



# Hardware Security/Protection of CE Devices: Threat Models and Defense against IP Piracy and IP Trojan

**Dr. Anirban Sengupta, Prof.**

Distinguished lecturer, IEEE CE Society

Technical Chair, IEEE ICCE '18

Senior Editor, IEEE Consumer Electronics (CEM), CE Society

Computer Science and Engineering

Indian Institute of Technology Indore, India

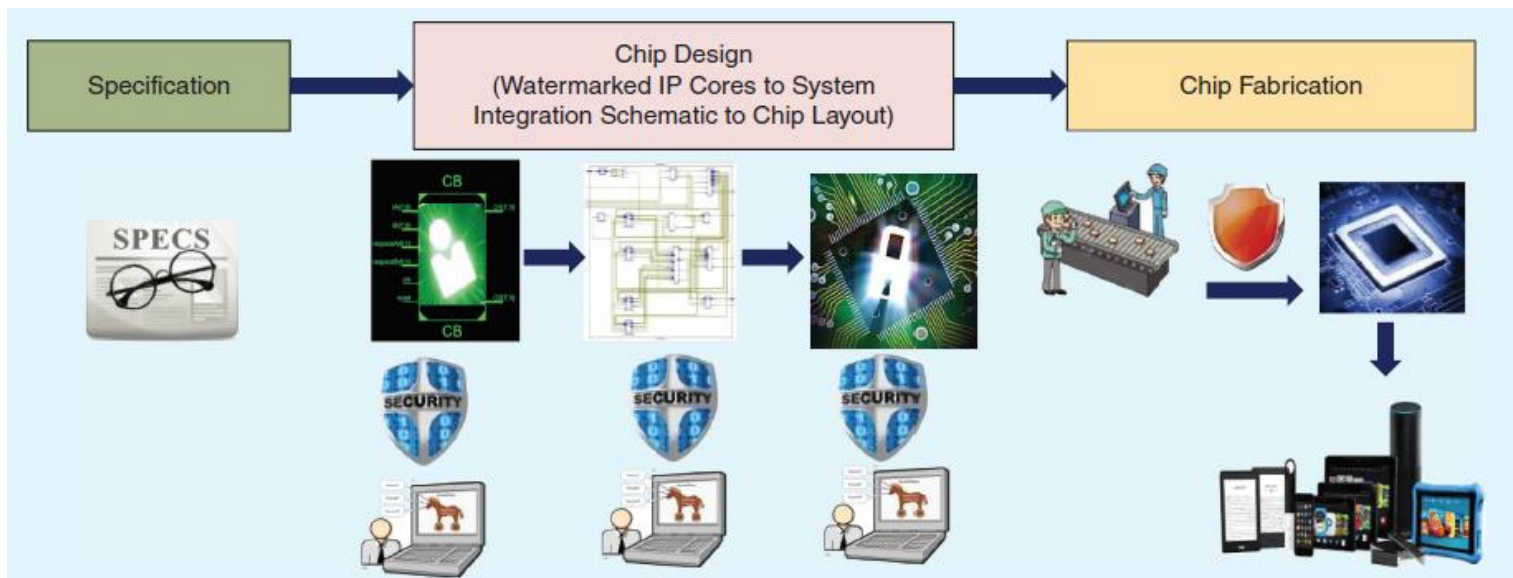
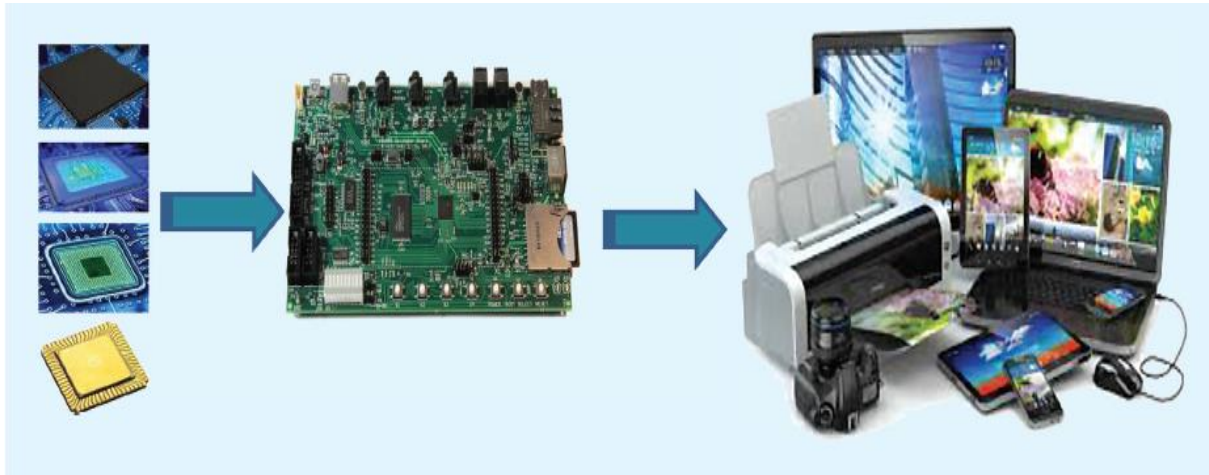
Email: [asengupt@iiti.ac.in](mailto:asengupt@iiti.ac.in)

Web: [www.iiti.ac.in/~asengupt](http://www.iiti.ac.in/~asengupt)

# My key Journal Contributions in CE Device Hardware Security

1. A. Sengupta et. al "Triple-Phase Watermarking for Reusable IP Core Protection during Architecture Synthesis", [IEEE Transactions on Computer Aided Design of Integrated Circuits & Systems \(TCAD\)](#), 2017
2. A. Sengupta et. al "Securing IoT Hardware: Threat models and Reliable, Low-power Design Solutions", [IEEE Transactions on Very Large Scale Integration \(VLSI\) Systems](#), 2017
3. A. Sengupta et. al "TL-HLS: Methodology for Low Cost Hardware Trojan Security Aware Scheduling with Optimal Loop Unrolling Factor during High Level Synthesis", [IEEE Transactions on Computer Aided Design of Integrated Circuits & Systems \(TCAD\)](#), 2016
4. A. Sengupta et. al "Exploring Low Cost Optimal Watermark for Reusable IP Cores during High Level Synthesis" , [IEEE Access Journal](#), 2016
5. A. Sengupta et. al "Protecting an Intellectual Property Core during Architectural Synthesis using High-Level Transformation Based Obfuscation", [IET Electronics Letters](#), 2017
6. A. Sengupta et. al "IP core Protection of CDFGs using Robust Watermarking during Behavioral Synthesis Based on User Resource Constraint and Loop Unrolling Factor", [IET Electronics Letters](#), 2016
7. A. Sengupta et. al "Low Cost Security Aware High Level Synthesis Methodology", [IET Journal on Computers & Digital Techniques \(CDT\)](#), 2016
8. A. Sengupta "Protection of IP-Core Designs for CE Products", [IEEE Consumer Electronics Magazine](#), 2015
9. A. Sengupta "Hardware Vulnerabilities and its Effect on CE Devices: Design-for-Security against Trojan", [IEEE Consumer Electronics Magazine](#), 2017
10. A. Sengupta et. al "Anti-Piracy aware IP Chipset Design for CE Devices: Robust Watermarking Approach", [IEEE Consumer Electronics Magazine](#), 2017.
11. A. Sengupta "Hardware Security of CE Devices: Threat Models and Defence against IP Trojans and IP Piracy", [IEEE Consumer Electronics Magazine](#), 2017
12. A. Sengupta et. al "Forensic Engineering for Resolving Ownership Problem of Reusable IP Core generated during High Level Synthesis", [Elsevier Journal on Future Generation Computer Systems](#), Aug 2018
13. A. Sengupta et. al "Low Overhead Symmetrical Protection of Reusable IP Core using Robust Fingerprinting and Watermarking during High Level Synthesis". [Elsevier Journal on Future Generation Computer Systems](#). 2017
14. A. Sengupta et. al "Automated Low Cost Scheduling Driven Watermarking Methodology for Modern CAD High-Level Synthesis Tools" [Elsevier Journal of Advances in Engineering Software](#), 2017
15. A. Sengupta et. al Low cost optimized Trojan secured schedule at behavioral level for single & Nested loop control data flow graphs, [Elsevier VLSI Integration](#), 2016
16. A. Sengupta et. al "Security and Reliability Aware System Design for Mobile Computing Systems", [IEEE Access Journal](#), 2016

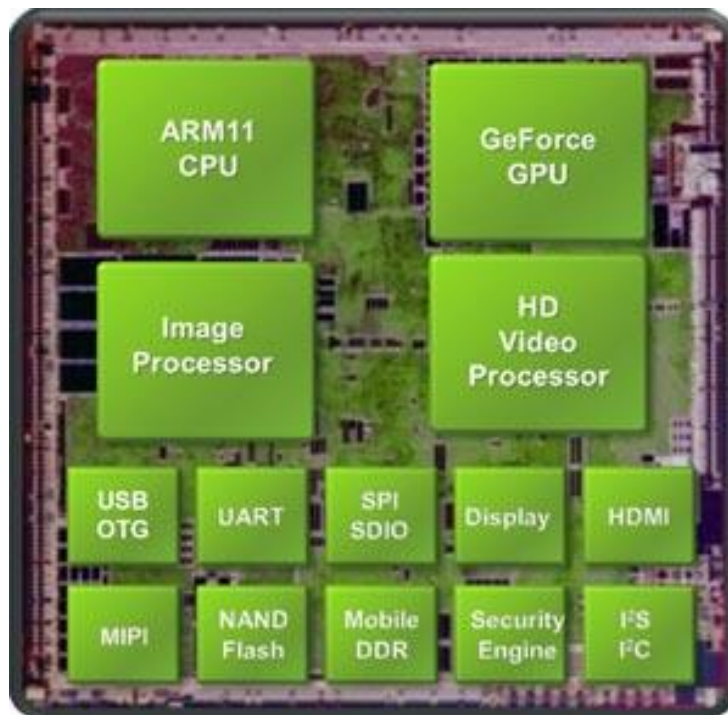
# CE Device Vulnerabilities



**A. Sengupta et. al** "Hardware Vulnerabilities and its Effect on CE Devices: Design-for-Security against Trojan", [IEEE Consumer Electronics Magazine](#), 2017

