IP Protection for FIR Filter FPGA Implementation

Wei Dai

Dept of ECE, University of Windsor

Supervisor : Dr.H.K.Kwan & Dr H.Wu

May.2004

Outline

1. Introduction to Digital Watermarking Technology for IP Protection
2. Review of IP Protection Technologies for ASIC and FPGA Design
3. Review of Watermarking Schemes for Filter Design
4. Proposed New Watermarking Methods for FPGA Implementation of FIR Filter

1. Introduction

- What is the significance of IP protection?
- What is a digital watermark?
- What is watermarking technology?
- How a digital watermark is embedded and detected?

1.1.Significance of IP Protection

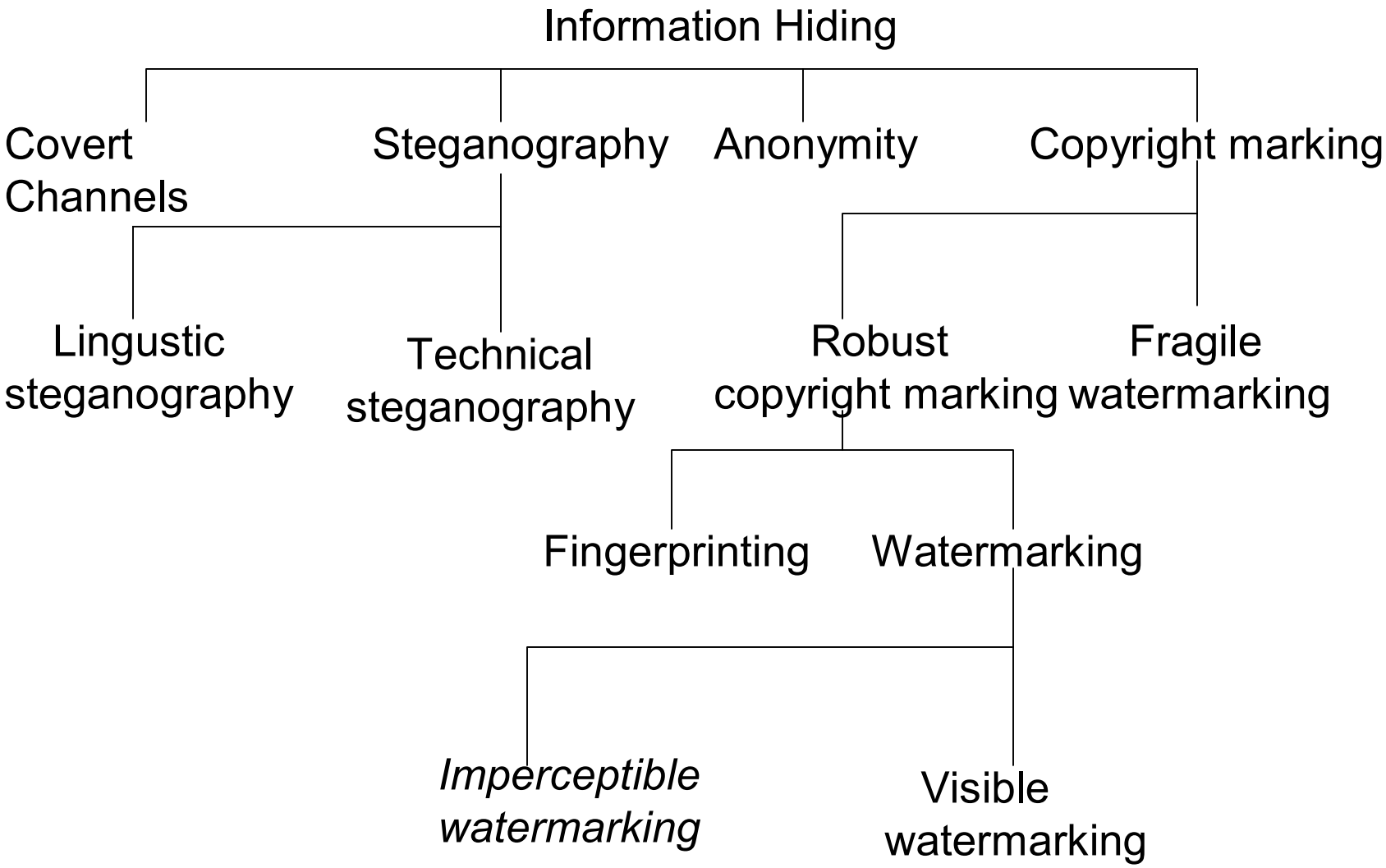
- Copyright protection is needed for various digital products
  - Digital still images
  - Digital audio products (i.e., music CD)
  - Digital video products (i.e., DVD movies)
  - Software source code
  - ASIC and FPGA IP cores
  - Electronic publications

1.2.What is a Digital Watermark and Watermarking Technology?

- A **digital watermark** is a piece of information hidden inside the IP that shows IP owner's identity.

- **Watermarking** refers to the technology to embed a digital watermark into a digital IP and to detect it from a digital IP with watermark.

