Hardware Watermarking of Transient Fault Detectable IP Designs using Multivariate Encoded HLS Scheduling based Multi Modal Security Methodology

Published in IET-CDT

Anirban Sengupta, Vishal Chourasia, Nabendu Bhui, Aditya Anshul, “Watermarking of Transient Fault-Detectable IP Designs Using Multivariate HLS Scheduling Based Multimodal Security:”, *IET Computers & Digital Techniques*, 2025, 5926846, 17 pages, 2025.

Introduction

- The Modern computing system development relies on collaboration with multiple third-party IP vendors, creating potential security risks.
- This growing threat highlights the urgent need for robust security measures to protect hardware IP cores from misuse and abuse [1].
- However besides robust security countermeasure as the need of the hour, there is also a growing need to design IP cores that are fault secure (detectable) or fault tolerant due to potential transient fault that may occur.

[1] H. Pearce, R. Karri, B. Tan. 2023. High-Level Approaches to Hardware Security: A Tutorial. ACM Trans. Embed. Compu. Syst. 22, 3, Article 45, May 2023..

Novel Contributions of the Paper

- Presents a hardware watermarking methodology for transient fault-detectable IP designs.

- Presents a hardware watermarking methodology that leverages multivariate encoded HLS scheduling based multi-modal security.

- The presented IP watermarking technique is more robust than the prior watermarking techniques in terms of probability of coincidence, tamper tolerance, and probability of watermark decoding attack.

Related Work

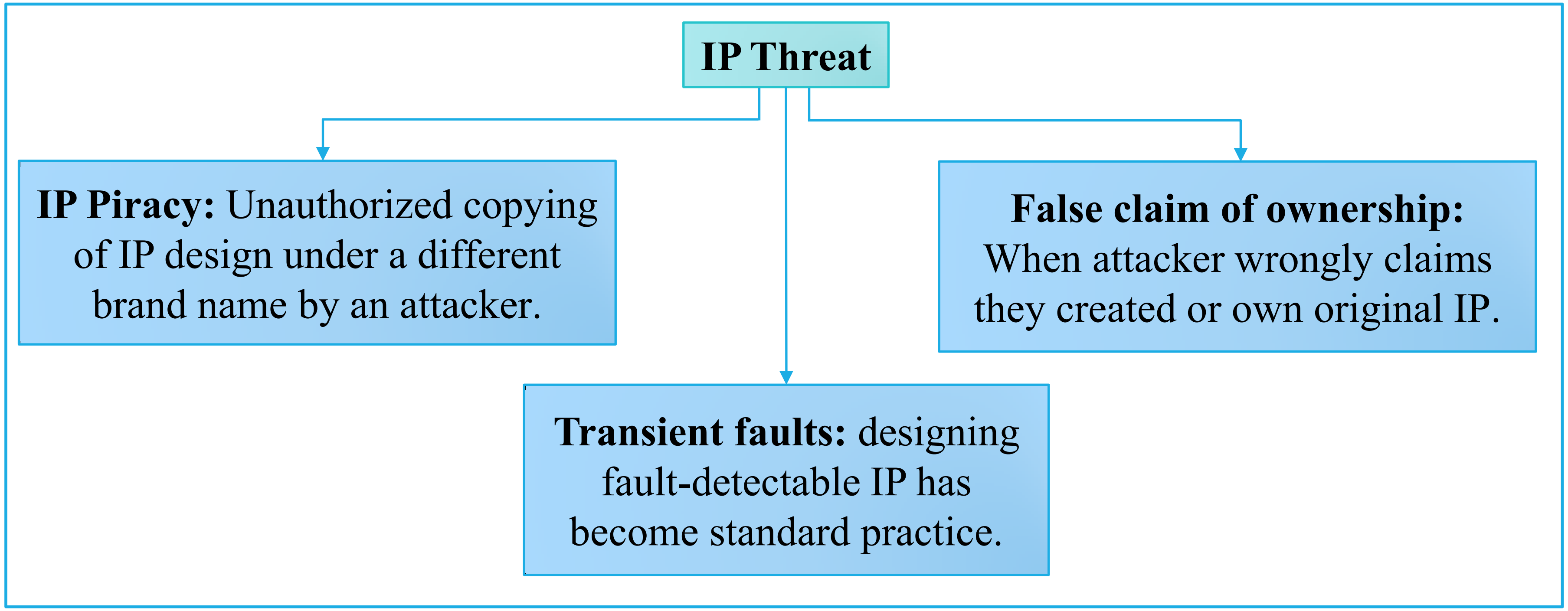
Sr. No.	Existing Work	Technique Used	Remark
1.	S. Rai, et.al., [21] (2019)	polymorphic inverter designs	However, [21],[22] produces watermark IP designs vulnerable to watermark collision and tampering attacks
2.	R. Karmakar, et.al.,[22] (2022)	sequential circuits using finite state machine	
3.	A. Sengupta et.al., [6] (2019)	hardware steganography-based security	However, bypass the piracy detection system by replicating the stego-mark

[21] S. Rai, A. Rupani, P. Nath and A. Kumar, "Hardware Watermarking Using Polymorphic Inverter Designs Based On Reconfigurable Nanotechnologies," *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2019, pp. 663-669..

[22] R. Karmakar, S. S. Jana and S. Chattopadhyay, "A Cellular Automata Guided Finite-State-Machine Watermarking Strategy for IP Protection of Sequential Circuits," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 2, pp. 806-823, 1 April-June 2022..

[6]. A. Sengupta, M. Rathor "IP Core Steganography for Protecting DSP Kernels used in CE Systems", IEEE Trans. on Consumer Electronics, Vol: 65, Issue: 4, pp. 506 – 515, Nov. 2019.

Threat Model



IP Threat

IP Piracy: Unauthorized copying of IP design under a different brand name by an attacker.

False claim of ownership: When attacker wrongly claims they created or own original IP.

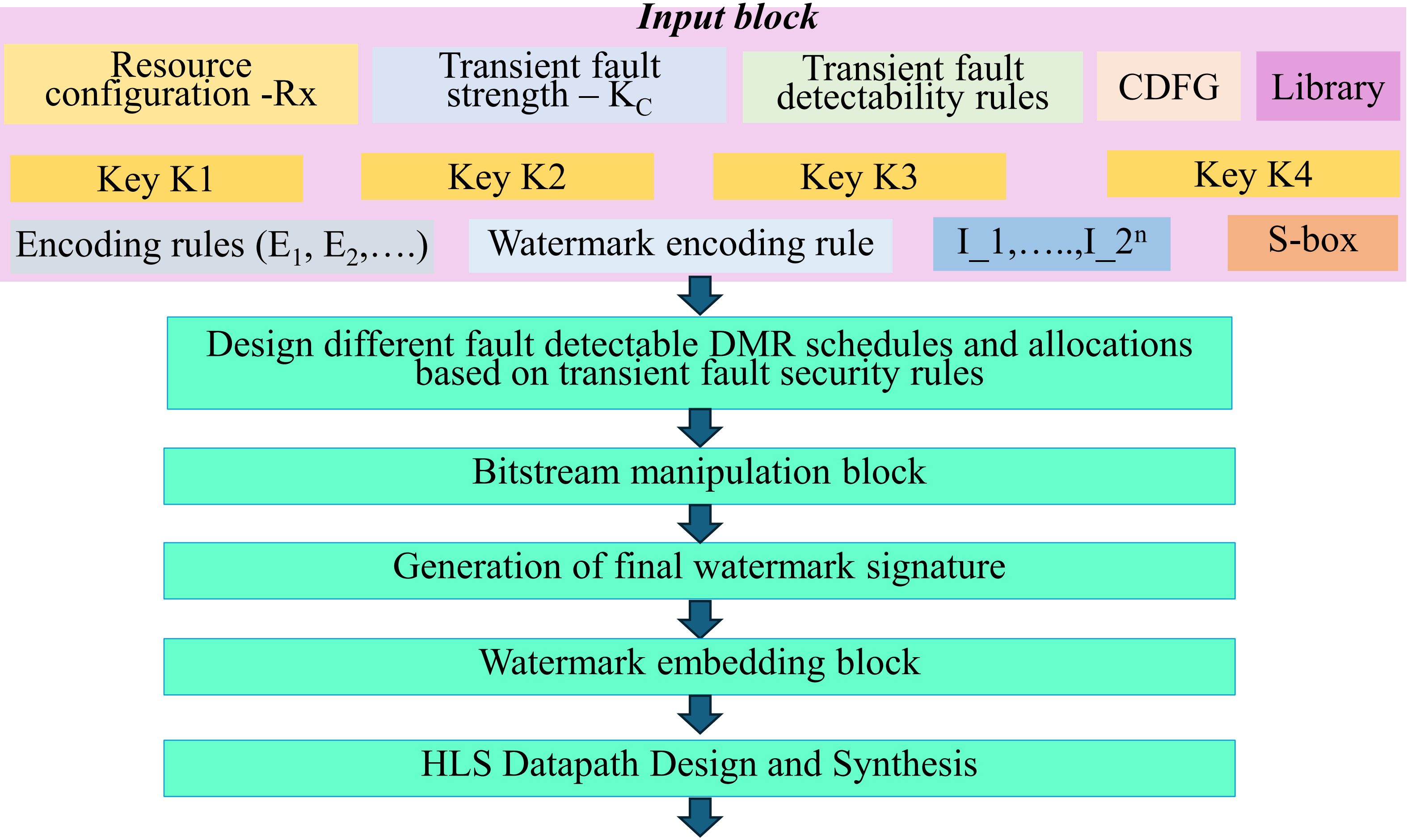
Transient faults: designing fault-detectable IP has become standard practice.