# Intellectual Property (IP) Core ...

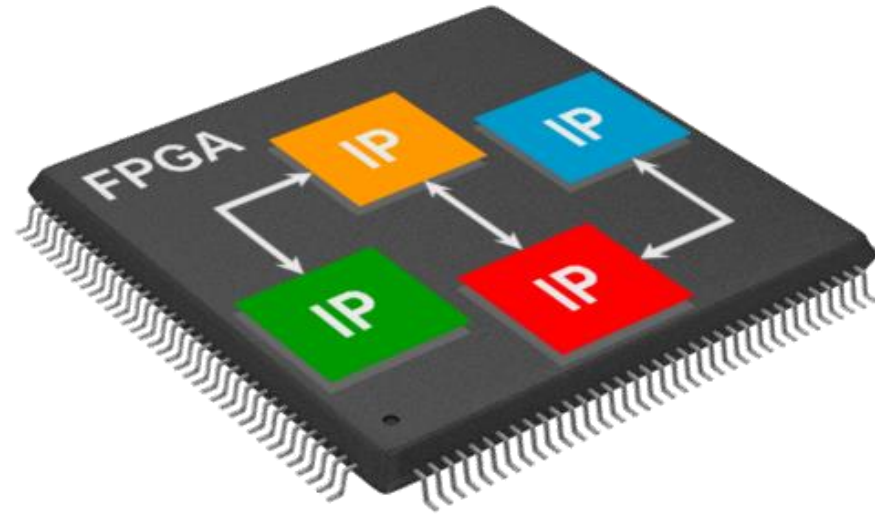
- Consumer Electronics is realized as SoC for low-power, low-cost and high performance requirements.
- Consumer Electronics SoC design challenges include:
  - Lower Cost, Lower Design Cost, and Shorter Time-to-Market



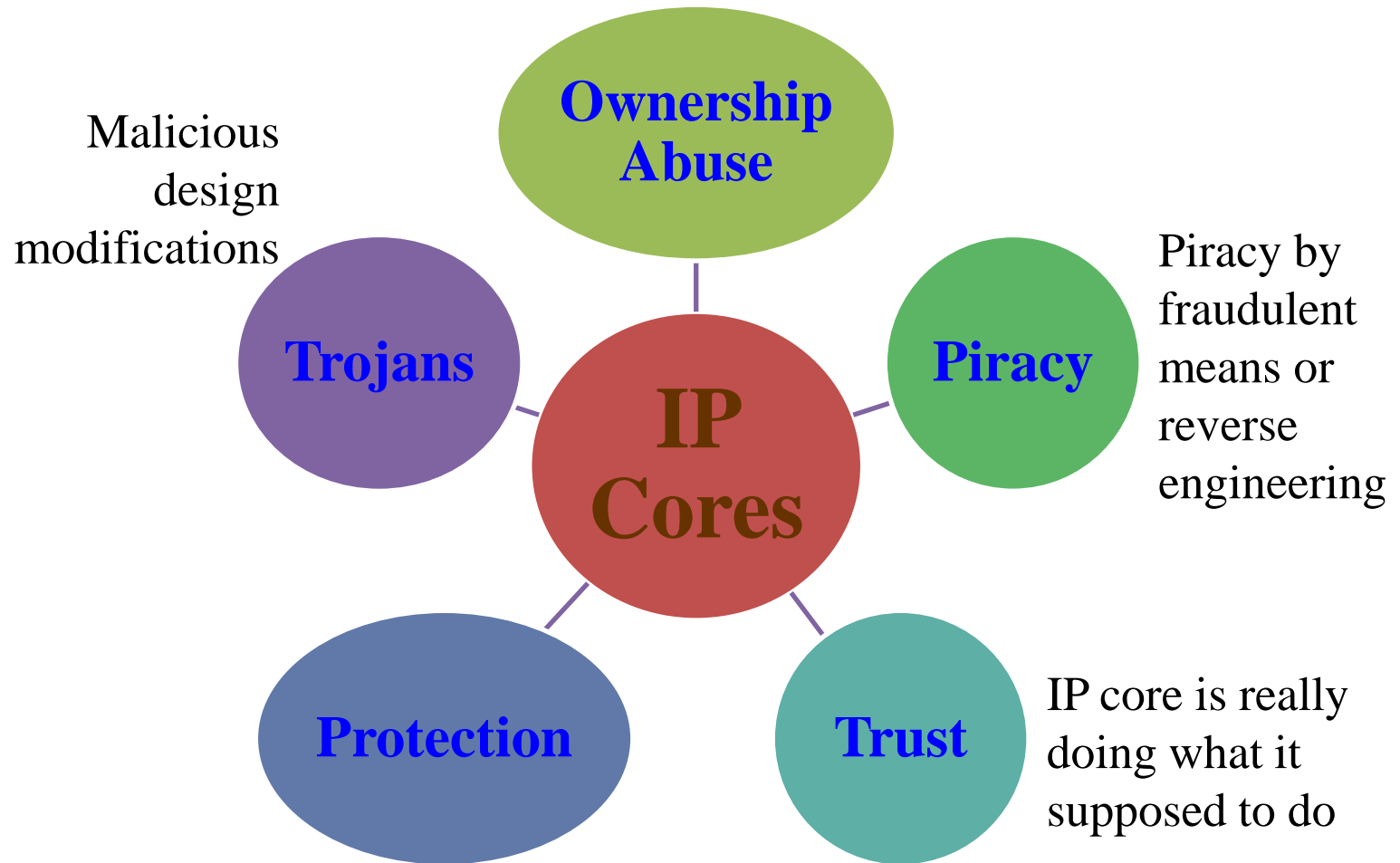
- IP cores based system design is used to meet the challenges
- IP cores (often supplied by third party vendors)
  - Maximize design productivity, minimize design time

# Intellectual Property (IP) Core

- 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.



# IP Core – Selected Issues/Challenges



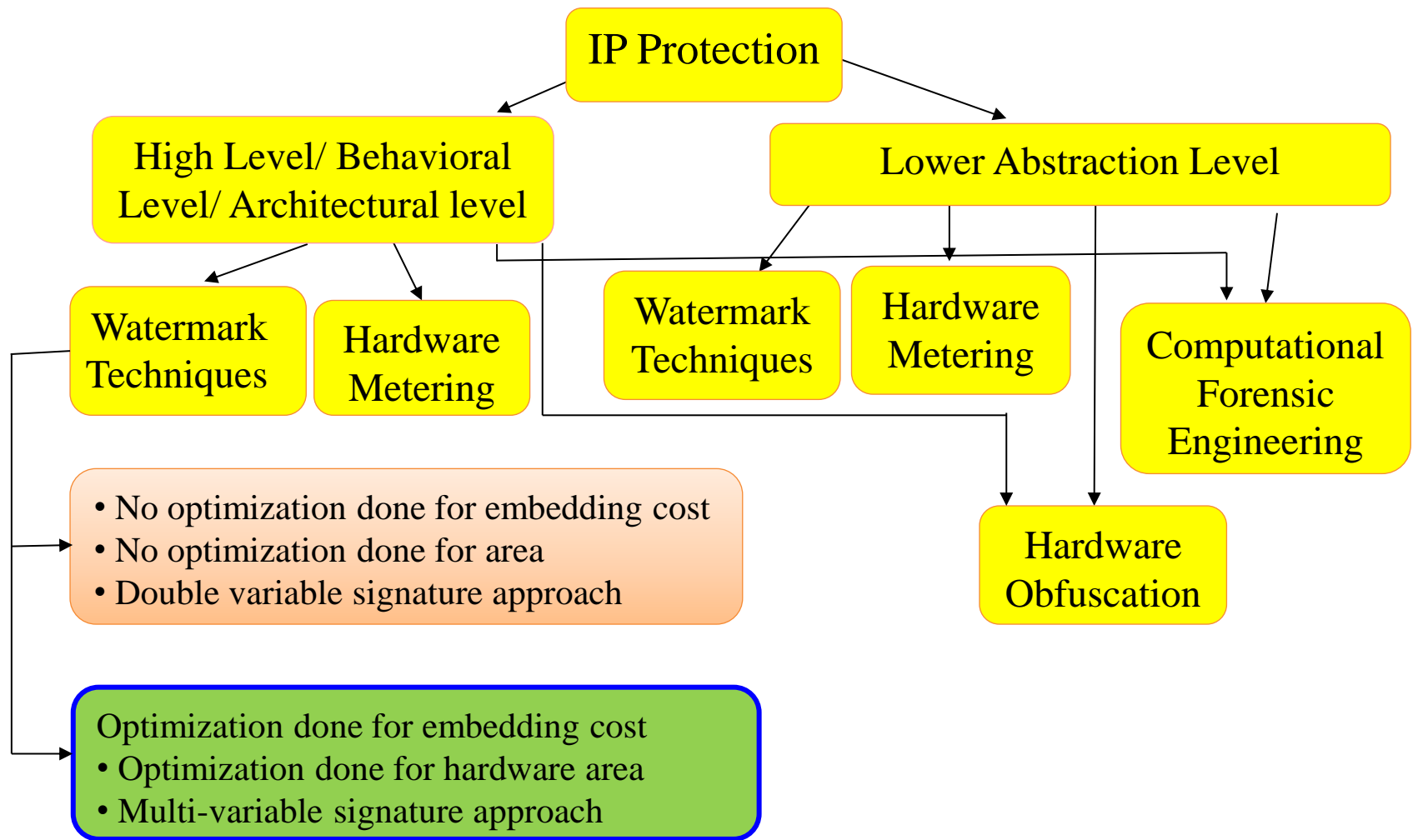
# IP Threat Models: Type 1

	3PIP Vendor	SoC Integrator/Buyer	Foundry	Security
Scenario 1	Watermark	Attacker	---	Vendor ownership
Scenario 2	Watermark	---	Attacker	Vendor ownership
Scenario 3	Attacker	Fingerprint	---	Buyer ownership