Information Hiding and Watermarking

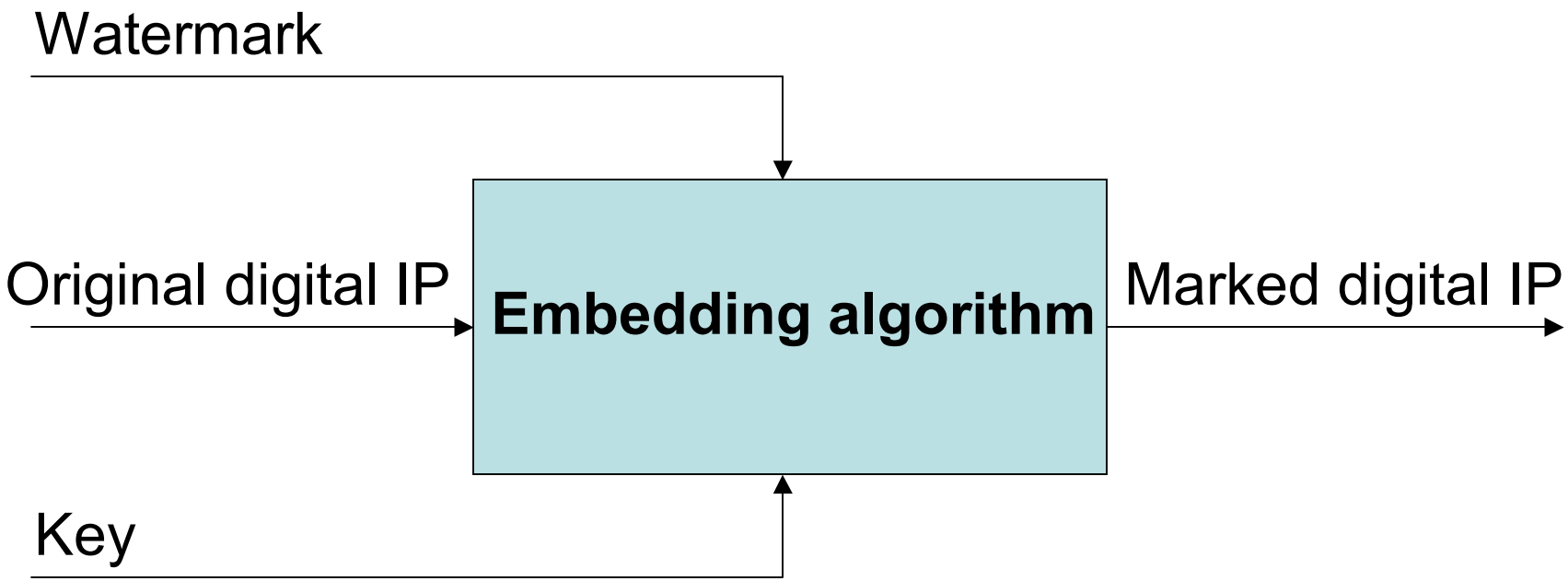


- Criteria for a watermarking technology
  - Easy to embed and detect a watermark
  - Difficult to remove the watermark
  - Low design and implementation overhead or cost

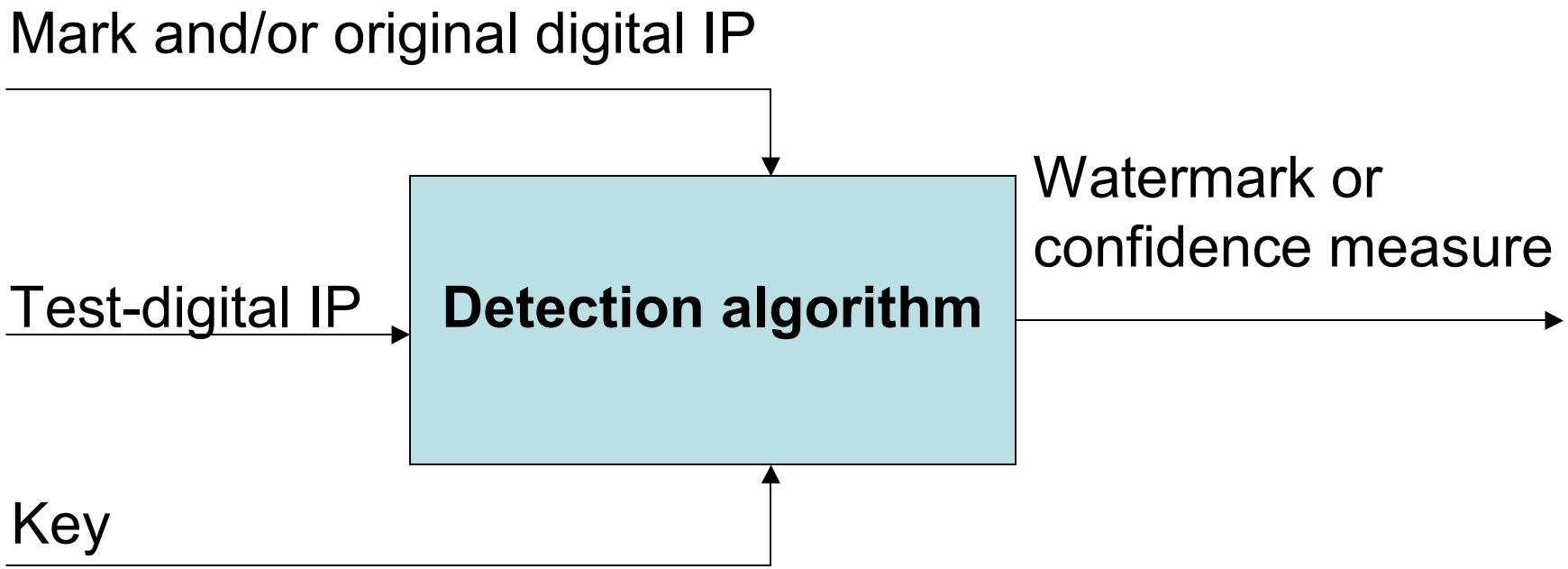
- Watermarking technology has applications for
  - Copyright protection
  - Fingerprinting
  - Data authentication
  - Data hiding