[5]. D. Karaklajić, J. -M. Schmidt and I. Verbauwhede, "Hardware Designer's Guide to Fault Attacks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 12, pp. 2295-2306, Dec. 2013.

[6] A. Sengupta, M. Rathor "IP Core Steganography for Protecting DSP Kernels used in CE Systems", *IEEE* Trans. on Consumer Electronics, Vol: 65, Issue: 4, pp. 506 – 515, Nov. 2019.

[7] M. Rostami, F. Koushanfar and R. Karri, "A Primer on Hardware Security: Models, Methods, and Metrics," Proceedings of the IEEE, vol. 102, no. 8, pp. 1283-1295, Aug. 2014

Proposed Methodology: Overview



Input block

Resource configuration -Rx

Transient fault strength – K_C

Transient fault detectability rules

CDFG

Library

Key K1

Key K2

Key K3

Key K4

Encoding rules (E_1, E_2,….)

Watermark encoding rule

I_1,…..,I_2^n

S-box

Design different fault detectable DMR schedules and allocations based on transient fault security rules

Bitstream manipulation block

Generation of final watermark signature

Watermark embedding block

HLS Datapath Design and Synthesis

Output: Watermarked Fault Detectable IP design

Fig. 1. Overview of the proposed approach

Details of Proposed Methodology

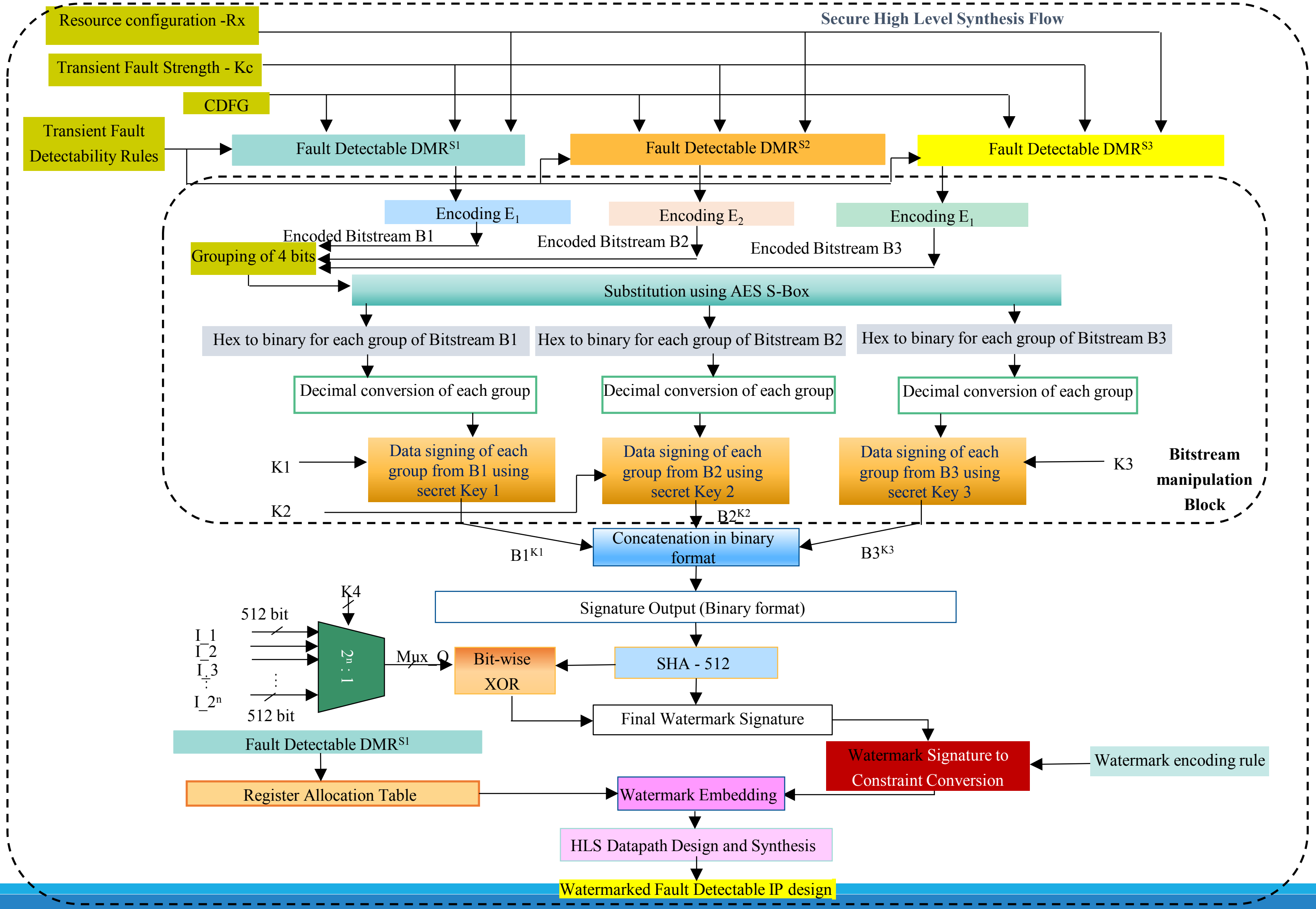
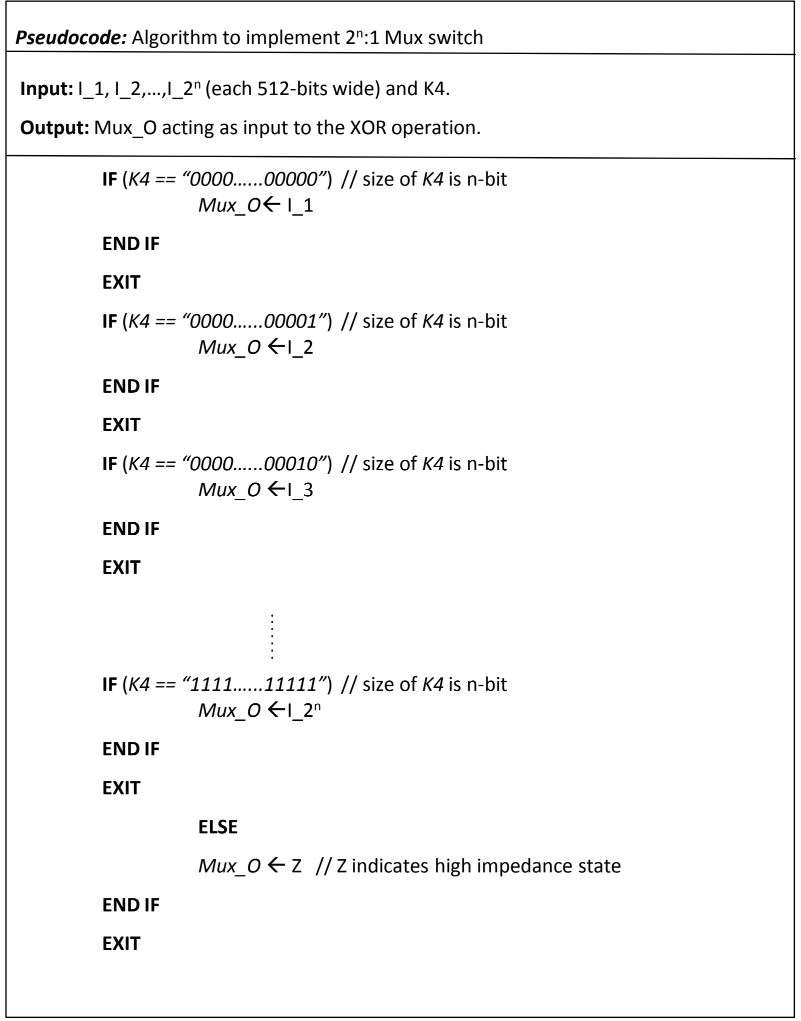