Typical attacks related to IP Piracy



# IP Protection





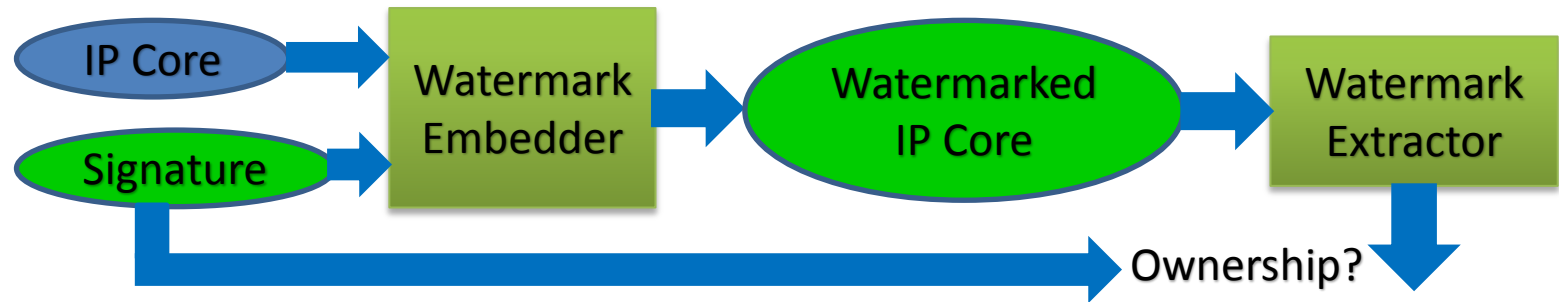
# Solution of IP protection - Watermarking



- Watermarking has 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.
- 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 for an entity who has full knowledge of the signature encoding rule

# Properties of Watermark Generated

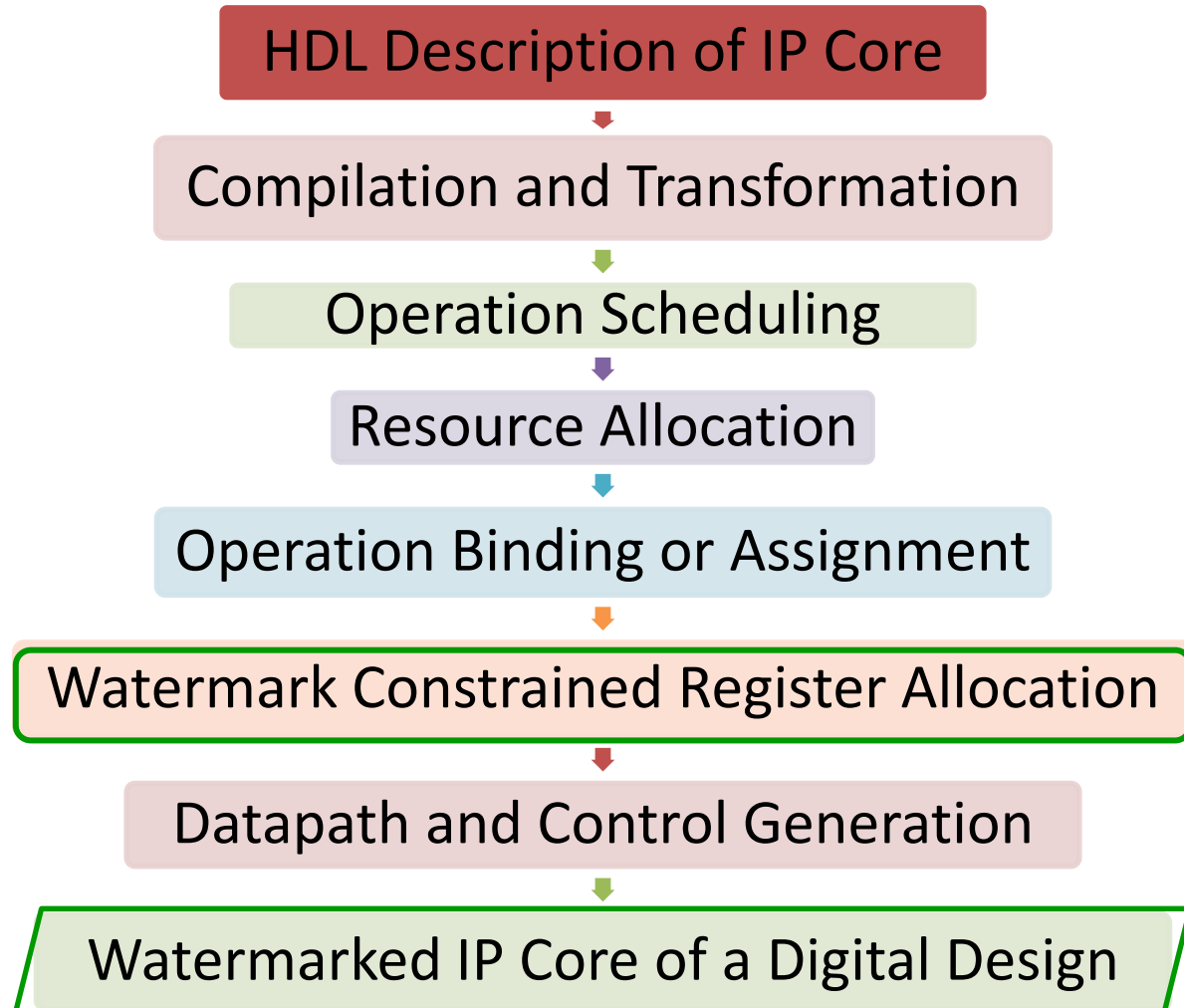
- Minimization of embedding cost
  - A solution is generated through PSO-driven exploration which considers minimization of hardware area and latency
- Resiliency against attacks
  - Generated watermark is based on multi-variable (4 variables) signature encoding therefore, it is resilient against attacks
- 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

# 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 [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.
- 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, a cost optimal watermark based on robust multi-variable signature encoding during HLS for reusable IP core protection is presented.

# High-Level Synthesis Flow for IP Protection

## – A Simplified View



# Watermarking ...

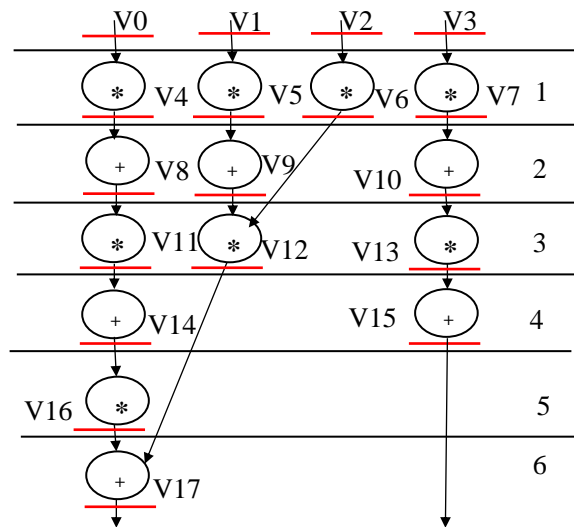
## Process for embedding watermark in the design

- 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  $(i, l, T, !)$ . Each variable of the generated signature maps onto a certain edge pair:
  - $i$  = encoded value of edge with node pair as (prime, prime)
  - $l$  = 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)

# Watermarking ...

## Process for embedding watermark in the design (contd..)

- 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.



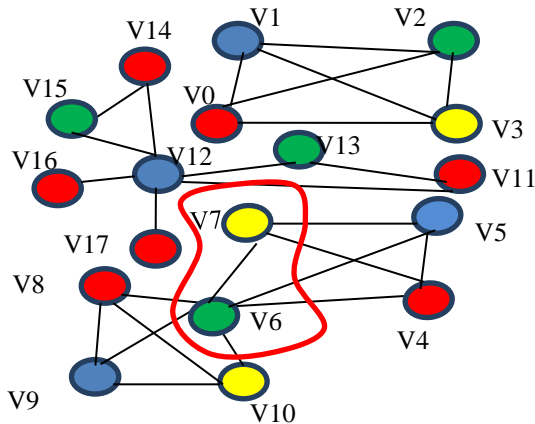
Scheduling of a CDFG with 3  
adders and 4 multipliers

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

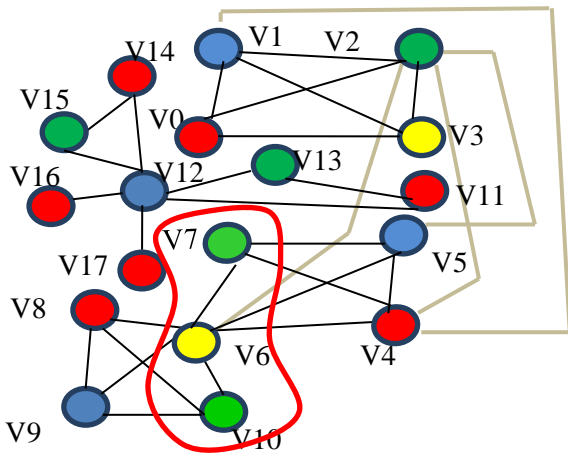
Controller for register allocation before  
embedding watermark



# Embedding Watermark in IP core....



Colored Interval Graph for the scheduling  
(before watermark)



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)
<b>I</b>	<b>(2, 6)</b>
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	v6	v7
2	v8	v9	v6	v10
3	v11	v12	v13	--
4	v14	v12	v15	--
5	v16	v12	v15	--
6	v17	--	v15	--

