardware IP Protection

- A watermark is a signature of the owner embedded in a IP core.

- A watermark:
  - should be capable to identify the owner/creator of the design
  - should be robust and difficult to remove
  - should be resilient against attacks like: ghost signature and tampering
  - should have minimal embedding cost to obtain the watermarked design
  - should be embedded in the IP design with minimal computation effort
  - should be easy to detect signature at the genuine receivers end for the receiver who has full knowledge of the signature encoding rule
Watermark – At High-Level – Prior Works

- Limited literature on watermarking for IP protection at the high-level or behavioral synthesis phase of IP design cycle.
- Hong-2005 [1]: A combination of 0 and 1 is used to encode signature in the form of adding additional edges in the colored interval graph during HLS.
- Gal-2012 [10]: Presented a watermarking based on mathematical relationships between numeric values as inputs and outputs at specified times.

- Drawbacks of existing works:
  - signature is susceptible to attacks/compromise, if encoding rule of both the variable is known.
  - watermark has high embedding cost and high storage overhead.

- To advance the state-of-the art, this current paper presents a cost optimal watermark based on robust multi-variable signature encoding during HLS for reusable IP core protection.
Proposed High-Level Synthesis Flow for IP Protection – A Simplified View

1. HDL Description of IP Core
2. Compilation and Transformation
3. Operation Scheduling
4. Resource Allocation
5. Operation Binding or Assignment
6. Watermark Constrained Register Allocation
7. Datapath and Control Generation
8. Watermarked IP Core of a Digital Design
Proposed Watermarking …

Process for embedding watermark in the design

• Schedule the CDFG based on resource configuration provided.
• Create the colored interval graph to find the minimum number of registers required for allocation.
• Generate a controller based on colored interval graph.
• Sort storage variables as per their number in increasing order.
• Generate a desired signature in the form of random combination of a tuple comprising of \((i, I, T, !)\). Each variable of the generated signature maps onto a certain edge pair:
  • \(i\) = encoded value of edge with node pair as (prime, prime)
  • \(I\) = encoded value of edge with node pair as (even, even)
  • \(T\) = encoded value of edge with node pair as (odd, even)
  • \(!\) = encoded value of edge with node pair as (0, any integer)
Proposed Watermarking …

Process for embedding watermark in the design

- Build a list $L[k]$ of additional edge pairs corresponding to its encoded values by traversing the sorted nodes.
- Insert additional edges as watermark in colored interval graph if a node is not already present in the graph.
- Modify controller design on the basis of created watermark.

Scheduling of a CDFG with 3 adders and 4 multipliers

<table>
<thead>
<tr>
<th>Control Step (c.s)</th>
<th>Red (R)</th>
<th>Blue (B)</th>
<th>Green (G)</th>
<th>Yellow (Y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>v0</td>
<td>v1</td>
<td>v2</td>
<td>v3</td>
</tr>
<tr>
<td>1</td>
<td>v4</td>
<td>v5</td>
<td>v6</td>
<td>v7</td>
</tr>
<tr>
<td>2</td>
<td>v8</td>
<td>v9</td>
<td>v6</td>
<td>v10</td>
</tr>
<tr>
<td>3</td>
<td>v11</td>
<td>v12</td>
<td>v13</td>
<td>--</td>
</tr>
<tr>
<td>4</td>
<td>v14</td>
<td>v12</td>
<td>v15</td>
<td>--</td>
</tr>
<tr>
<td>5</td>
<td>v16</td>
<td>v12</td>
<td>v15</td>
<td>--</td>
</tr>
<tr>
<td>6</td>
<td>v17</td>
<td>--</td>
<td>v15</td>
<td>--</td>
</tr>
</tbody>
</table>
Colored Interval Graph for the scheduling

Colored Interval Graph with additional edges (watermarking constraints) colored in grey

Desired signature (7-digit) | Corresponding additional edges to add in the colored interval graph
--- | ---
i | (2,3)
i | (2, 5)
I | (2, 4)
I | (2, 6)
T | (1, 2)
T | (1, 4)
! | (0, 1)

Signature and its decoded meaning

Control Step (c.s) | Red (R) | Blue (B) | Green (G) | Yellow (Y)
--- | --- | --- | --- | ---
0 | v0 | v1 | v2 | v3
1 | v4 | v5 | v7 | v6
2 | v8 | v9 | v10 | v6
3 | v11 | v12 | v13 | --
4 | v14 | v12 | v15 | --
5 | v16 | v12 | v15 | --
6 | v17 | -- | v15 | --

Controller for register allocation after embedding watermark
Motivation for Design Space Exploration (DSE) of Optimal Watermark

- Every solution impacts the latency and hardware area in a different way.
- Choosing a solution without performing trade-off affects the latency and area of the final IP core design.
- Before deciding a solution for inserting a watermark that yields lowest cost, many factors have to be considered.
- DSE process helps in identifying an optimal watermarked solution, which satisfies the user specified upper bounds of latency and hardware area as well as ensures that a low cost solution is found.
Proposed Particle Swarm Optimization (PSO) driven DSE for Optimal Watermark

Input Engine
- Module Library
- DFG
- User Constraints
- Control parameter e.g. Swarm size, # iteration, acceleration coefficient

DSE Engine
- Area Evaluation
- Execution Time Evaluation
- PSO- DSE
- Construct a CDFG based on $R_x$