Secure High Level Synthesis Flow
Resource configuration -Rx
Transient Fault Strength - Kc
CDFG
Transient Fault Detectability Rules
Fault Detectable DMR^{S1}
Fault Detectable DMR^{S2}
Fault Detectable DMR^{S3}
Encoding E_1
Encoding E_2
Encoding E_1
Encoded Bitstream B1
Encoded Bitstream B2
Encoded Bitstream B3
Grouping of 4 bits
Substitution using AES S-Box
Hex to binary for each group of Bitstream B1
Hex to binary for each group of Bitstream B2
Hex to binary for each group of Bitstream B3
Decimal conversion of each group
Decimal conversion of each group
Decimal conversion of each group
K1
Data signing of each group from B1 using secret Key 1
K2
Data signing of each group from B2 using secret Key 2
Data signing of each group from B3 using secret Key 3
K3
Bitstream manipulation Block
B2^{K2}
B1^{K1}
Concatenation in binary format
B3^{K3}
Signature Output (Binary format)
K4
512 bit
I_1
I_2
I_3
I_2^n
512 bit
$2^n : 1$
Mux_O
Bit-wise XOR
SHA - 512
Final Watermark Signature
Fault Detectable DMR^{S1}
Watermark Signature to Constraint Conversion
Watermark encoding rule
Register Allocation Table
Watermark Embedding
HLS Datapath Design and Synthesis
Watermarked Fault Detectable IP design

Fig. 2(a). Details of the proposed hardware IP watermarking approach

Details of Proposed Methodology



```
Pseudocode: Algorithm to implement $2^n$:1 Mux switch

Input: I_1, I_2,…,I_$2^n$ (each 512-bits wide) and K4.
Output: Mux_O acting as input to the XOR operation.

        IF (K4 == "0000……00000")  // size of K4 is n-bit
                Mux_O← I_1
        END IF
        EXIT
        IF (K4 == "0000……00001")  // size of K4 is n-bit
                Mux_O ←I_2
        END IF
        EXIT
        IF (K4 == "0000……00010")  // size of K4 is n-bit
                Mux_O ←I_3
        END IF
        EXIT
                        ⋮
        IF (K4 == "1111……11111")  // size of K4 is n-bit
                Mux_O ←I_$2^n$
        END IF
        EXIT
                ELSE
                Mux_O ← Z   // Z indicates high impedance state
        END IF
        EXIT
```

Fig.2(b). Pseudocode to implement 2^n:1 Mux switch

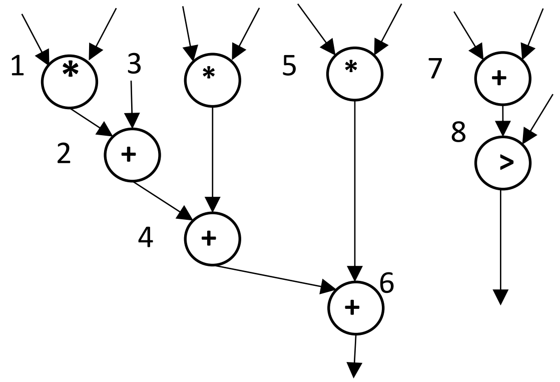


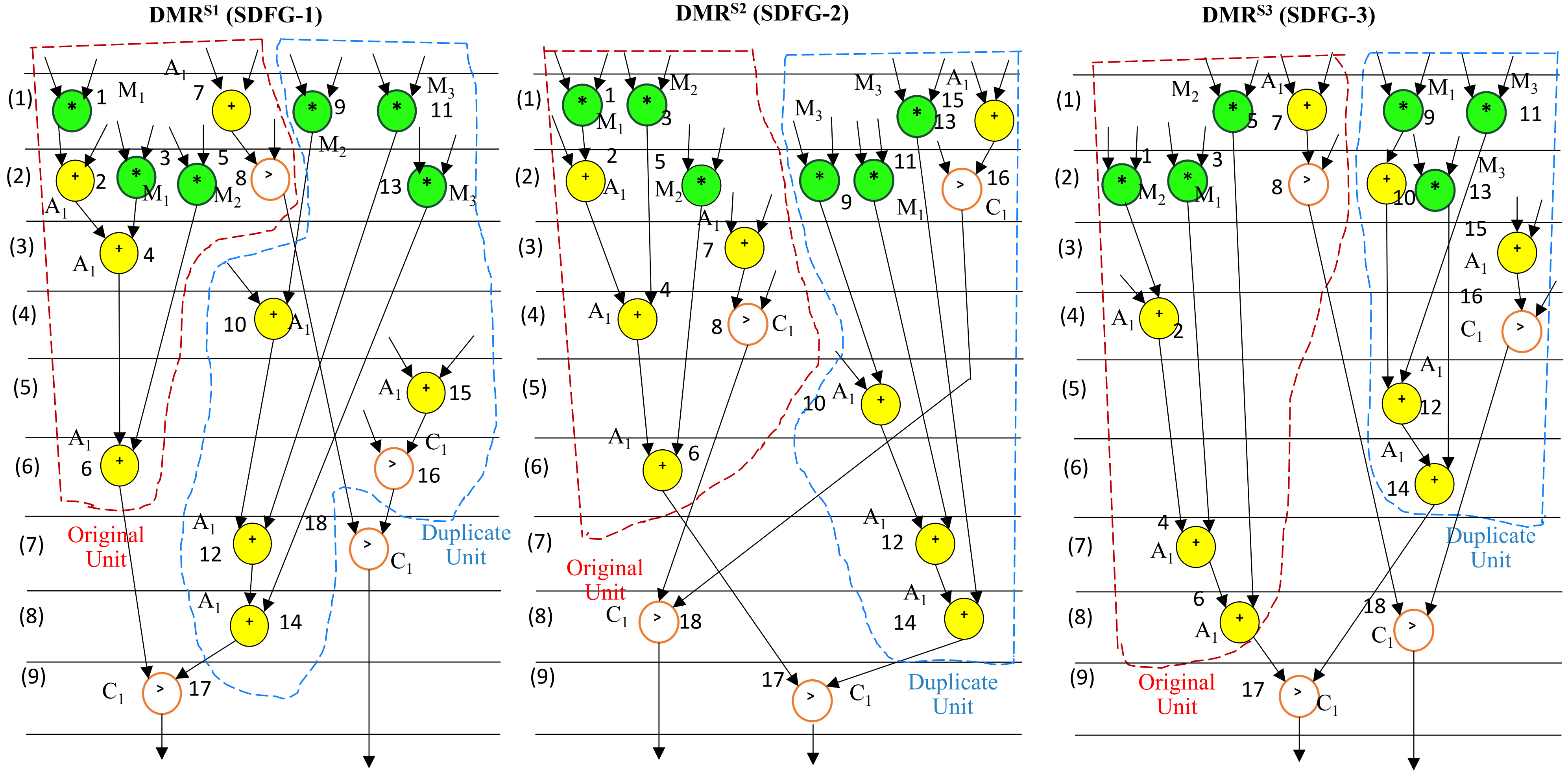
Fig. 3. Data flow graph of FIR filter (UF=3)

Generation of Fault Detectable Scheduled Design

Transient Fault-Detectability Rules

1. If original and sister operations of respective original and duplicate units are *KC-control step* apart, then allocate the same hardware resource (*i.e.*, multiplier, adder, or subtractor) in the sister operations accordingly.
2. If there is availability of multiple hardware instances of the same type, then allocate distinct hardware resources in the sister operation of the DMR to ensure fault detection.
3. If the above rules do not ensure fault detectability, then reschedule the sister operations of the duplicate unit by shifting downward, one control step at a time, until it complies with the first rule.

Generation of Fault Detectable Scheduled Design



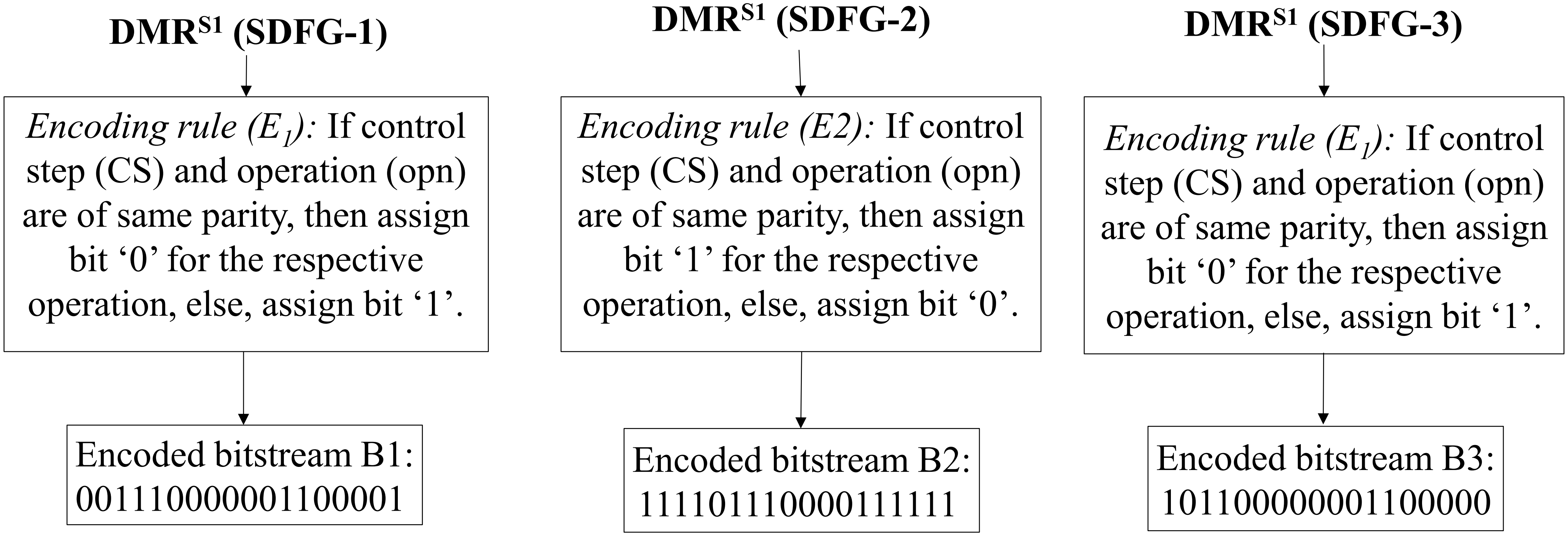
DMR^{S1} (SDFG-1)

DMR^{S2} (SDFG-2)

DMR^{S3} (SDFG-3)

Fig.4. (a). Generation of different scheduled fault detectable FIR filter (UF=3).