Controller for register allocation before watermark

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	--

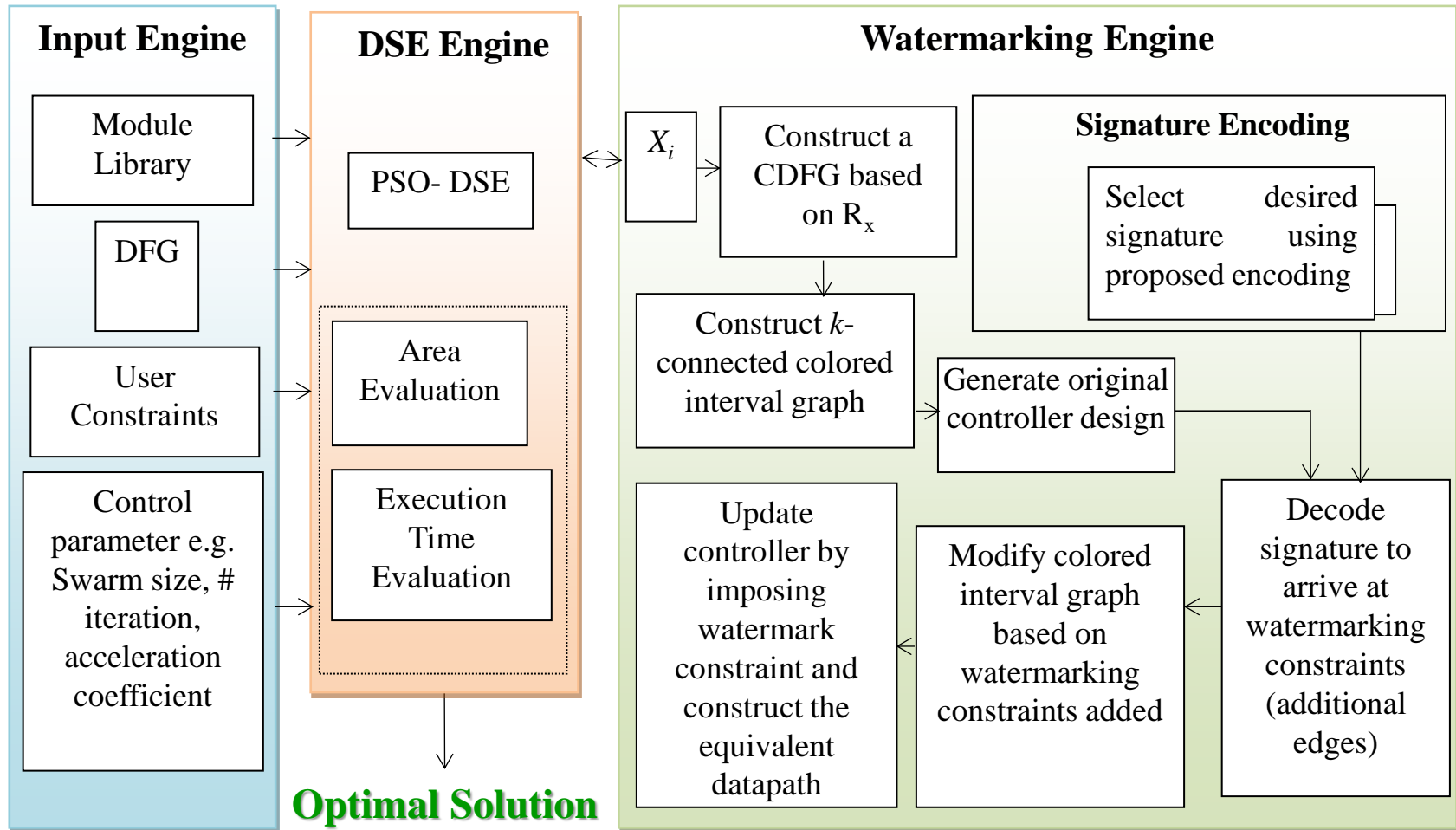
Controller for register allocation after watermark

- If the lifetime of two variables overlaps, then there will be an edge between the same.
- Having an edge between two storage variables of a colored interval graph indicates that a common register cannot be allocated for storing the two storage variables.

# 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.

# Particle Swarm Optimization (PSO) driven DSE for Optimal Watermark



# Optimization Methodology

- Problem Formulation
  - Given a control data flow graph (CDFG), determine, optimal watermarked solution  $(X_i) = N(R_1), N(R_2), \dots, N(R_D)$  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

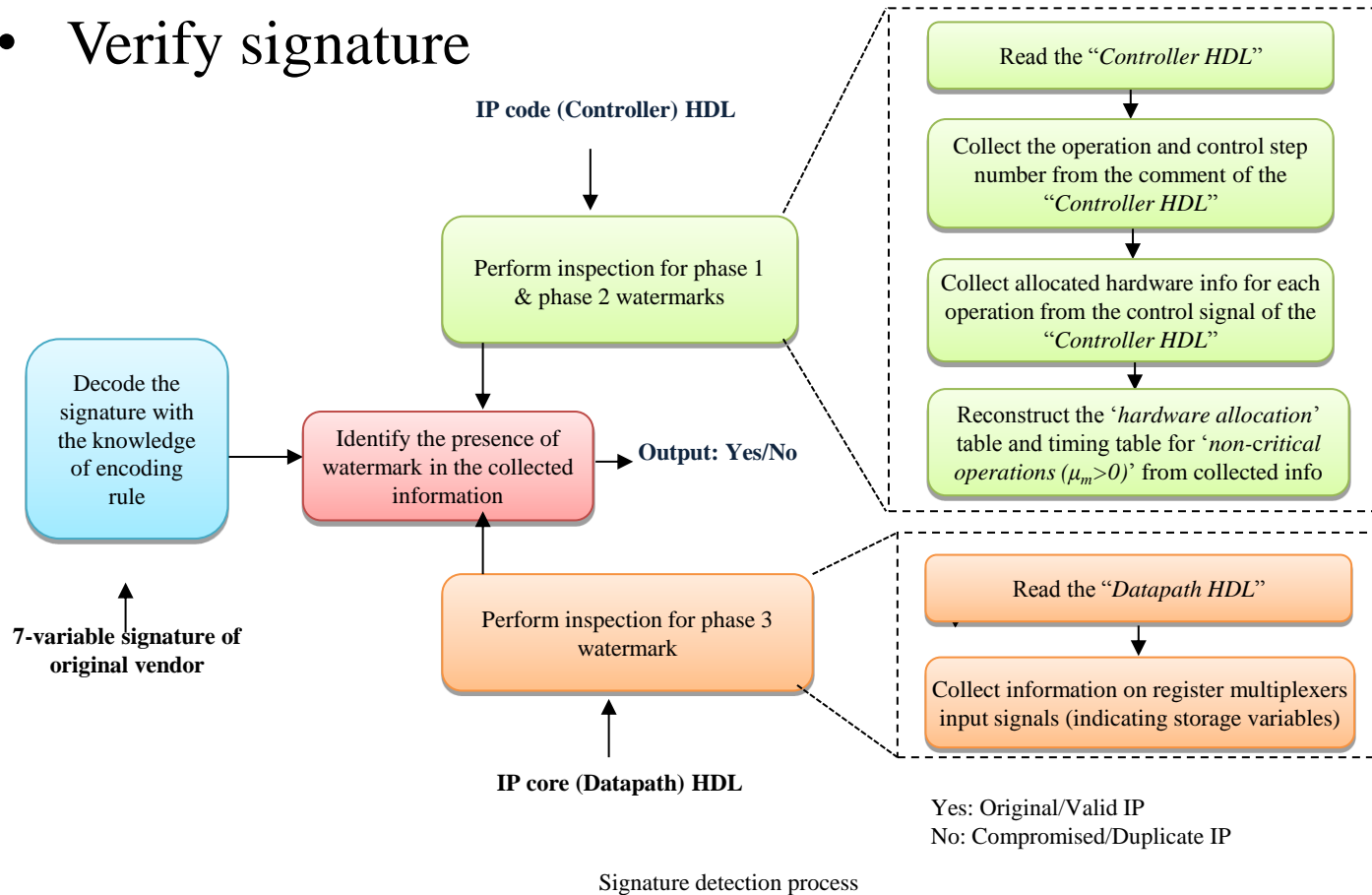
$A_{max}$  and  $L_{max}$  correspond to solutions with maximum area and delay in the design space

$W_1, W_2$  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

- Perform inspection of DUT (design under test)
- Verify signature



# Results and Analysis : Cost

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

Benchmark	Proposed Watermarked Solution		Watermarked Solution for [1]		Cost of Watermarked Solution	
	FU's	Registers	FU's	Registers	Proposed	[1]
<b>DWT</b>	1(+), 3(*)	6	2(+), 3(*)	5	-0.01	0.04
<b>ARF</b>	2(+), 4(*)	8	4(+), 2(*)	8	-0.21	0.02
<b>MPEG</b>	2(+), 5(*)	14	3(+), 7(*)	14	-0.44	-0.36
<b>IDCT</b>	4(+), 2(*)	8	4(+), 2(*)	8	0.08	0.08
<b>MESA</b>	3(+), 8(*)	48	9(+), 16(*)	48	-0.49	-0.38

# Results : Probability of Coincidence

**TABLE III: Measuring probability of coincidence ( $P_c$ ) as strength of watermark**

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

Benchmark	# of storage variables	$S(NW)$	$P_c$			
			# of watermarking constraints ( $w$ )			
			15	30	60	120
DWT	22	5	0.03	$1.23 \times 10^{-3}$	$1.53 \times 10^{-6}$	$2.3 \times 10^{-12}$
ARF	36	8	0.13	0.01	$3.3 \times 10^{-4}$	$1.09 \times 10^{-7}$
IDCT	50	8	0.13	0.01	$3.3 \times 10^{-4}$	$1.09 \times 10^{-7}$
MESA	139	48	0.72	0.53	0.28	0.07
MPEG	42	14	0.32	0.10	0.01	$1.37 \times 10^{-4}$

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

where

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

$c$  = number of colors used,

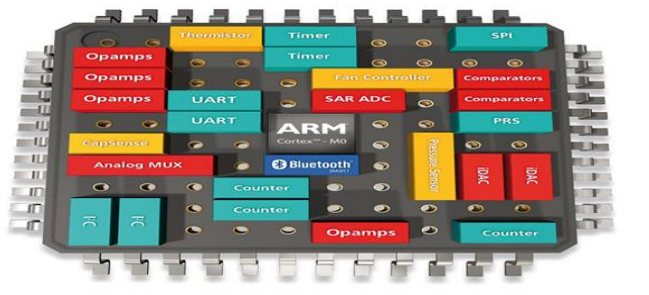
$w$  = # of watermarking constraints

(strength of the signature in terms of # of digits used).

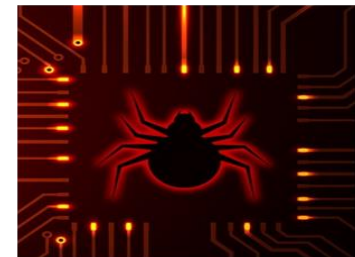


# Symmetrical IP Core Protection

Hardware (SoC) integrated third party IPs



Threats



Buyer fingerprint and seller watermark embedded in IP cores to protect against threats



IP 1



+



Buyer1



IP 2



Seller1

+



Buyer2

# Symmetrical IP Core Protection

- What is symmetrical IP core protection?
  - Seller and watermark.
  - Buyer and fingerprint.
- Why symmetrical IP core protection?
  - Tracing illegally resold/redistributed copies of a reusable IP core.
  - Piracy/forgery.
  - False claim of ownership.
- Why symmetrical IP core protection during HLS?
  - To meet the time to market demand.
  - Performance optimization.
  - Protects higher level as well as lower level designs.