1.3. Watermark Embedding and Detection:

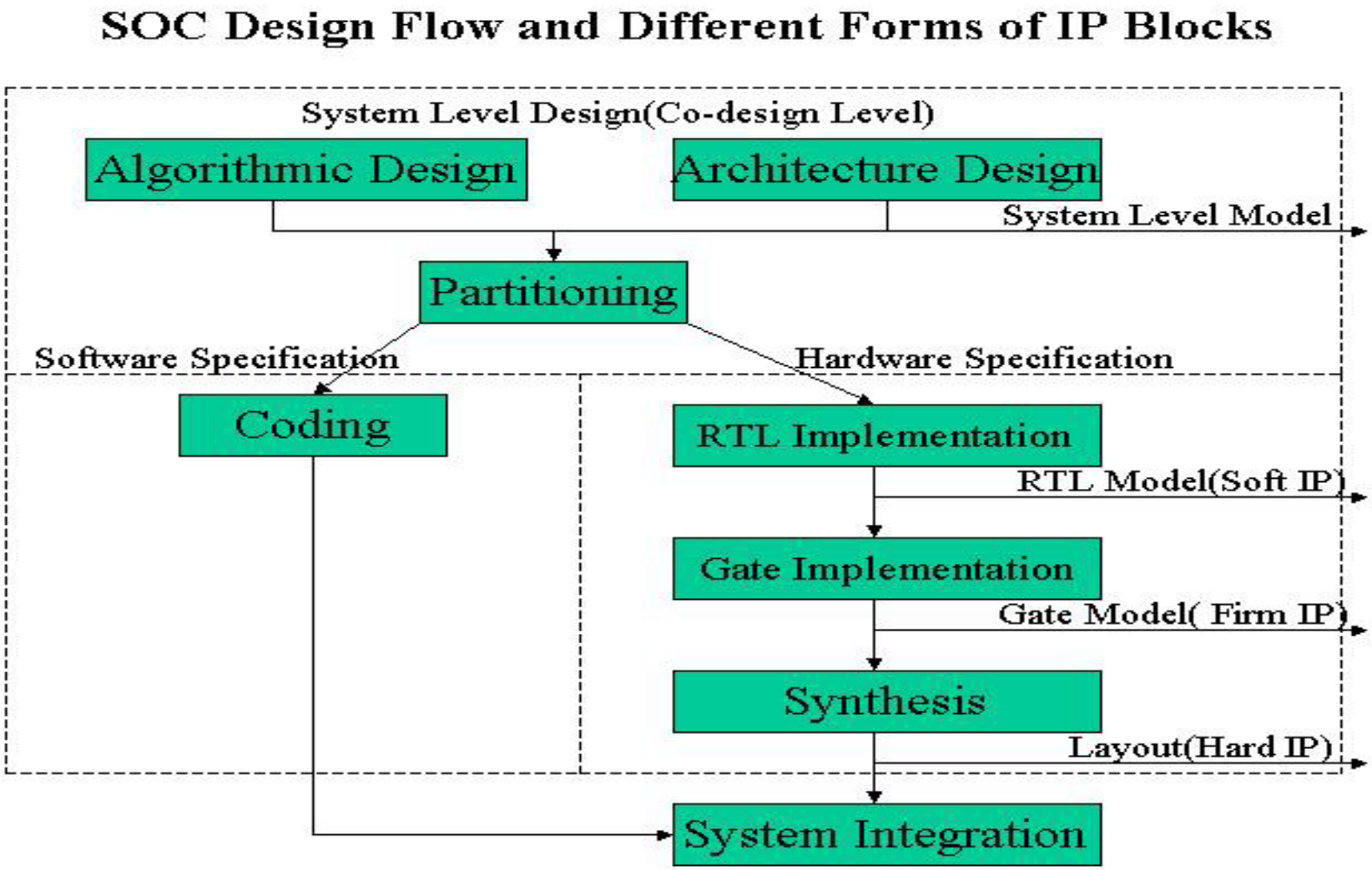
Scheme for Embedding a Watermark



Scheme for Watermark Detection



2. IP Protection for ASIC and FPGA Designs



2.1. Non-watermarking methods for ASIC and FPGA design IP protection:

- IP owners try to protect their copyright by *encrypting* VHDL/Verilog source codes before sending them to the users.
- The encrypted codes are loaded into *authorized simulators* or synthesis tools.
- The source codes are *invisible* to the system designer who uses the IP blocks.

Non-watermarking methods are <u>not</u> secure enough:

- The *CAD tool* maintains the safety of the copyright.
- In practice, this can often be broken by attacking the CAD tool (simulators, synthesis tool) directly.
- *Better* methods are needed for protection of the ASIC and FPGA IP.

2.2. Watermarking methods for VLSI/FPGA

- Advantages of watermarking methods
  - VLSI/FPGA IP watermarking can help *deter theft* and *counterfeiting*.
  - The embedded watermark serves as *evidence of ownership*.

- A watermarking method includes two phases
  - Watermark synthesis
  - Watermark detection

FPGA and ASIC Watermarking Schemes

1. **Fingerprinting for FPGA IP protection**
   - This method uses the unused LUT (look-up table) bits to embed the signature bits.

2. **Change of FPGA bit-stream data**
   - The method substitutes watermark bits for some of the bits in the configuration bit-stream that controls multiplexers for the unused CLB outputs.

3. **Hierarchical watermarking**
   - It uses an unique mapping of *topological* information onto a sequence of symbols.
4. **Protocols for IP protection**
   - Hide watermark data at the combinational logic synthesis level.
5. **Finger-marking**
   - Hide watermark at the layout level by modifying the transistor W/L and the number of fingers.