Generation of Encoded Bitstream



DMR^{S1} (SDFG-1)

Encoding rule (E_1): If control step (CS) and operation (opn) are of same parity, then assign bit ‘0’ for the respective operation, else, assign bit ‘1’.

Encoded bitstream B1: 001110000001100001

DMR^{S1} (SDFG-2)

Encoding rule (E2): If control step (CS) and operation (opn) are of same parity, then assign bit ‘1’ for the respective operation, else, assign bit ‘0’.

Encoded bitstream B2: 111101110000111111

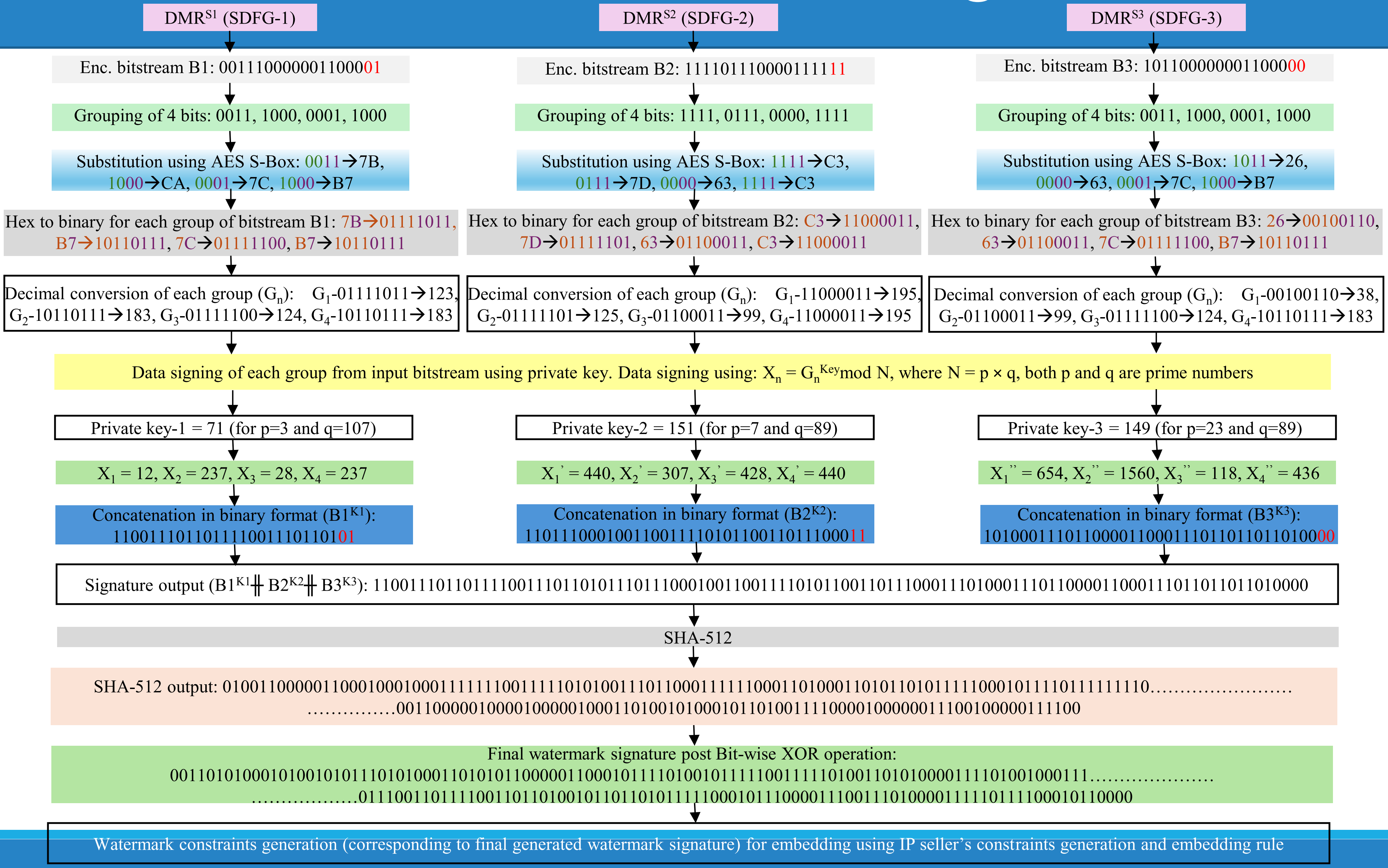
DMR^{S1} (SDFG-3)

Encoding rule (E_1): If control step (CS) and operation (opn) are of same parity, then assign bit ‘0’ for the respective operation, else, assign bit ‘1’.

Encoded bitstream B3: 101100000001100000

Fig. 4(b). Generation of encoded bitstream (B1, B2, and B3)

Generation of Final Watermark signature



DMR^{S1} (SDFG-1)

Enc. bitstream B1: 00111000000110000 1

Grouping of 4 bits: 0011, 1000, 0001, 1000

Substitution using AES S-Box: 0011→7B, 1000→CA, 0001→7C, 1000→B7

Hex to binary for each group of bitstream B1: 7B→01111011, B7→10110111, 7C→01111100, B7→10110111

Decimal conversion of each group (G_n): G_1-01111011→123, G_2-10110111→183, G_3-01111100→124, G_4-10110111→183

DMR^{S2} (SDFG-2)

Enc. bitstream B2: 111101110000111111

Grouping of 4 bits: 1111, 0111, 0000, 1111

Substitution using AES S-Box: 1111→C3, 0111→7D, 0000→63, 1111→C3

Hex to binary for each group of bitstream B2: C3→11000011, 7D→01111101, 63→01100011, C3→11000011

Decimal conversion of each group (G_n): G_1-11000011→195, G_2-01111101→125, G_3-01100011→99, G_4-11000011→195

DMR^{S3} (SDFG-3)

Enc. bitstream B3: 101100000001100000

Grouping of 4 bits: 0011, 1000, 0001, 1000

Substitution using AES S-Box: 1011→26, 0000→63, 0001→7C, 1000→B7

Hex to binary for each group of bitstream B3: 26→00100110, 63→01100011, 7C→01111100, B7→10110111

Decimal conversion of each group (G_n): G_1-00100110→38, G_2-01100011→99, G_3-01111100→124, G_4-10110111→183

Data signing of each group from input bitstream using private key. Data signing using: $X_n = G_n^{Key} \bmod N$, where $N = p \times q$, both p and q are prime numbers

Private key-1 = 71 (for p=3 and q=107)

$X_1 = 12, X_2 = 237, X_3 = 28, X_4 = 237$

Concatenation in binary format ($B1^{K1}$): 110011101101111001110110101

Private key-2 = 151 (for p=7 and q=89)

$X_1' = 440, X_2' = 307, X_3' = 428, X_4' = 440$

Concatenation in binary format ($B2^{K2}$): 11011100010011001111010110011011100011

Private key-3 = 149 (for p=23 and q=89)

$X_1'' = 654, X_2'' = 1560, X_3'' = 118, X_4'' = 436$

Concatenation in binary format ($B3^{K3}$): 101000111011000011000111011011011010000

Signature output ($B1^{K1} \# B2^{K2} \# B3^{K3}$): 11001110110111100111011010111011100010011001111010110011011100011101000111011000011000111011011011010000

SHA-512

SHA-512 output: 0100110000011000100010001111111001111101010011101100011111100011010001101011010111110001011110111111110……………………
……………0011000000100001000000100011010010100010110100111100001000000111001000000111100

Final watermark signature post Bit-wise XOR operation:
001101010001010010101110101000110101011000001100010111101001011111001111101001101010000111101001000111…………………
………………011100110111110011011010010110110101111100010111000011100111010000111110111100010110000

Watermark constraints generation (corresponding to final generated watermark signature) for embedding using IP seller’s constraints generation and embedding rule

Fig. 5. Demonstration of final watermark signature generation using proposed approach (for FIR filter)

Watermark Embedding Rule

Embedding rule:

1. If the watermark signature bit is 0, then even-even storage variable pairs (Pi, Pj) from the fault detectable DMR design are generated as watermark constraints, where i and j represent the storage variable indices.
2. If the watermark signature bit 1, the generated watermark constraints are odd-odd storage variable pairs of the fault detectable DMR design.