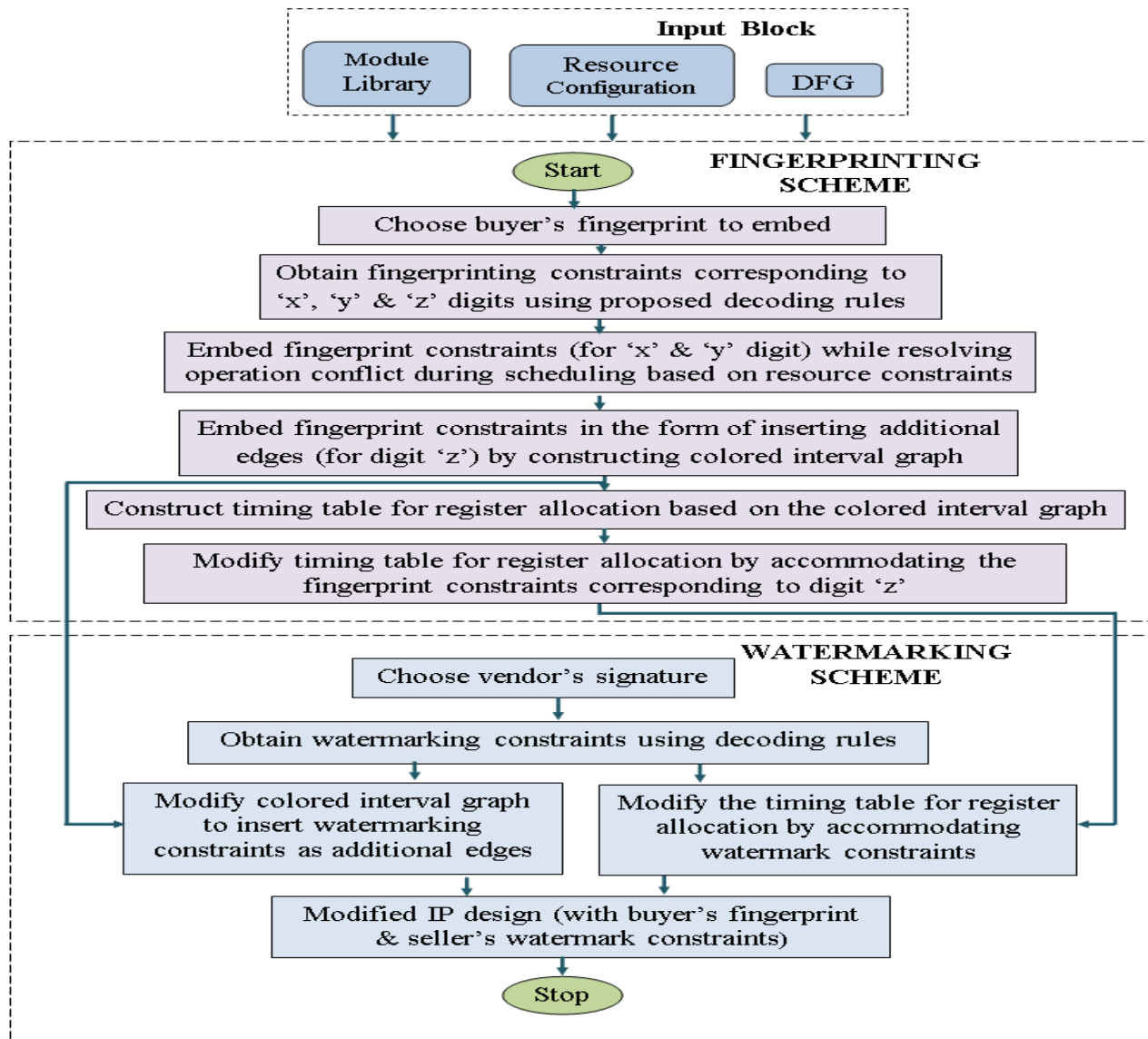
# Desired properties of signature

- Low embedding cost overhead
- Resiliency against attacks
- Fault tolerance
- Adaptability to any CAD Tool
- Signature creation and detection time
- Preserve correctness and functionalities

# Solution: Symmetrical IP Core

- Proposes multi-variable fingerprinting methodology during scheduling and register allocation phases of HLS.
- Proposes symmetrical IP core protection methodology first-time during HLS.
- Proposes symmetrical IP core protection with extremely low design overhead.
- Offers higher robustness, lower embedding cost, fault tolerance and faster signature encoding/decoding.

# Solution



# Fingerprinting methodology

## Fingerprint encoding

- **x** = Force **even operation** in **odd control** step while resolving scheduling conflict.
- **y** = Force **odd operation** in **even control** step while resolving scheduling conflict.
- **z** = Encoded value of edge with node pair (**odd, odd**) in colored interval graph (CIG).

## Fingerprint embedding process

- 1) Select desired buyer signature.
- 2) Decode buyer signature to its equivalent constraints.
- 3) Sort the operations in increasing order number.
- 4) Use the decoded constraints to perform scheduling during operation conflict.
- 5) Assign storage variables to registers from the schedule using the concept of CIG.
- 6) Insert additional edges in the CIG based on decoded constraints and perform re-assignment of register allocation.

# Watermarking methodology

## Watermark encoding

- **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, integer)**

## Watermark embedding process

- 1) Select desired seller signature.
- 2) Decode the seller watermark into its equivalent constraints.
- 3) Construct a CIG to represent registers required for storage variables in the fingerprint embedded schedule.
- 4) Insert additional edges in the CIG based on decoded watermarking constraints and embed the seller watermark into the fingerprinted design.



# Decoding of signatures

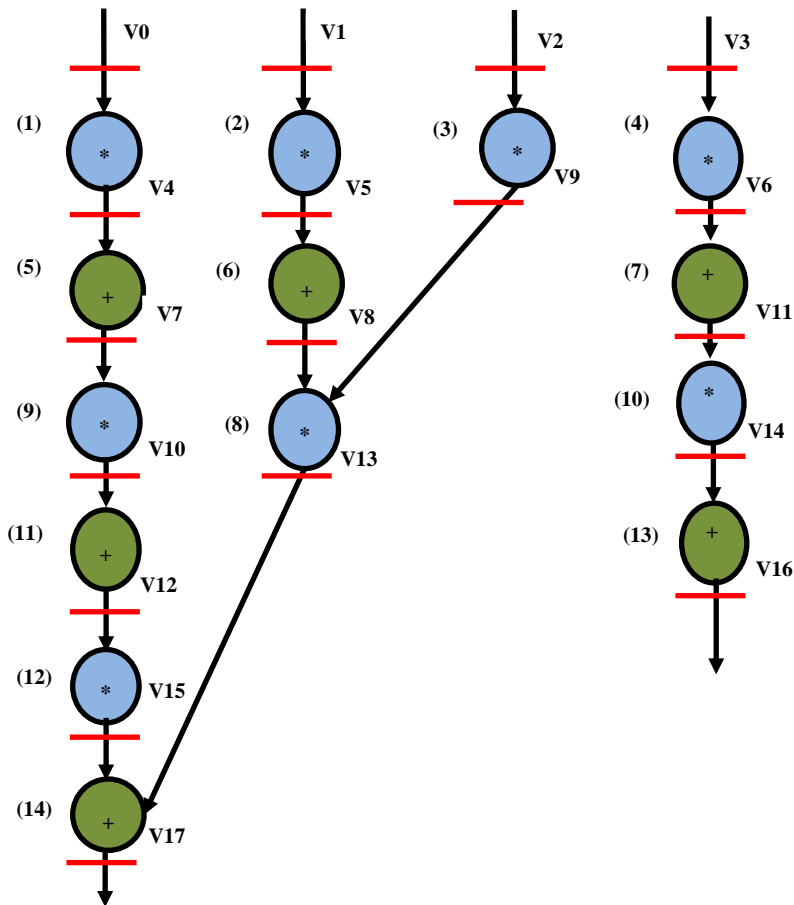
## User signature and its meaning

Fingerprint	Meaning
x	Assign opn 2 in cs1
x	Assign opn 4 in cs1
y	Assign opn 3 in cs2
x	Assign opn 6 in cs3
y	----
z	Insert edge between V1,V3
z	Insert edge between V1,V5

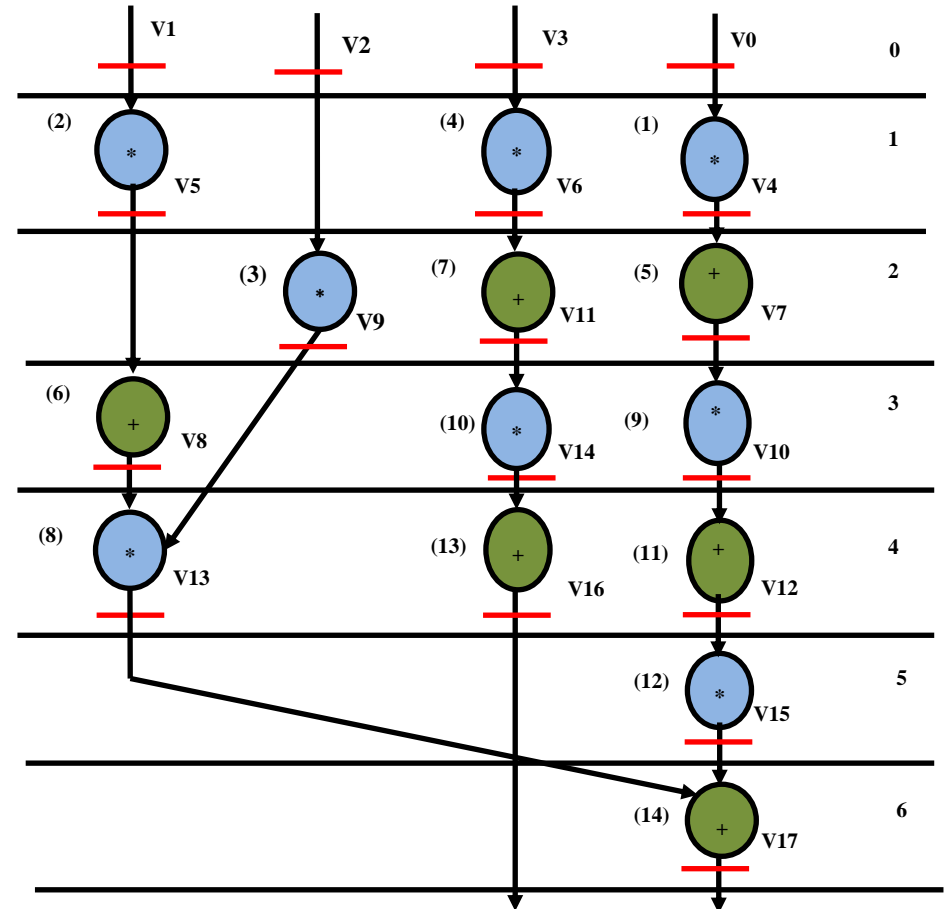
## Vendor signature its meaning

Watermark	Meaning
i	Insert edge between V2,V3
i	Insert edge between V2,V5
i	Insert edge between V2,V7
l	Insert edge between V2,V4
i	Insert edge between V2,V9
!	Insert edge between V0,V1
T	Insert edge between V1,V2