3. Review of Filter Design Watermarking

- Three technologies have been proposed:
  - Magnitude modification
  - Filter tap's equal-replacement
  - Windowing function watermarking

3.1.Magnitude Modification

- Step 1: Prepare watermark code (i.e., 7-bits)
- Step 2: Separate the filter stop or pass-band to several equal width zones (i.e., seven zones)
- Step 3: Modify the filter magnitude response according to the watermark bits:
  - If the bit is **1**,**decrease** the filter magnitude response by **1 dB**.
  - If the bit is **0,increase** the filter magnitude response by **1 dB**.

3.1.Magnitude Modification (Con't)

– Step 4: Use the modified filter magnitude response as the design constrains input to the design tool.

– Step 5: Obtain the filter coefficients.

Magnitude response of the filter with watermarking

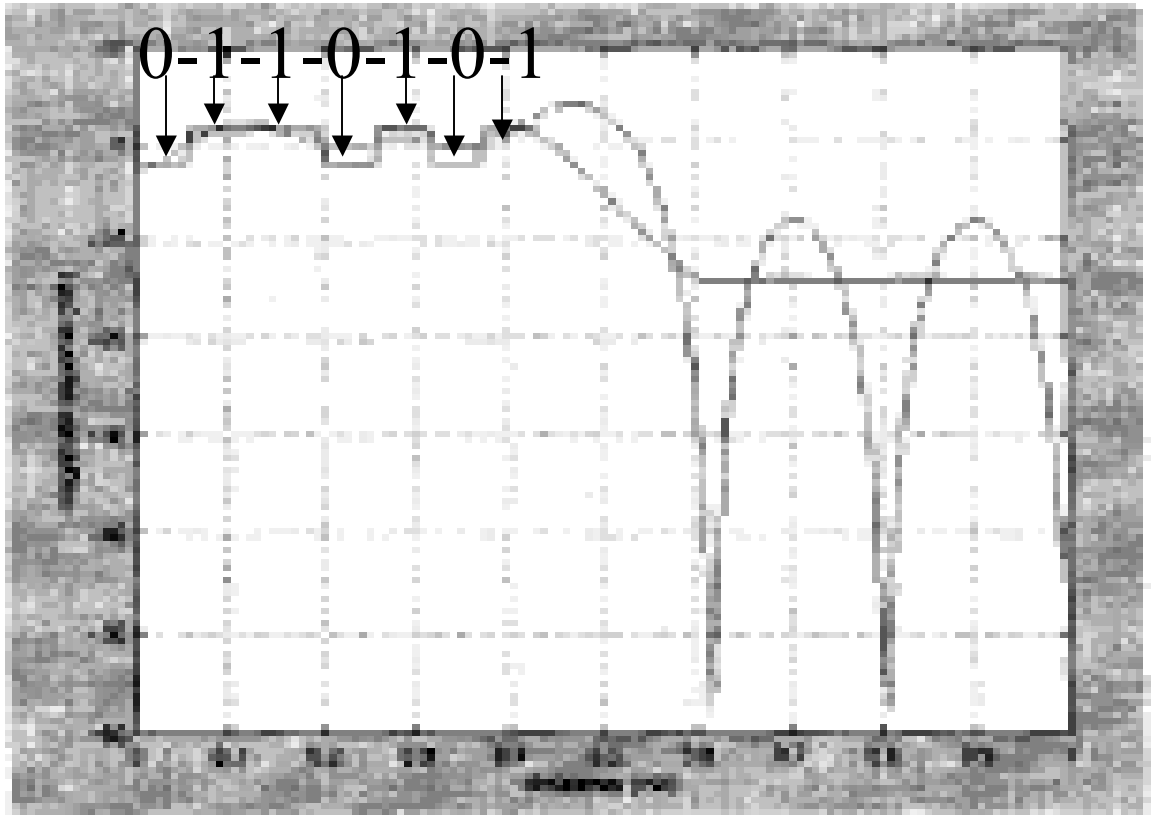


Fig. 2 Magnitude response of the filter specified in Fig. 1

Advantages of this method:

- Add watermark at *algorithm* level, the highest level for filter design
- Hard to remove from lower level of filter's implementation, like logic, layout or circuit level

Disadvantages of this method:

- Increase the *complexity* of filter design.
- Increase the *order* of filter, since we introduce new ripple constrain to the filter magnitude response
- Increase the *hardware cost* by +7%

3.2.Filter watermarking by filter tap's equal-replacement

- Procedure:
  - Step 1: Prepare the watermark (7-bits).
  - Step 2: Design the filter with the original performance specification.
  - Step 3: Replace the filter taps by using equal filter structure replacement.
    - There are three equal function filter structures, **A, B** and **C**.
    - When the watermarking bit is **0**,use **B** to implement this tap.
    - When the watermarking bit is **1**,use **C** to implement this tap.