Watermark Embedding Process

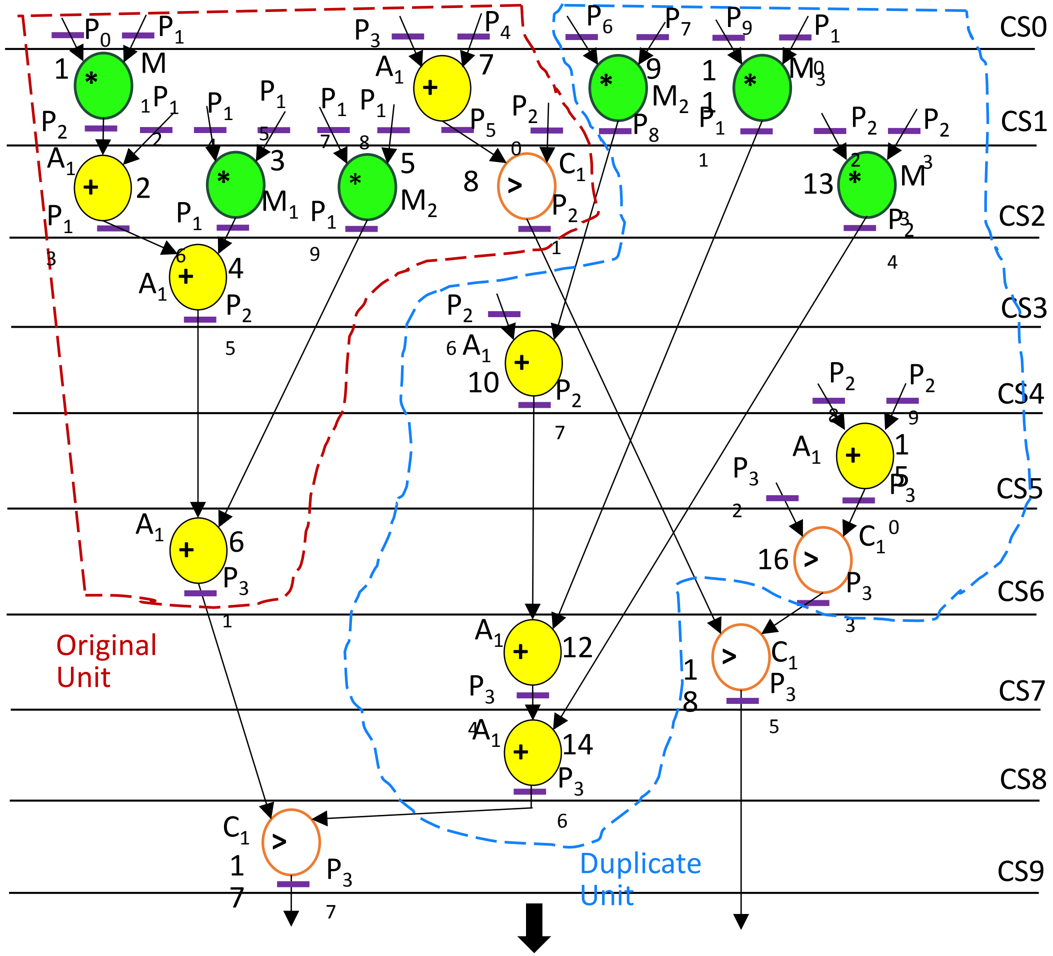
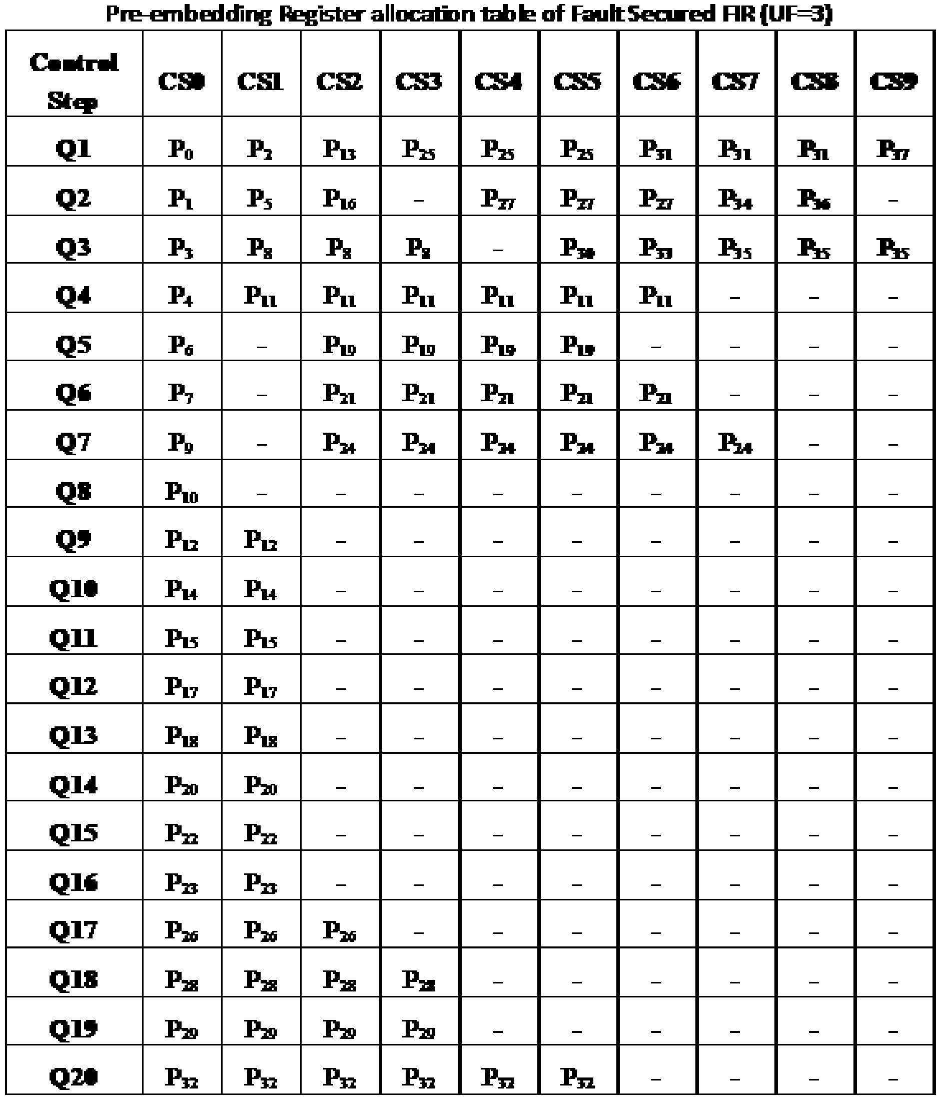


Fig. 6. Fault schedule DMR SDFG of FIR filter (UF=3, using Rx = 1 adder (+), 3 multiplier (*), 1 comparator (>), and Kc = 2CS) and its corresponding RAT (pre-embedding)



Pre-embedding Register allocation table of Fault Secured FIR (UF=3)

Control Step	CS0	CS1	CS2	CS3	CS4	CS5	CS6	CS7	CS8	CS9
Q1	P_0	P_2	P_{13}	P_{25}	P_{25}	P_{25}	P_{31}	P_{31}	P_{31}	P_{37}
Q2	P_1	P_5	P_{16}	–	P_{27}	P_{27}	P_{27}	P_{34}	P_{36}	–
Q3	P_3	P_8	P_8	P_8	–	P_{30}	P_{33}	P_{35}	P_{35}	P_{35}
Q4	P_4	P_{11}	P_{11}	P_{11}	P_{11}	P_{11}	P_{11}	–	–	–
Q5	P_6	–	P_{19}	P_{19}	P_{19}	P_{19}	–	–	–	–
Q6	P_7	–	P_{21}	P_{21}	P_{21}	P_{21}	P_{21}	–	–	–
Q7	P_9	–	P_{24}	P_{24}	P_{24}	P_{24}	P_{24}	P_{24}	–	–
Q8	P_{10}	–	–	–	–	–	–	–	–	–
Q9	P_{12}	P_{12}	–	–	–	–	–	–	–	–
Q10	P_{14}	P_{14}	–	–	–	–	–	–	–	–
Q11	P_{15}	P_{15}	–	–	–	–	–	–	–	–
Q12	P_{17}	P_{17}	–	–	–	–	–	–	–	–
Q13	P_{18}	P_{18}	–	–	–	–	–	–	–	–
Q14	P_{20}	P_{20}	–	–	–	–	–	–	–	–
Q15	P_{22}	P_{22}	–	–	–	–	–	–	–	–
Q16	P_{23}	P_{23}	–	–	–	–	–	–	–	–
Q17	P_{26}	P_{26}	P_{26}	–	–	–	–	–	–	–
Q18	P_{28}	P_{28}	P_{28}	P_{28}	–	–	–	–	–	–
Q19	P_{29}	P_{29}	P_{29}	P_{29}	–	–	–	–	–	–
Q20	P_{32}	P_{32}	P_{32}	P_{32}	P_{32}	P_{32}	–	–	–	–