# Embedding Buyer Fingerprint in IP core during it's scheduling phase in architectural synthesis



Unscheduled DFG benchmark (before embedding buyer fingerprint)

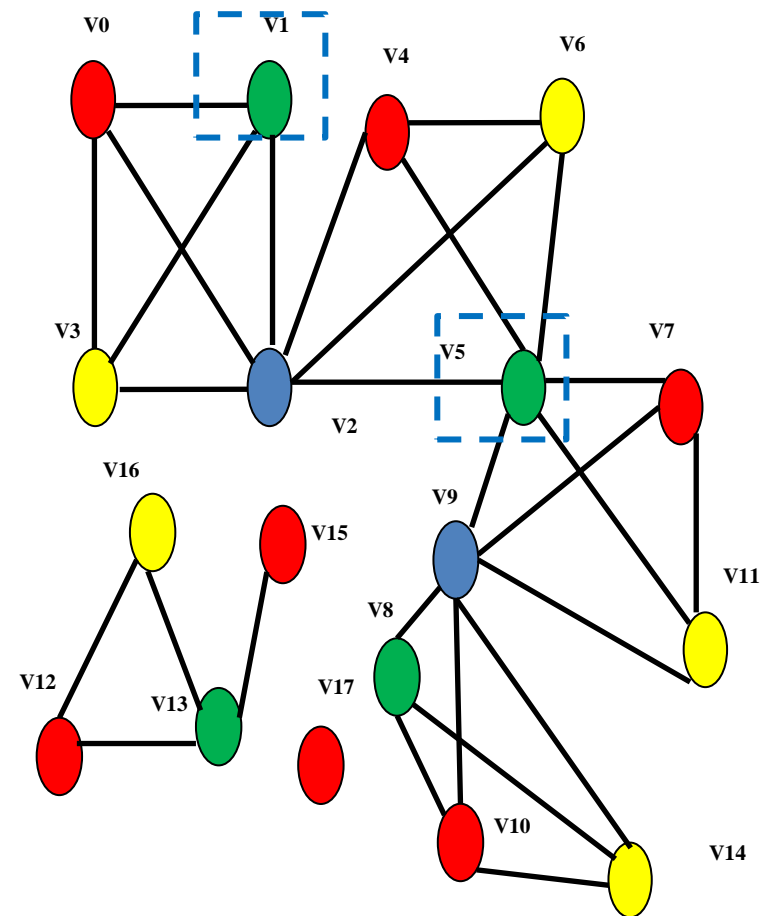


Scheduled IP core DFG based on 3M, 2A (after embedding buyer fingerprint)

# Before Embedding User Fingerprint in IP core

TIMING TABLE FOR REGISTER ALLOCATION BEFORE EMBEDDING  
ADDITIONAL EDGES FOR 'Z' DIGITS AS FINGERPRINT CONSTRAINTS

Control Step	Red (R)	Green (G)	Blue (B)	Yellow (Y)
0	V0	V1	V2	V3
1	V4	V5	V2	V6
2	V7	V5	V9	V11
3	V10	V8	V9	V14
4	V12	V13	-	V16
5	V15	V13	-	-
6	V17	-	-	-

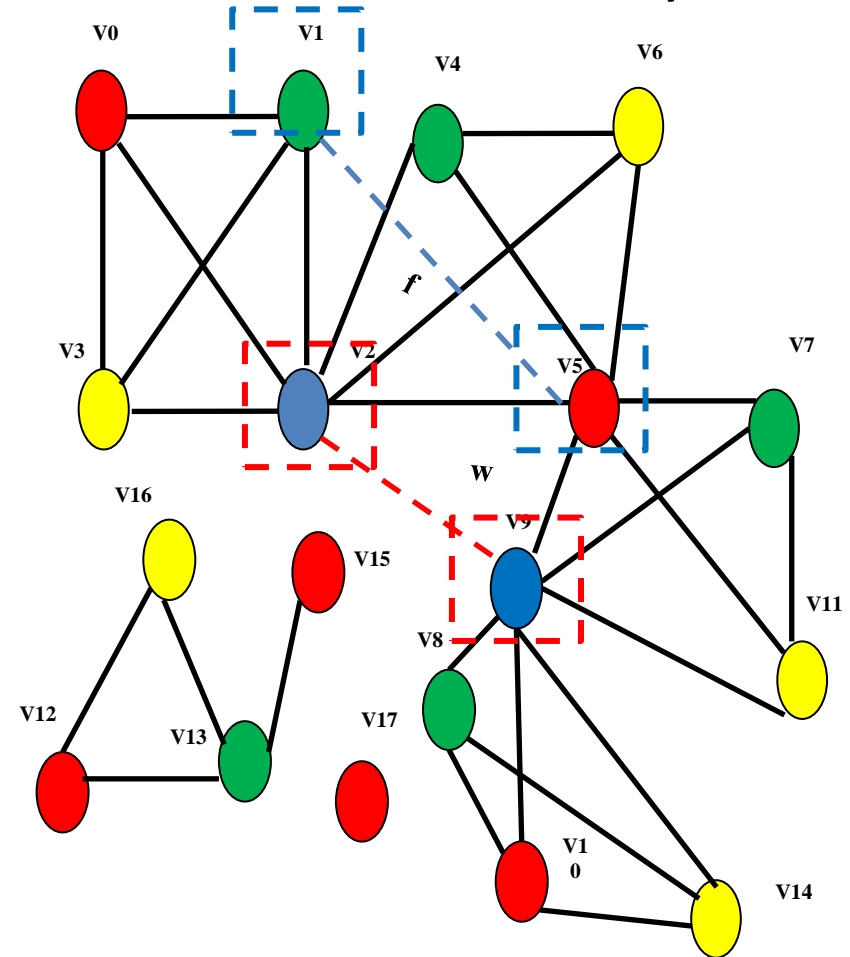


CIG with fingerprinting constraints

# After Embedding User Fingerprint (but before embedding vendor watermark)

TIMING TABLE FOR REGISTER ALLOCATION AFTER EMBEDDING  
ADDITIONAL EDGES AS FINGERPRINT CONSTRAINTS

Control Step	Red (R)	Green (G)	Blue (B)	Yellow (Y)
0	V0	V1	V2	V3
1	V5	V4	V2	V6
2	V5	V7	V9	V11
3	V10	V8	V9	V14
4	V12	V13	-	V16
5	V15	V13	-	-
6	V17	-	-	-

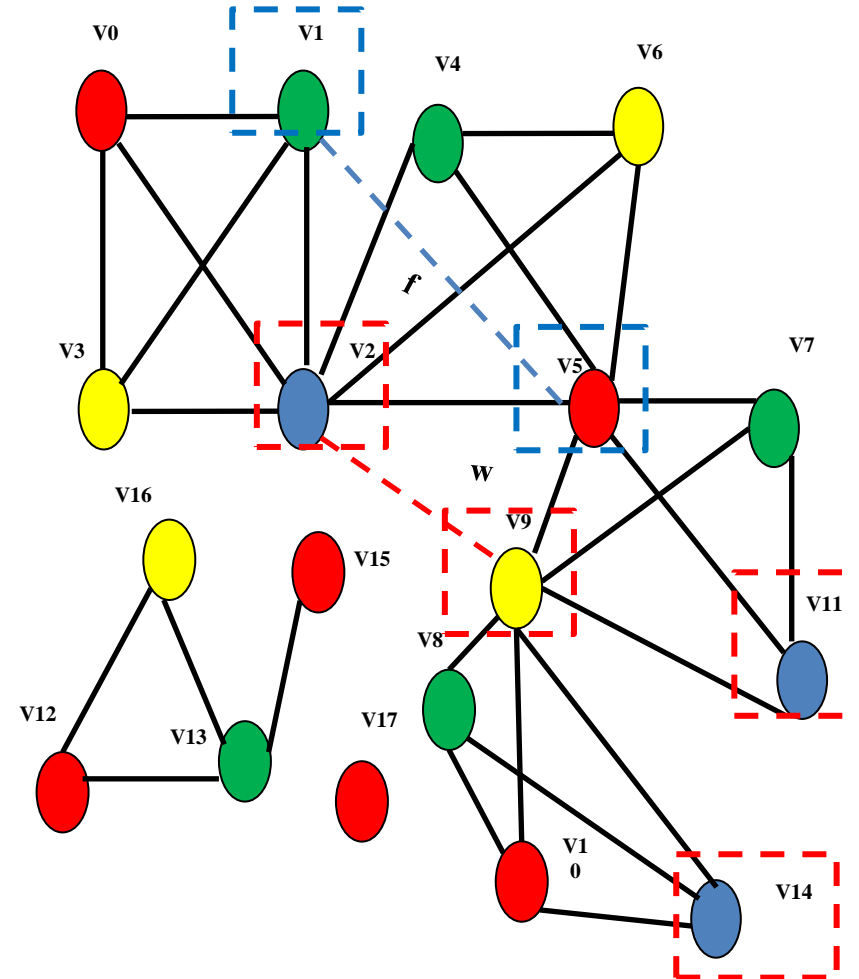


CIG denoting fingerprinting and watermarking constraints

# Before Embedding Buyer Fingerprint and Vendor Watermark in IP core

FINAL TIMING TABLE FOR REGISTER ALLOCATION AFTER EMBEDDING FINGERPRINT AND WATERMARK

Control Step	Red (R)	Green (G)	Blue (B)	Yellow (Y)
0	V0	V1	V2	V3
1	V5	V4	V2	V6
2	V5	V7	V11	V9
3	V10	V8	V14	V9
4	V12	V13	-	V16
5	V15	V13	-	-
6	V17	-	-	-