Equal-replacement filter watermarking

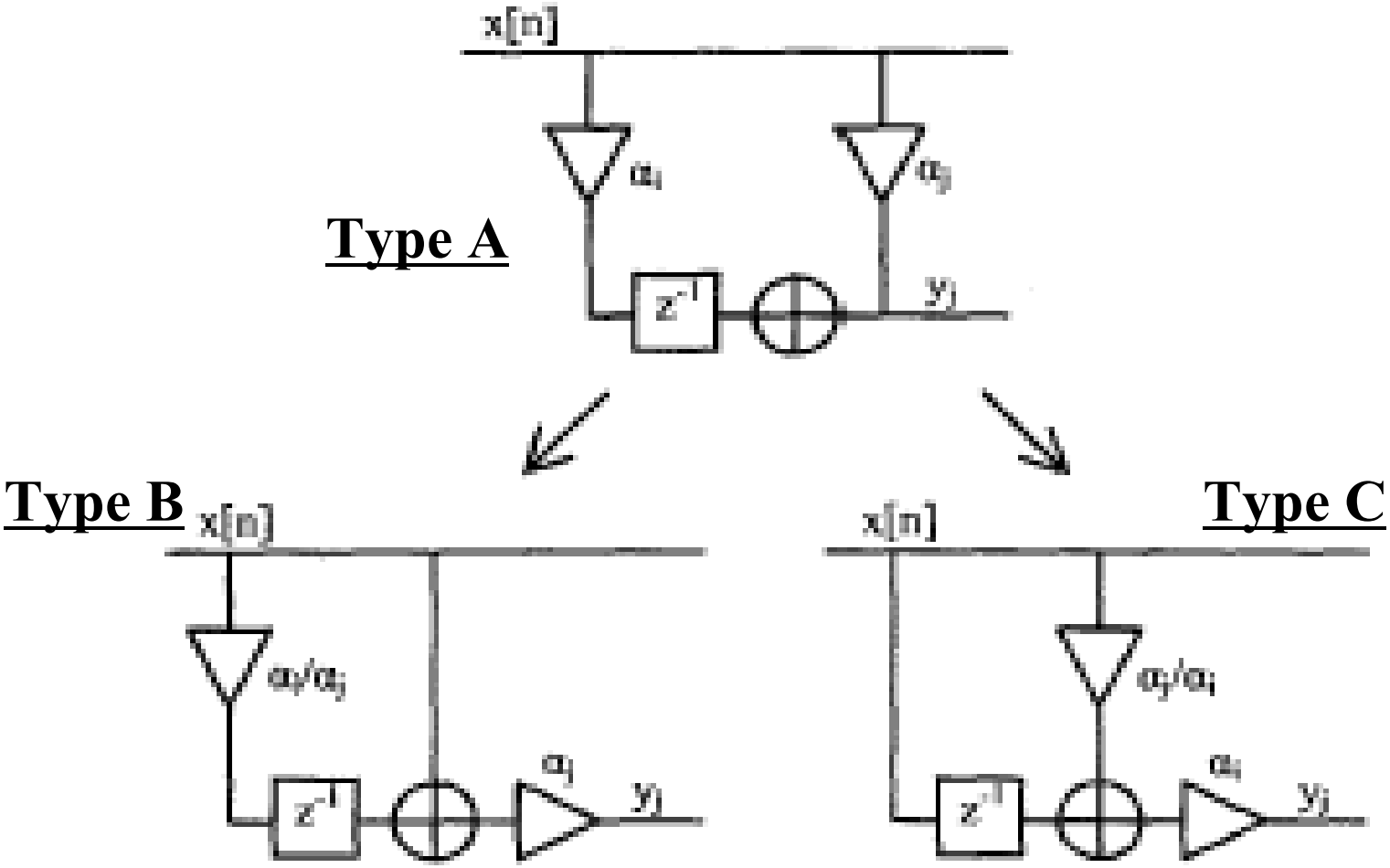


Fig. 3 Architecture level circuit transformations

Advantages of this method:

- Watermarking at *algorithm* level, hard to remove at the lower levels
- No degradation of filter performance

Disadvantages of this method:

- Increase hardware cost dramatically by +33%
- Make the filter structure not uniform, and increase the design time for implementation with ASIC or FPGA

3.3.Windowing function watermarking

- Procedure:
  - Step 1:
    - Suppose W(n) is the original window function, where 1<= n <=N.
    - Add random noise to W(n) to obtain Wm(n)

      $Wm(n)=W(n)+a * r(n), 1<=n<=N,$
    - Select a as 0.001
    - r(n) is a random sequence with zero mean

– <u>Step 2</u>:

$$Wc(n) = \begin{cases} Wm(n), & 1 <= n <= i-1 \\ Wm(n)+b*c(n-i+1), & i <= n <= i+P-1 \\ Wm(n), & i+P <= n <= N/2 \\ Wc(N+1-n), & N/2+1 <= n <= N. \end{cases}$$

- Select b=0.0001.
- The starting bit of the P-bit watermark code sequence c(n-i+1) is bit i of Wm(n).
- The sequence Wc(n), n=1,…,N, is the new window function which contains the watermark information.

- Advantages of this method:
  - Embedding watermark at algorithm level which is hard to remove at a lower lever
  - Simple and direct watermark embedding scheme

- Disadvantages of this method:
  - Increase design complexity

4. New Proposals for FIR Filter Watermarking with FPGA Implementation

- **Proposal one:**
  - Embedding watermark at FIR filter coefficients' LSB
- **Proposal two:**
  - FPGA RAM cell locations' watermarking