Watermark Embedding Process

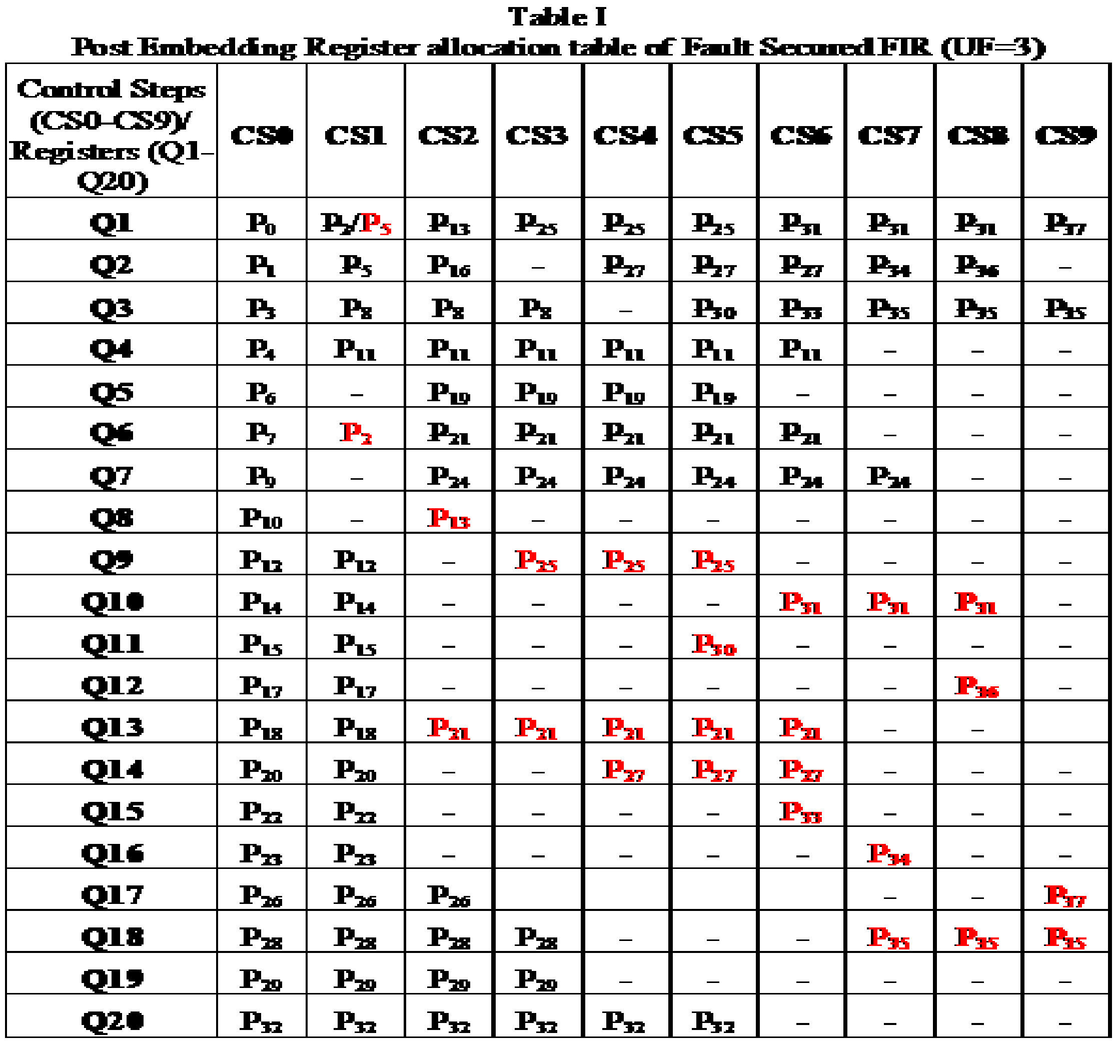


Table I

Post Embedding Register allocation table of Fault Secured FIR (UF=3)

Control Steps (CS0–CS9)/ Registers (Q1–Q20)	CS0	CS1	CS2	CS3	CS4	CS5	CS6	CS7	CS8	CS9
Q1	P_0	P_2/P_5	P_{13}	P_{25}	P_{25}	P_{25}	P_{31}	P_{31}	P_{31}	P_{37}
Q2	P_1	P_5	P_{16}	–	P_{27}	P_{27}	P_{27}	P_{34}	P_{36}	–
Q3	P_3	P_8	P_8	P_8	–	P_{30}	P_{33}	P_{35}	P_{35}	P_{35}
Q4	P_4	P_{11}	P_{11}	P_{11}	P_{11}	P_{11}	P_{11}	–	–	–
Q5	P_6	–	P_{19}	P_{19}	P_{19}	P_{19}	–	–	–	–
Q6	P_7	P_2	P_{21}	P_{21}	P_{21}	P_{21}	P_{21}	–	–	–
Q7	P_9	–	P_{24}	P_{24}	P_{24}	P_{24}	P_{24}	P_{24}	–	–
Q8	P_{10}	–	P_{13}	–	–	–	–	–	–	–
Q9	P_{12}	P_{12}	–	P_{25}	P_{25}	P_{25}	–	–	–	–
Q10	P_{14}	P_{14}	–	–	–	–	P_{31}	P_{31}	P_{31}	–
Q11	P_{15}	P_{15}	–	–	–	P_{30}	–	–	–	–
Q12	P_{17}	P_{17}	–	–	–	–	–	–	P_{36}	–
Q13	P_{18}	P_{18}	P_{21}	P_{21}	P_{21}	P_{21}	P_{21}	–	–	
Q14	P_{20}	P_{20}	–	–	P_{27}	P_{27}	P_{27}	–	–	–
Q15	P_{22}	P_{22}	–	–	–	–	P_{33}	–	–	–
Q16	P_{23}	P_{23}	–	–	–	–	–	P_{34}	–	–
Q17	P_{26}	P_{26}	P_{26}					–	–	P_{37}
Q18	P_{28}	P_{28}	P_{28}	P_{28}	–	–	–	P_{35}	P_{35}	P_{35}
Q19	P_{29}	P_{29}	P_{29}	P_{29}	–	–	–	–	–	–
Q20	P_{32}	P_{32}	P_{32}	P_{32}	P_{32}	P_{32}	–	–	–	–