CIG with fingerprinting and watermarking constraints embedded

# After Embedding User Fingerprint & Vendor Watermark

VARIATION OF AREA, LATENCY AND COST WITH THE INCREMENT OF WATERMARK SIZE AFTER EMBEDDING FINGERPRINT

Benchmarks	Resource Configuration	# of watermark constraints (W) after embedding fingerprint								
		F=30, W=10			F=30, W=20			F=30, W=30		
		Area( $\mu m^2$ )	Latency(ps)	Cost	Area( $\mu m^2$ )	Latency(ps)	Cost	Area( $\mu m^2$ )	Latency(ps)	Cost
ARF	2(+), 4(*)	195.82	2.59	0.8391	196.61	2.59	0.8393	196.61	2.59	0.8393
DCT	4(+), 2(*)	222.56	3.80	0.8340	223.35	3.80	0.8343	223.35	3.80	0.8343
IDCT	4(+), 2(*)	223.35	3.73	0.8267	223.35	3.73	0.8267	223.35	3.73	0.8267
BPF	2(+), 2(*)	202.11	3.77	0.8784	202.90	3.77	0.8787	202.90	3.77	0.8787
FIR	4(+), 4(*)	179.31	1.86	0.7521	180.09	1.86	0.7526	180.09	1.86	0.7526
MPEG	3(+), 5(*)	224.13	2.38	0.6645	224.13	2.38	0.6645	224.13	2.38	0.6645
JPEG	4(+), 4(*)	724.30	14.24	0.7349	724.30	14.24	0.7349	724.30	14.24	0.7349







COMPARISON OF PROPOSED SYMMETRICAL IP CORE PROTECTION METHODOLOGY

Benchmarks	Resource Configuration	Area( $\mu m^2$ )			Latency(ps)			Cost		
		[11]	Proposed	Overhead(%)	[11]	Proposed	Overhead(%)	[11]	Proposed	Overhead(%)
ARF	2(+), 4(*)	196.61	196.61	0	2.46	2.59	5.02	0.8187	0.8393	2.45
DCT	4(+), 2(*)	223.35	223.35	0	3.73	3.80	1.84	0.8267	0.8343	0.91
IDCT	4(+), 2(*)	223.35	223.35	0	3.72	3.73	0.27	0.8248	0.8267	0.23
BPF	2(+), 2(*)	202.90	202.90	0	3.69	3.77	2.12	0.8705	0.8787	0.93
FIR	4(+), 4(*)	180.09	180.09	0	1.80	1.86	3.23	0.7375	0.7526	2.01
MPEG	3(+), 5(*)	224.13	224.13	0	2.36	2.38	0.84	0.6639	0.6645	0.09
JPEG	4(+), 4(*)	724.3	724.3	0	14.24	14.24	0	0.7349	0.7349	0

MEASURING PROBABILITY OF COINCIDENCE ( $P_c$ ) AS STRENGTH OF WATERMARK

Benchmarks	# of registers before fingerprint	$P_c$		
		# of watermark constraints (W)		
		W=10	W=20	W=30
ARF	8	0.26308	0.06921	0.01821
DCT	8	0.26308a	0.06921	0.01821
IDCT	9	0.30795	0.09483	0.0292
BPF	7	0.21406	0.04582	0.00981
FIR	8	0.26308	0.06921	0.01821
MPEG	14	0.47660	0.22715	0.10826
JPEG	12	0.41890	0.17548	0.07351

# IP Threat Models: Type 2

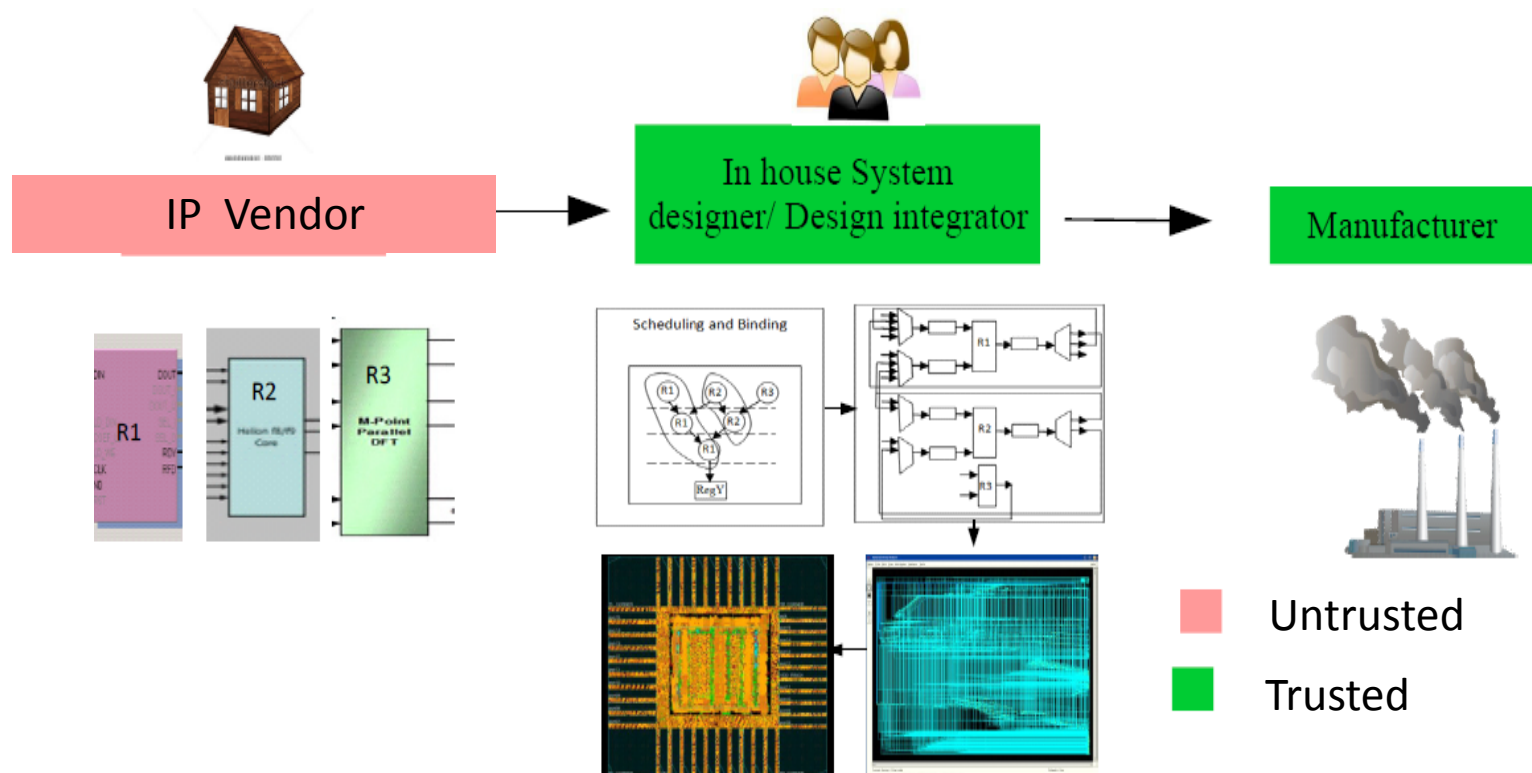
	3PIP Vendor	SoC Integrator	Foundry
Scenario 1			
Scenario 2			

Typical Trojan attack scenario in IC development cycle [6] (Note: Red star indicates attacker and green plus indicates shielding party/protector)



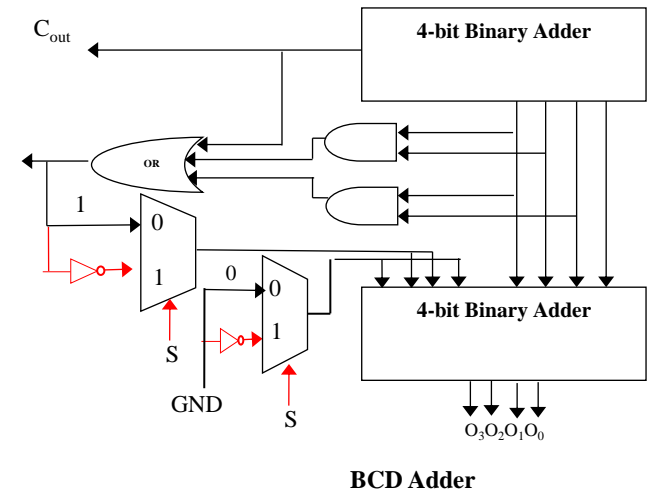
# IP Core design flow

- 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 Trojan: Security

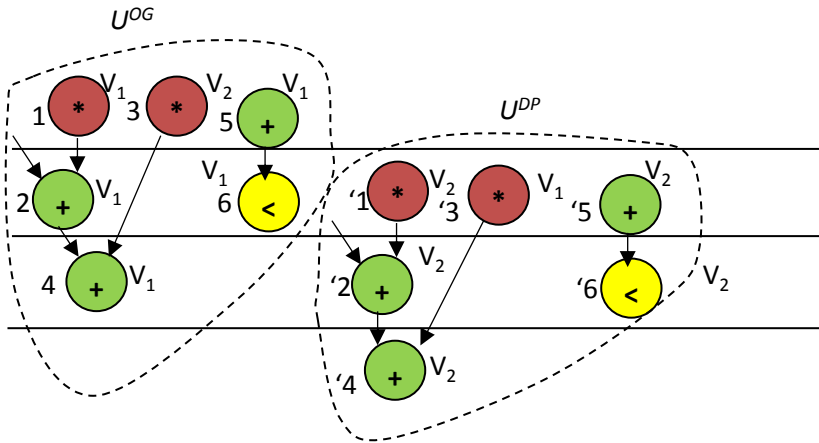
- What is a Trojan?
  - Malicious modification of an IC.
  - Trigger and payload.
  - External activation (antennas or sensors) or internal activation (FSM or counters).
- What are the different types of Trojan?
  - rare value triggered
  - time-triggered
- How a Trojan can be inserted?
  - Through third party IP (3PIP) cores.
- How to achieve Trojan secure design?
  - Dual Modular Redundant (DMR) schedule during HLS.