4.1.Proposal One: FIR Filter Coefficients' LSB Watermarking

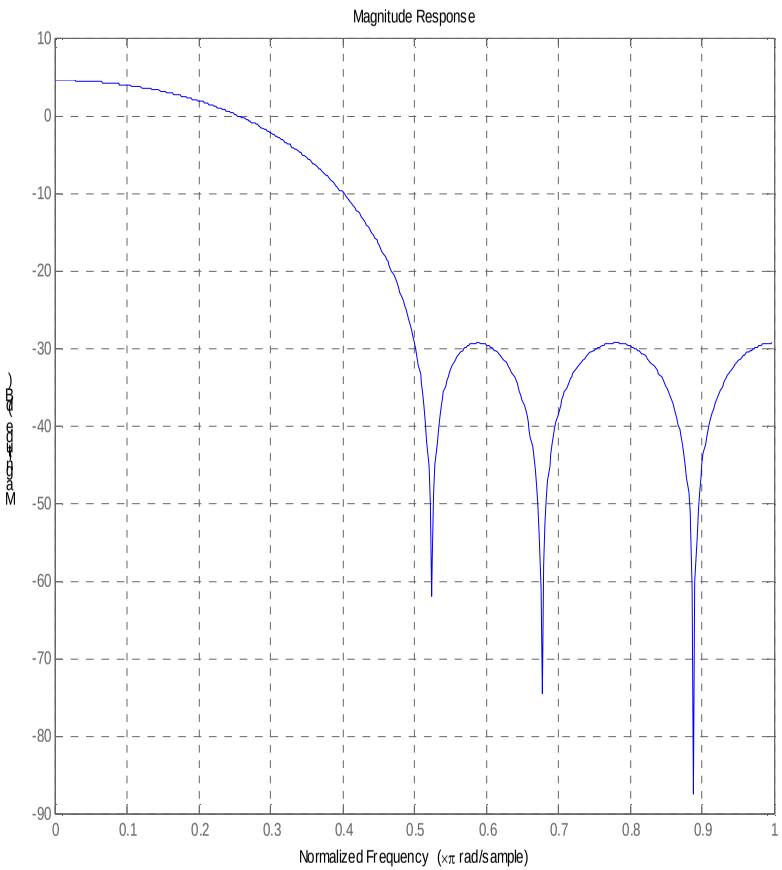
– Let a FIR filter design be given by

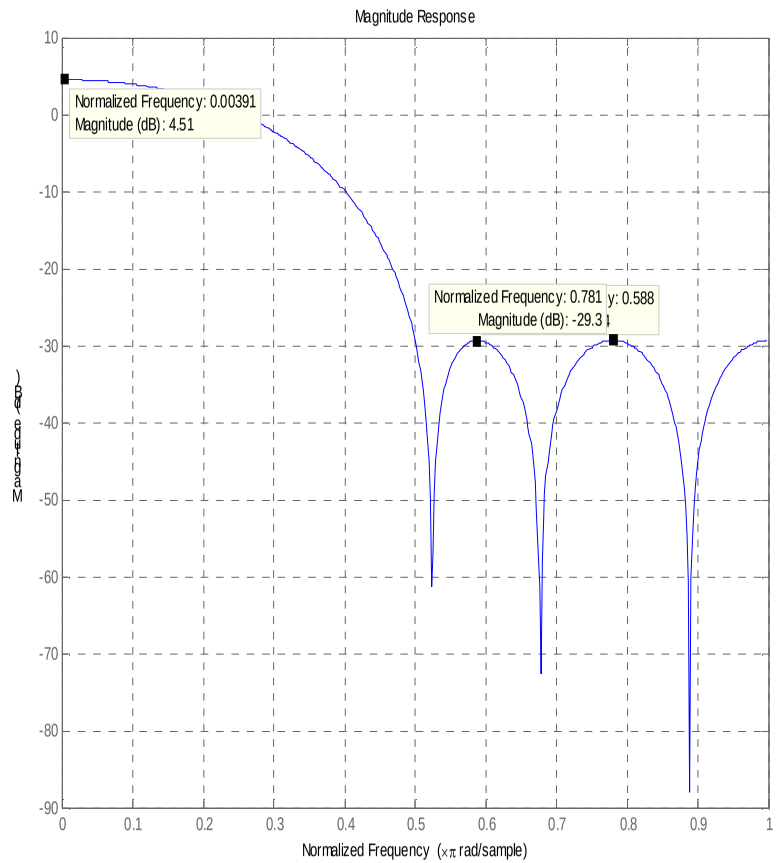
Y(k)=A0*x(k)+A1*x(k-1)+……+An-1*x(k-N-1),
where k=0,1,……,N-1.
A0,A1,……An-1 are filter coefficients.

– A watermark, for instance, is given by
10001010
– Watermark embedding process is to replace the filter coefficients' LSBs with the watermark bits.

Original Filter Coefficients	Modified Filter Coefficients	Watermark bits
A0:1111-1111-1100-0100	1111-1111-1100-010**1**	**1**
A1:1111-1111-0100-1101	1111-1111-0100-110**0**	**0**
A2:1111-1111-1010-1001	1111-1111-1010-100**0**	**0**
A3:0000-0000-1110-1010	0000-0000-1110-101**0**	**0**
A4:0000-0011-0111-1000	0000-0011-0111-100**1**	**1**
A5:0000-1000-1101-1111	0000-1000-1101-111**0**	**0**
A6:0000-0001-1111-1010	0000-0001-1111-101**1**	**1**
A7:1111-1110-0101-0110	1111-1110-0101-011**0**	**0**