Watermark Embedding Process

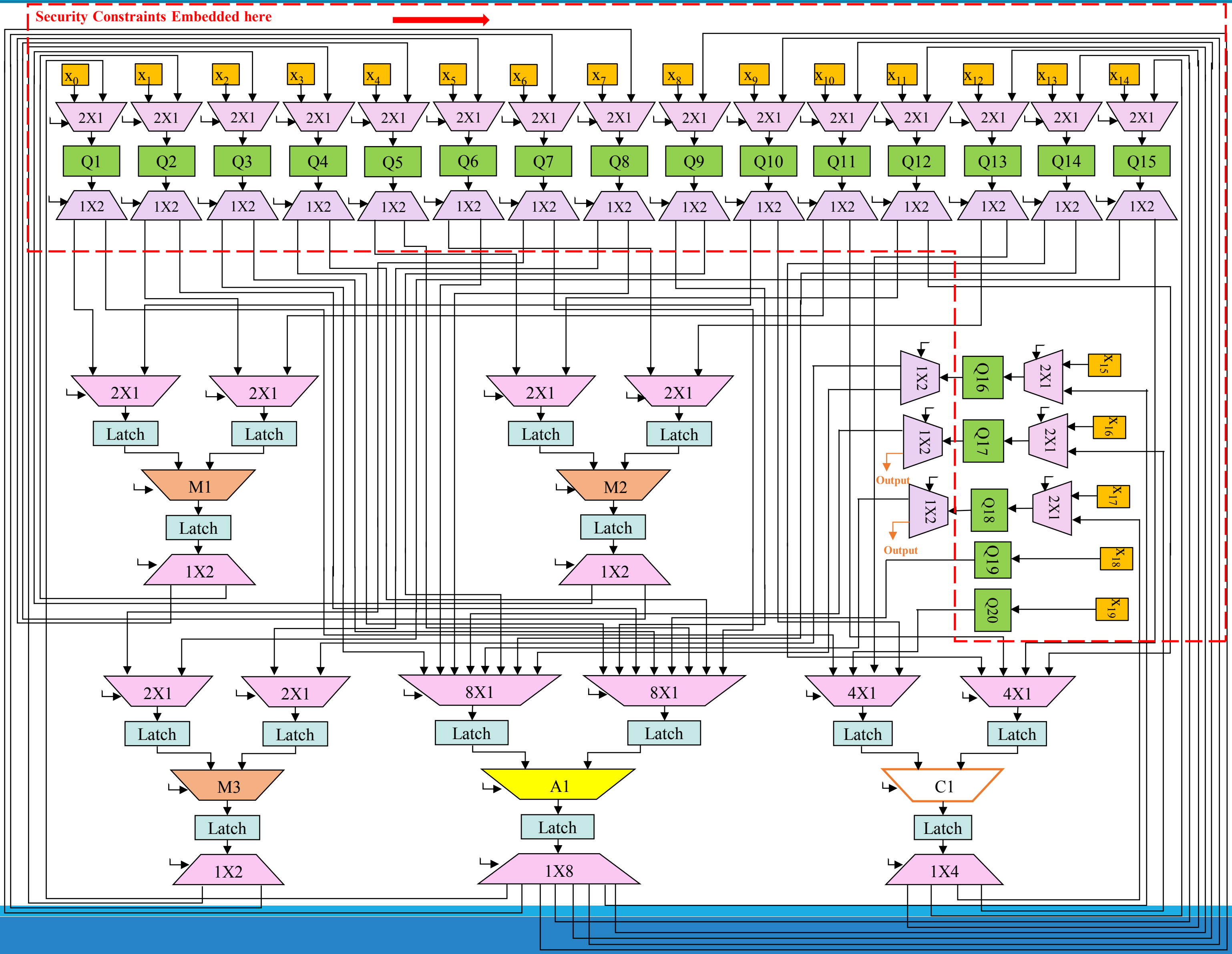


Fig. 7. RTL datapath corresponding to watermarked fault-detectable FIR filter

Watermark Detection/ Validation

- An attacker needs to decode several security layers and parameters used for watermark signature generation, in order to establish the watermark:

i. Multivariate fault detectable DMR design and their corresponding encoded bitstream generation process using encoding keys E1, E2..., En.
ii. Employed AES S-box and data signing keys K1, K2, and K3.
iii. Concatenation rule to generate the binary signature output.
iv. SHA-512 and 2n:1 Mux-controlled bit-wise XOR operation.
v. Encoding rule used for watermark constraint generation and embedding.

These parameters are unknown to an attacker. Thus, the proposed approach provides more definite proof-of-IP authenticity.

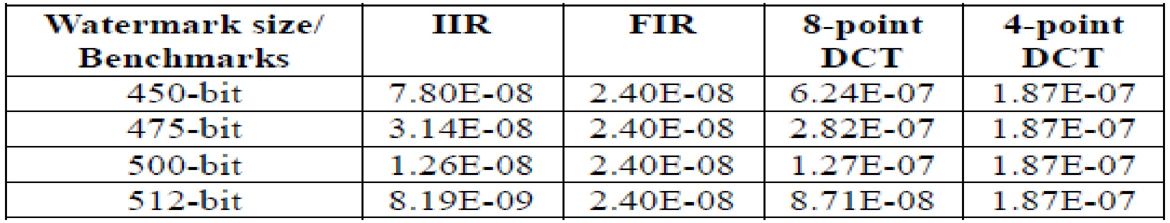
Results and Analysis

Two standard security metrics *viz.* probability of coincidence (PC) and tamper tolerance (TT)

The PC is given as: $PC = (1-(1/x))^w$

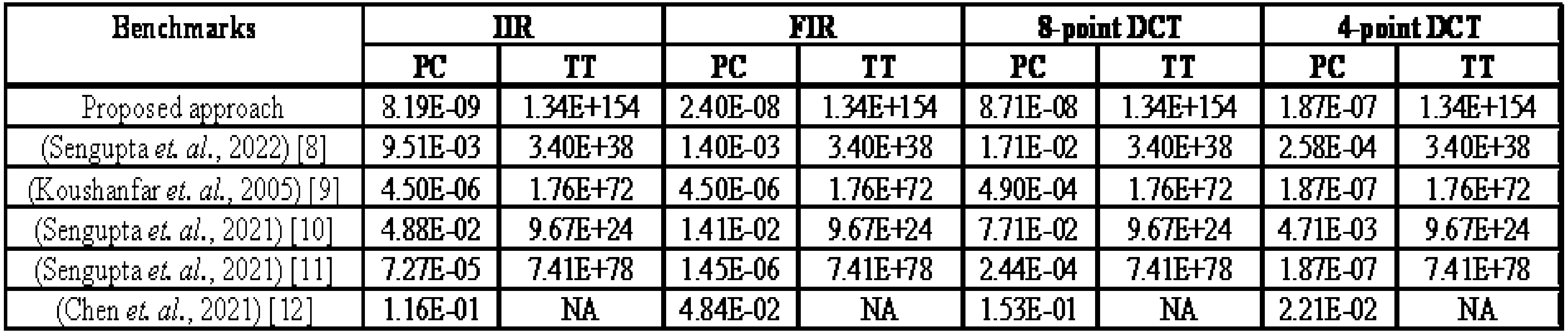
The TT is given as: $TT = Z^S$

Table II
Comparison of PC corresponding to varying watermark signature size for the proposed approach



Watermark size/ Benchmarks	IIR	FIR	8-point DCT	4-point DCT
450-bit	7.80E-08	2.40E-08	6.24E-07	1.87E-07
475-bit	3.14E-08	2.40E-08	2.82E-07	1.87E-07
500-bit	1.26E-08	2.40E-08	1.27E-07	1.87E-07
512-bit	8.19E-09	2.40E-08	8.71E-08	1.87E-07

Table III
Comparison of PC and TT between proposed approach and [8]-[12]



Benchmarks	IIR		FIR		8-point DCT		4-point DCT	
	PC	TT	PC	TT	PC	TT	PC	TT
Proposed approach	8.19E-09	1.34E+154	2.40E-08	1.34E+154	8.71E-08	1.34E+154	1.87E-07	1.34E+154
(Sengupta *et. al.*, 2022) [8]	9.51E-03	3.40E+38	1.40E-03	3.40E+38	1.71E-02	3.40E+38	2.58E-04	3.40E+38
(Koushanfar *et. al.*, 2005) [9]	4.50E-06	1.76E+72	4.50E-06	1.76E+72	4.90E-04	1.76E+72	1.87E-07	1.76E+72
(Sengupta *et. al.*, 2021) [10]	4.88E-02	9.67E+24	1.41E-02	9.67E+24	7.71E-02	9.67E+24	4.71E-03	9.67E+24
(Sengupta *et. al.*, 2021) [11]	7.27E-05	7.41E+78	1.45E-06	7.41E+78	2.44E-04	7.41E+78	1.87E-07	7.41E+78
(Chen *et. al.*, 2021) [12]	1.16E-01	NA	4.84E-02	NA	1.53E-01	NA	2.21E-02	NA

[9] F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," ACM Trans. Design Autom. Electron. Syst., vol. 10, no. 3, 523–545, Jul. 2005.
[15] W. Hu, C. Chang, A. Sengupta, S. Bhunia, R. Kastner, H. Li, "An Overview of Hardware Oriented Security and Trust: Threats, Countermeasures and Design Tools", *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, Vol. 40 (6), pp. 1010-1038, 2021.
[16] M. Potkonjak, "Methods and systems for the identification of circuits and circuit designs," *US Patent*, US7017043B1, 2006.

Results and Analysis

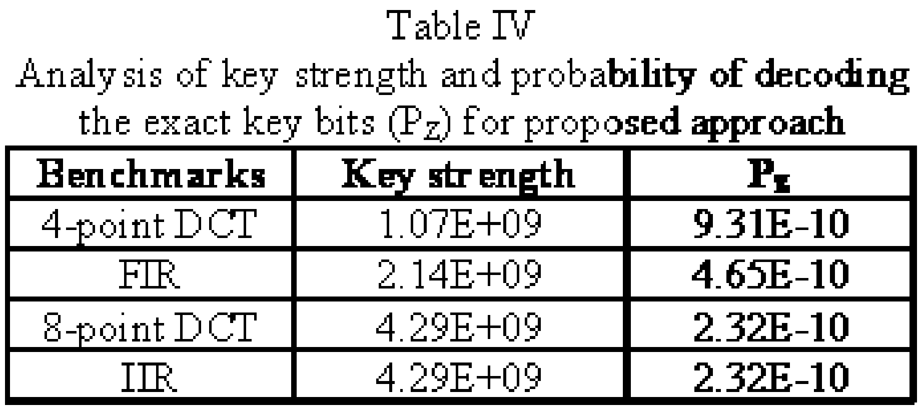


Table IV
Analysis of key strength and probability of decoding the exact key bits (P_Z) for proposed approach

Benchmarks	Key strength	P_Z
4-point DCT	1.07E+09	9.31E-10
FIR	2.14E+09	4.65E-10
8-point DCT	4.29E+09	2.32E-10
IIR	4.29E+09	2.32E-10