Optimal Solution

Proposed Watermarking Engine
- Construct $k$-connected colored interval graph
- Update controller by imposing watermark constraint and construct the equivalent datapath
- Modify colored interval graph based on watermarking constraints added
- Generate original controller design
- Modify colored interval graph based on watermarking constraints added
- Update controller by imposing watermark constraint and construct the equivalent datapath
- Update controller by imposing watermark constraint and construct the equivalent datapath
- Update controller by imposing watermark constraint and construct the equivalent datapath

Signature Encoding
- Select desired signature using proposed encoding
- Decode signature to arrive at watermarking constraints (additional edges)

Signature Encoding
- RSA

Input Engine Module Library DFG User Constraints Control parameter e.g. Swarm size, # iteration, acceleration coefficient

DSE Engine Area Evaluation Execution Time Evaluation PSO- DSE Construct a CDFG based on $R_x$

Optimal Solution

Proposed Watermarking Engine Construct $k$-connected colored interval graph Update controller by imposing watermark constraint and construct the equivalent datapath Modify colored interval graph based on watermarking constraints added Generate original controller design Decode signature to arrive at watermarking constraints (additional edges)

Signature Encoding RSA

Indian Institute of Technology Indore
Proposed Optimization Methodology

Problem Formulation

- Given a control data flow graph (CDFG), determine, optimal watermarked solution \( X_i = N(R_1), N(R_2), \ldots N(R_D) \) with minimum Hybrid Cost \( A_T, L_T \)

\[
C_f (X_i) = W_1 \frac{L_T - L_{cons}}{L_{max}} + W_2 \frac{A_T - A_{cons}}{A_{max}}
\]

Subjected to: \( A_T \leq A_{cons}, L_T \leq L_{cons} \), and

\( w \) is # of watermarking constraint generated corresponding to a signature

\( A_T \) and \( L_T \) are area and delay of watermarked solutions

\( A_{max} \) and \( L_{max} \) correspond to solutions with maximum area and delay in the design space

\( W_1, W_2 \) are the user defined weights, e.g. both 0.5 for equal weightage

\( N(R_D) \) is the number of a resource type \( R_D \)
Watermark Signature Detection

- Reverse Engineering
- Signature Verification

Original IP from sender

Reverse Engineering
- Collect structural information
- Create controller
- Evaluate presence of extra constraints in controller design

Output: Yes/ No

Watermarked cipher text from sender

Signature Verification
- Decrypt cipher text using RSA
- Decode the decrypted signature using knowledge of encoding rules
- Re-construct colored interval graph to realize the extra edges as watermarking constraints

Yes $\rightarrow$ Received IP is authentic
No $\rightarrow$ Received IP is compromised

If Watermarked cipher text from sender:

Yes $\rightarrow$ Received IP is authentic
No $\rightarrow$ Received IP is compromised
Properties of Watermark Generated

- Minimization of embedding cost
  - A solution is generated through PSO-driven exploration which considers minimization of hardware area and latency

- Resiliency against attacks
  - Generated watermark is based on multi-variable (4 variables) signature encoding and RSA encryption therefore, it is resilient against attacks

- Fault Tolerance
  - The watermarking constraints are distributed throughout the design

- Watermark creation time and signature detection time
  - Time taken to embed a watermark is less
### Results and Analysis: Cost

**TABLE I: Comparison of proposed watermarking approach with [1]**  
(# of watermark constraint (w) = 15)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FU’s</td>
<td>Registers</td>
<td>FU’s</td>
</tr>
<tr>
<td>DWT</td>
<td>1(+)</td>
<td>3(*)</td>
<td>6</td>
</tr>
<tr>
<td>ARF</td>
<td>2(+)</td>
<td>4(*)</td>
<td>8</td>
</tr>
<tr>
<td>MPEG</td>
<td>2(+)</td>
<td>5(*)</td>
<td>14</td>
</tr>
<tr>
<td>IDCT</td>
<td>4(+)</td>
<td>2(*)</td>
<td>8</td>
</tr>
<tr>
<td>MESA</td>
<td>3(+)</td>
<td>8(*)</td>
<td>48</td>
</tr>
</tbody>
</table>
TABLE III: Measuring probability of coincidence ($P_c$) as strength of watermark

Note: $S(NW) = \#$ of storage hardware in non-watermarked solutions

<table>
<thead>
<tr>
<th>Benchmark</th>
<th># of storage variables</th>
<th>$S(NW)$</th>
<th>$P_c$ # of watermarking constraints (w)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>DWT</td>
<td>22</td>
<td>5</td>
<td>0.03</td>
</tr>
<tr>
<td>ARF</td>
<td>36</td>
<td>8</td>
<td>0.13</td>
</tr>
<tr>
<td>IDCT</td>
<td>50</td>
<td>8</td>
<td>0.13</td>
</tr>
<tr>
<td>MESA</td>
<td>139</td>
<td>48</td>
<td>0.72</td>
</tr>
<tr>
<td>MPEG</td>
<td>42</td>
<td>14</td>
<td>0.32</td>
</tr>
</tbody>
</table>

$$P_c = (1 - 1/c)^w$$

where

$P_c$ = the probability of coincidence (the probability of generating the same colored solution with the signature),

c = number of colors used,

$w = \#$ of watermarking constraints

(strength of the signature in terms of $\#$ of digits used).
Tradeoffs for a specific design

For Band Pass Filter (BPF)
Conclusion and Future Research

- A novel solution to the protection of reusable IP core through a low cost robust watermarking solution embedded during register allocation step in high level synthesis is presented.
- We plan to work on architecture-level synthesis based obfuscation technique, IP trust, process variation awareness, and fault tolerance.
References