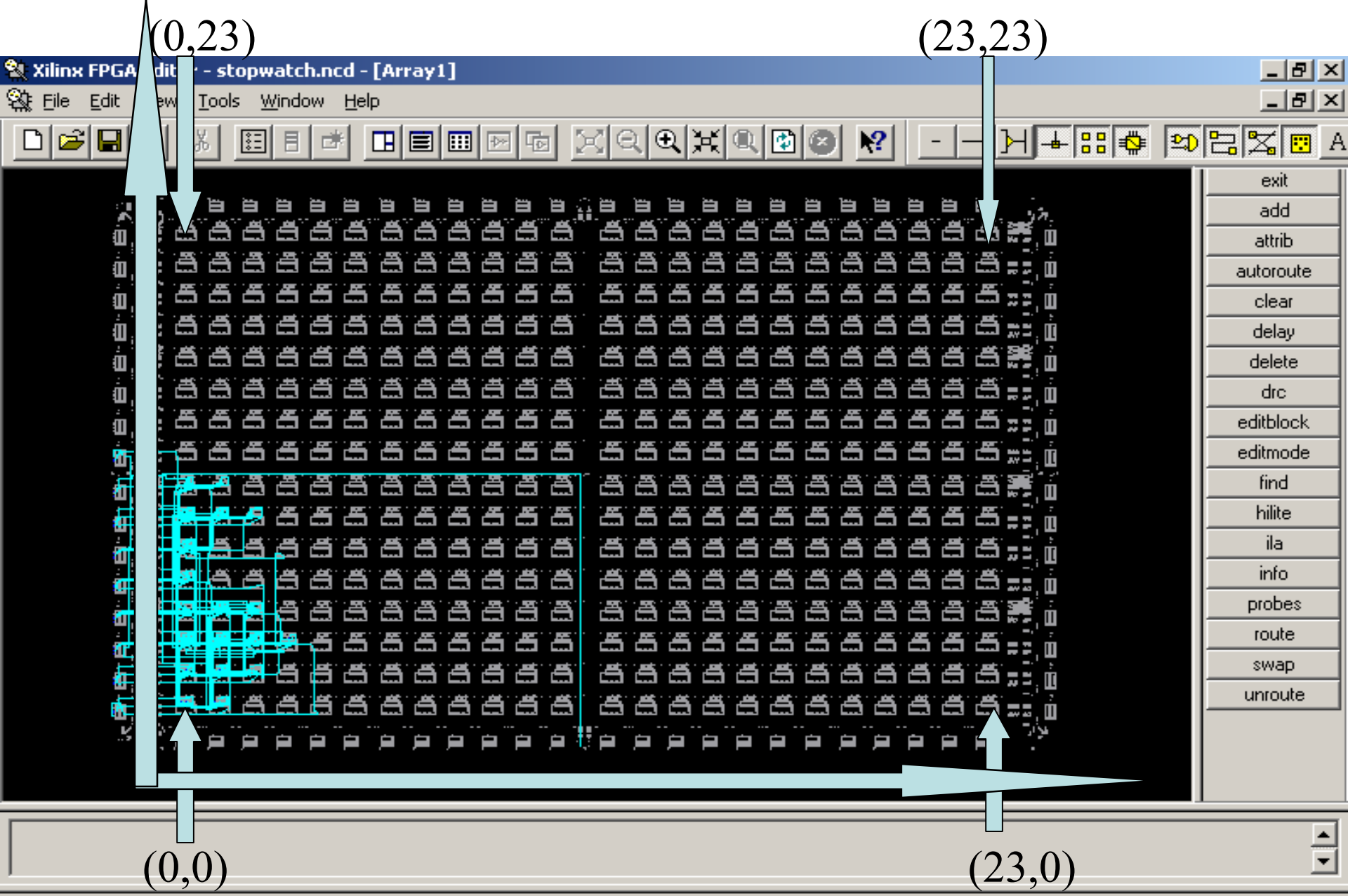
Magnitude response simulation results: FIR filter coefficients' LSB watermarking (left: before watermarking right: after watermarking)





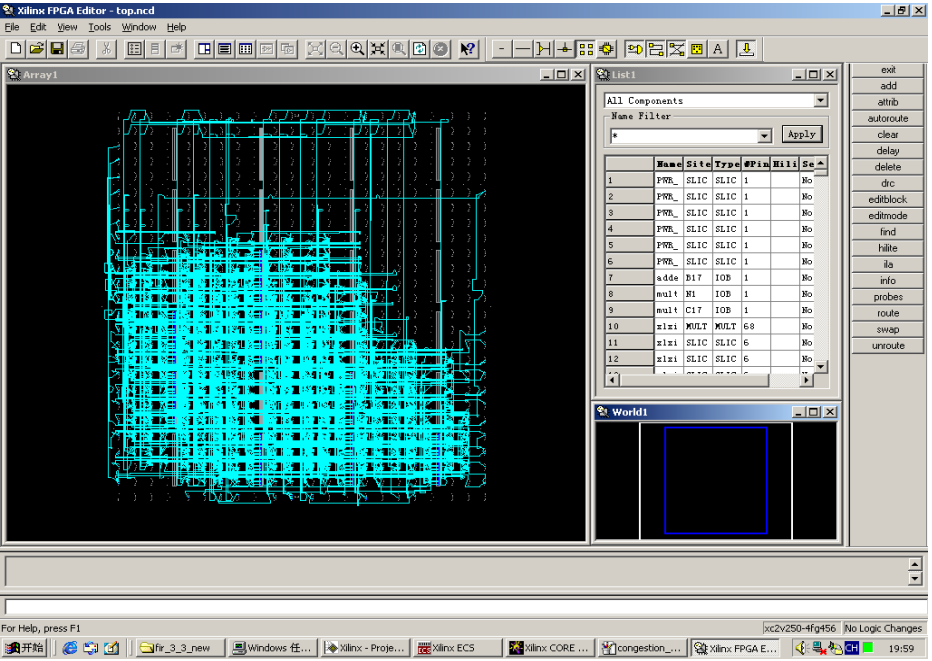
4.2.Proposal Two: FPGA RAM Cell Locations' Watermarking