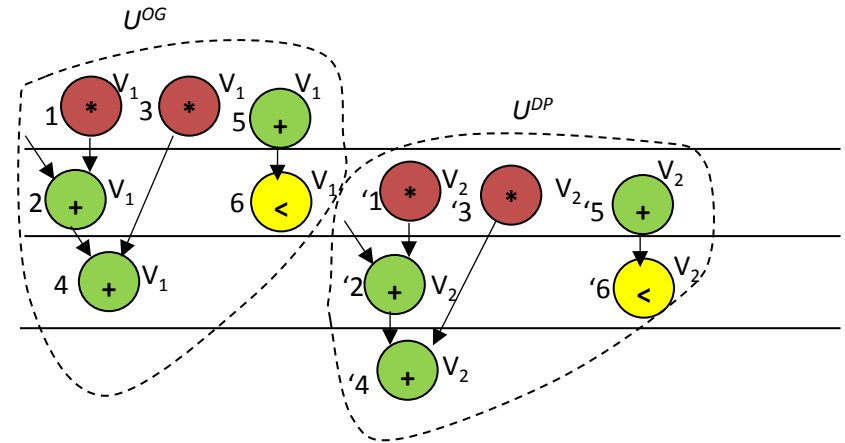
# IP core Trojan Detection Rules

- **Rule 1:** Vendor allocation procedure (Type 1):  $A_v = 00$ 
  - Alternate vendor assignment to operations in control step of a unit.
  - Similar operations of both  $U^{OG}$  and  $U^{DP}$  being assigned to different vendors.
- **Rule 2:** Vendor allocation procedure (Type 2):  $A_v = 01$ 
  - All operations of a specific unit being assigned to resources of same vendor type.
  - Similar operations of both original unit ( $U^{OG}$ ) and duplicate unit ( $U^{DP}$ ) being assigned to different vendors.
- **Rule 3:** Vendor allocation procedure (Type 3):  $A_v = 10$ 
  - All operations within critical path of a specific unit being strictly assigned to a vendor type while all operations of non critical path through alternate vendor type.
  - Operations of critical path of  $U^{OG}$  and  $U^{DP}$  are assigned to distinct vendors.
  - Similar operations of non critical path in both  $U^{OG}$  and  $U^{DP}$  being assigned to different vendors.
- **Rule 4:** Vendor allocation procedure (Type 4):  $A_v = 11$ 
  - Alternate vendor assignment to operations belonging to subsequent unrolled loop iterations within a unit.
  - Similar operations of unrolled loop iteration in both  $U^{OG}$  and  $U^{DP}$  assigned to different vendors.

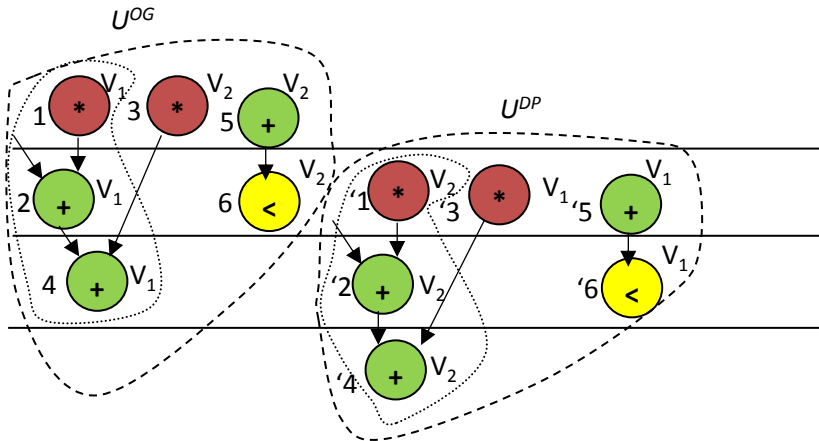
# Example to Secure an IP core



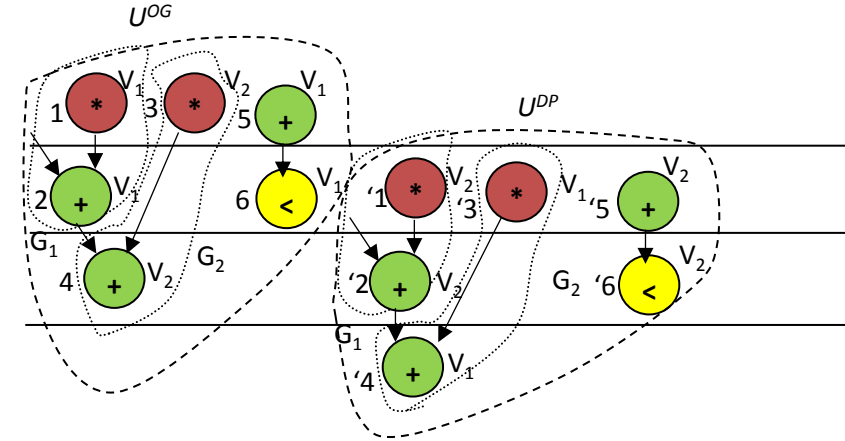
Scheduling and Binding of FIR for  $X_i = 2(+), 2(*), 2(<)$ ,  $U=2, I=4$   
based on Vendor Allocation Mode  $A_v = 00$ ;  
 $T_E^{DMR} = 45080$  ns and  $A_T^{DMR} = 13064$  au



Scheduling and Binding of FIR for  $X_i = 2(+), 2(*), 2(<)$ ,  $U=2, I=4$   
based on Vendor Allocation Mode  $A_v = 01$ ;  
 $T_E^{DMR} = 43080$  ns and  $A_T^{DMR} = 17996$  au



Scheduling and Binding of FIR for  $X_i = 2(+), 2(*), 2(<)$ ,  $U=2, I=4$   
based on Vendor Allocation Mode  $A_v = 10$ ;  
 $T_E^{DMR} = 45080$  ns and  $A_T^{DMR} = 13064$  au



Scheduling and Binding of FIR for  $X_i = 2(+), 2(*), 2(<)$ ,  $U=2, I=4$   
based on Vendor Allocation Mode  $A_v = 11$ ;  
 $T_E^{DMR} = 45070$  ns and  $A_T^{DMR} = 15096$  au

# My key Journal Contributions in CE Device Hardware Security

1. **A. Sengupta et. al** "Triple-Phase Watermarking for Reusable IP Core Protection during Architecture Synthesis", [IEEE Transactions on Computer Aided Design of Integrated Circuits & Systems \(TCAD\)](#), 2017
2. **A. Sengupta et. al** "Securing IoT Hardware: Threat models and Reliable, Low-power Design Solutions", [IEEE Transactions on Very Large Scale Integration \(VLSI\) Systems](#), 2017
3. **A. Sengupta et. al** "TL-HLS: Methodology for Low Cost Hardware Trojan Security Aware Scheduling with Optimal Loop Unrolling Factor during High Level Synthesis", [IEEE Transactions on Computer Aided Design of Integrated Circuits & Systems \(TCAD\)](#), 2016
4. **A. Sengupta et. al** "Exploring Low Cost Optimal Watermark for Reusable IP Cores during High Level Synthesis" , [IEEE Access Journal](#), 2016
5. **A. Sengupta et. al** "Protecting an Intellectual Property Core during Architectural Synthesis using High-Level Transformation Based Obfuscation", [IET Electronics Letters](#), 2017
6. **A. Sengupta et. al** "IP core Protection of CDFGs using Robust Watermarking during Behavioral Synthesis Based on User Resource Constraint and Loop Unrolling Factor", [IET Electronics Letters](#), 2016
7. **A. Sengupta et. al** "Low Cost Security Aware High Level Synthesis Methodology", [IET Journal on Computers & Digital Techniques \(CDT\)](#), 2016
8. **A. Sengupta et. al** "Protection of IP-Core Designs for CE Products", [IEEE Consumer Electronics Magazine](#), 2015
9. **A. Sengupta et. al** "Hardware Vulnerabilities and its Effect on CE Devices: Design-for-Security against Trojan", [IEEE Consumer Electronics Magazine](#), 2017
10. **A. Sengupta et. al** "Anti-Piracy aware IP Chipset Design for CE Devices: Robust Watermarking Approach", [IEEE Consumer Electronics Magazine](#), 2017.
11. **A. Sengupta et. al** "Hardware Security of CE Devices: Threat Models and Defence against IP Trojans and IP Piracy", [IEEE Consumer Electronics Magazine](#), 2017
12. **A. Sengupta et. al** "Forensic Engineering for Resolving Ownership Problem of Reusable IP Core generated during High Level Synthesis", [Elsevier Journal on Future Generation Computer Systems](#), Aug 2018
13. **A. Sengupta et. al** "Low Overhead Symmetrical Protection of Reusable IP Core using Robust Fingerprinting and Watermarking during High Level Synthesis", [Elsevier Journal on Future Generation Computer Systems](#), 2017
14. **A. Sengupta et. al** "Automated Low Cost Scheduling Driven Watermarking Methodology for Modern CAD High-Level Synthesis Tools" [Elsevier Journal of Advances in Engineering Software](#), 2017
15. **A. Sengupta et. al** Low cost optimized Trojan secured schedule at behavioral level for single & Nested loop control data flow graphs, [Elsevier VLSI Integration](#), 2016
16. **A. Sengupta et. al** "Security and Reliability Aware System Design for Mobile Computing Systems", [IEEE Access Journal](#), 2016

# References

1. I. Hong and M. Potkonjak, F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral Synthesis Techniques for Intellectual Property Protection," *ACM Trans. Des. Autom. Electron. Syst.*, July 2005, 523–545.
2. I. Hong and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," in *Proc. of the 36th Design Automation Conference*, 1999, pp. 849–854.
3. S. Meguerdichian and M. Potkonjak, "Watermarking while preserving the critical path," in *Proc. of 37th ACM/IEEE DAC*. 2000, pp.108–111.
4. A. L. Oliveira, "Techniques for the creation of digital watermarks in sequential circuit designs," *IEEE Trans. on CAD*, Vol. 20, No. 9, 2001, pp.1101–1117.
5. E. Charbon, "Hierarchical watermarking in IC design," in *Proc. of IEEE Custom Integrated Circuits Conf.*, 1998, pp. 295–298.
6. A. Sengupta, V. K. Mishra, "Swarm Intelligence Driven Simultaneous Adaptive Exploration of Datapath and Loop Unrolling Factor during Area-Performance Tradeoff ", *Proc. IEEE Symp. on VLSI* , 2014, pp. 106 112.
7. D. L. Irby, et al., "Placement watermarking of standard-cell designs in Mixed-Signal Design," in *Proc. of the SSMSD*, 2001, pp. 116–120.
8. S. P. Mohanty, et al., "Datapath Scheduling Using Dynamic Frequency Clocking", in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, 2002, pp. 58-63.
9. A. Sengupta and S. Bhadauria, "Untrusted Third Party Digital IP cores: Power-Delay Trade-off Driven Exploration of Hardware Trojan Secured Datapath during High Level Synthesis", *Proceedings of 25th Great Lake Symposium on VLSI (GLSVLSI)*, 2015, 167 – 172.
10. B. Le Gal and L. Bossuet, "Automatic low-cost IP watermarking technique based on output mark insertions", *Design Automation of Embedded Systems*, vol. 16, no. 2, pp. 71–92, June 2012.
11. F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," *ACM Trans. Design Autom. Electron. Syst.*, vol. 10, no. 3, pp. 523–545, Jul. 2005.

Thank you