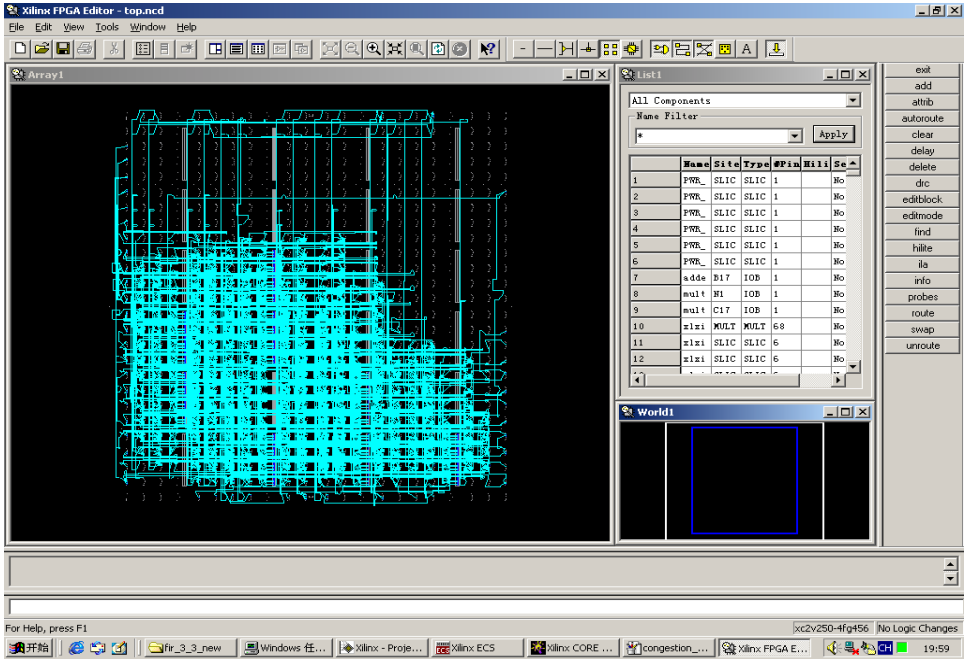
- Let a FIR filter design be given by
  Y(k)=A0*X(k)+A1*X(k-1)+……+An-1*X(k-N-1),
  where k=0,1,….N-1.
- Suppose a watermark consists of 8 bits as
  10001010.
- Let the RAM cell to store Ai be RAM_i, i=0,1,…,7.
- A RAM cell on FPGA circuits can be identified by its coordinates (a, b).
- Watermark bits embedding rule:
  - If the bits are **00**, we choose **both** a and b as **even**.
  - If the bits are **01**, we choose **a as even** and **b as odd**.
  - If the bits are **10**, we choose **a as odd** and **b as even**.
  - If the bits are **11**, we choose **both** a and b as **odd**.


(0,23)
(23,23)
Xilinx FPGA Editor - stopwatch.ncd - [Array1]
File Edit View Tools Window Help
exit
add
attrib
autoroute
clear
delay
delete
drc
editblock
editmode
find
hilite
ila
info
probes
route
swap
unroute
(0,0)
(23,0)
For Help, press F1
xcv50-6bg256
No Logic Changes
开始
雅虎通
Microsoft ...
Acrobat R...
Xilinx - Pr...
Xilinx FPG...
CH
19:48

RAM_i for storing A_i	Watermarking bits	New Location of the chosen RAM cell
RAM_0	1	(1, 4)
	0	
RAM_1	0	(4,4)
	0	
RAM_2	1	(3,2)
	0	
RAM_3	1	(3,0)
	0	

FIR filter place and route results:
left:before watermarking
right:after watermarking





	Max Frequency	Hardware cost (slice)
Before watermarking	**107.654 MHz**	**488/1536**
After watermarking	**106.474 MHz**	**489/1536**

Advantages of the proposed methods:

- Simple watermark embedding and extraction
- Watermark is invisible to the potential attacker.
- Embedding watermark at both algorithm and layout level
- Increasing secure strength and the time for reverse engineering

Disadvantages of the proposed methods:

- For the first proposed method, filter performance may slightly worsen due to the modification of filter coefficients' LSB.
- For the second proposal, max frequency and hardware usage may be slightly worse than the original design.

Comparison of Proposal One with Current Methods (at algorithm level)

	Magnitude Modification	Tap's Equal Replacement	Windowing Function Watermarking	Proposal One (FIR Filter LSB Watermarking)
Filter Performance Degradation	Medium	Small	Small	Small
Hardware Usage Increase	+7%	+29%	N/A	0%
Design Overhead	High	Medium	Low	Low
Extraction Cost	Low	Medium	Low	Medium
Probability of Coincidence	Low	Low	Low	Low
Security	Medium	Medium	Low	Medium

Comparison of Proposal Two with Current Methods (at physical/RTL design level)

	Using Spare LUT	Bit-stream Modifica-tion	Hierarchical Watermark-ing	Watermarking by Using Protocols	Finger -mark-ing	Proposal Two
Add Level	Layout	Layout	RTL	RTL	Layout	Layout
Embedding Cost	Medium	Medium	Medium	Medium	Med	Med
Design Overhead	Medium	Medium	Medium	Medium	High	Low
Extraction Cost	Medium	High	High	High	Low	Med
Probability of Coincidence	Low	Low	Low	Low	Mediu m	Low
Security	Low	High	High	Medium	Low	High
Applied Area	FPGA	FPGA	ASIC (digital)	ASIC (digital)	ASIC (mix-signal)	FPGA

- *Thanks for your time !*
- *Any questions & advice?*