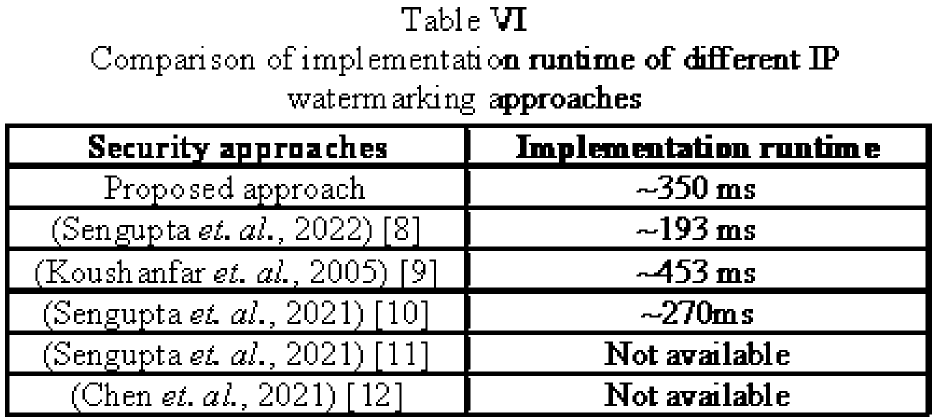


Table VI
Comparison of implementation runtime of different IP watermarking approaches

Security approaches	Implementation runtime
Proposed approach	~350 ms
(Sengupta *et. al.*, 2022) [8]	~193 ms
(Koushanfar *et. al.*, 2005) [9]	~453 ms
(Sengupta *et. al.*, 2021) [10]	~270ms
(Sengupta *et. al.*, 2021) [11]	Not available
(Chen *et. al.*, 2021) [12]	Not available

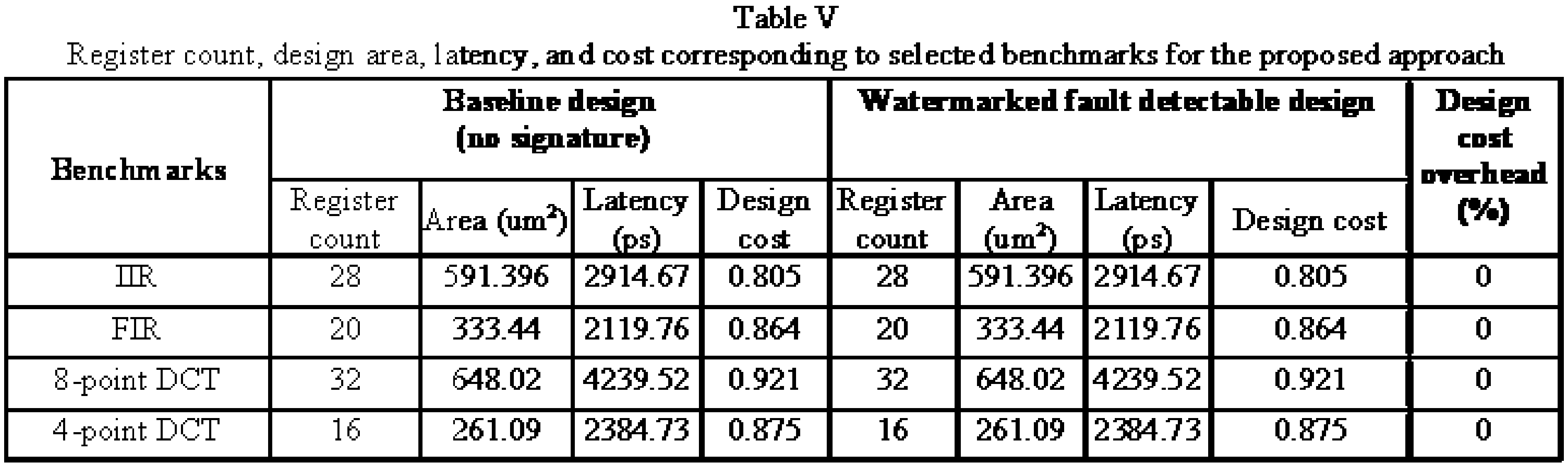


Table V
Register count, design area, latency, and cost corresponding to selected benchmarks for the proposed approach

Benchmarks	Baseline design (no signature)				Watermarked fault detectable design				Design cost overhead (%)
	Register count	Area (um^2)	Latency (ps)	Design cost	Register count	Area (um^2)	Latency (ps)	Design cost	
IIR	28	591.396	2914.67	0.805	28	591.396	2914.67	0.805	0
FIR	20	333.44	2119.76	0.864	20	333.44	2119.76	0.864	0
8-point DCT	32	648.02	4239.52	0.921	32	648.02	4239.52	0.921	0
4-point DCT	16	261.09	2384.73	0.875	16	261.09	2384.73	0.875	0

Results and Analysis

Fig. 8 and Fig 9 depicts the graphical comparison of PC and TT of proposed approach with [8]-[12]

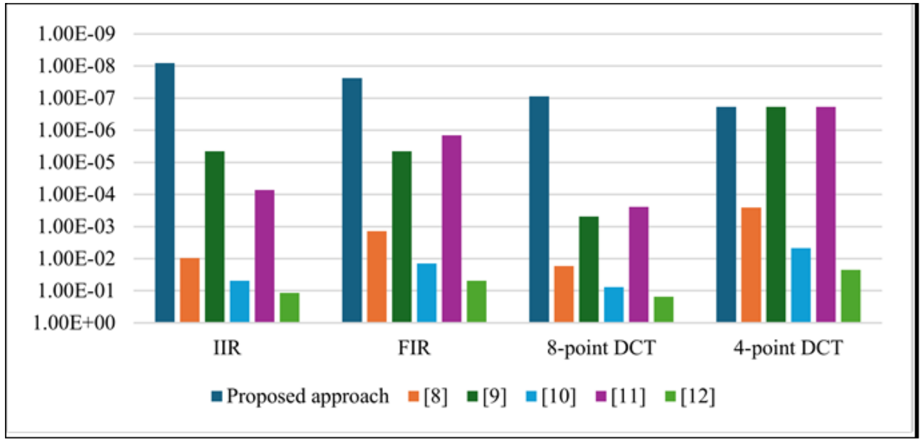


Fig. 8. Comparison of probability of coincidence (PC) between proposed approach and [8]-[12]
(*Note: Lower PC is desirable*)

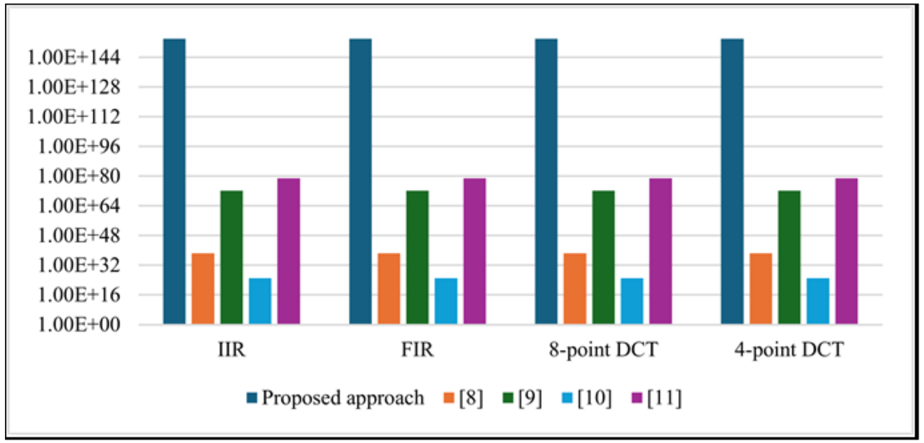


Fig. 9. Comparison of tamper tolerance (TT) between proposed approach and [8]-[11]
(*Note: Lower TT is desirable*)

Handling Different Attacks

1. **Tampering Attack**: The proposed watermarking approach has been evaluated against tampering attack. Tampering attack aims to tamper or remove the embedded watermark.

2. **Ghost Signature Search Attack (a.k.a Watermark Collision):** The proposed watermarking approach has been evaluated against watermark collision attack/ghost signature search attack.

3. **Watermark Decoding Attack:** The proposed watermarking approach has been evaluated against watermark decoding attack. In order to exactly decode and prove the embedded watermark constraints in front of third-party authenticator, an attacker needs to completely (and successfully) break all the security layers

Conclusion

This paper presented a novel hardware watermarking methodology for transient fault-detectable IP designs. The proposed methodology presents a multivariate encoded HLS scheduling based multi-modal security framework for securing fault-detectable IP designs.

References

[1] H. Pearce, R. Karri, B. Tan. 2023. High-Level Approaches to Hardware Security: A Tutorial. ACM Trans. Embed. Compu. Syst. 22, 3, Article 45, May 2023.

[2] B. C. Schafer and Z. Wang, "High-Level Synthesis Design Space Exploration: Past, Present, and Future," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 10, pp. 2628-2639, Oct. 2020.

[3] F. Koushanfar, “Hardware metering: A survey,” in Introduction to Hardware Security and Trust, Tehranipoor, M. and Wang, C. eds. New York, NY, USA: Springer, 2012.

[4] M. Shayan, K. Basu and R. Karri, "Hardware Trojans Inspired IP Watermarks," IEEE Design & Test, vol. 36, no. 6, pp. 72-79, Dec. 2019.

[5] D. Karaklajić, J. -M. Schmidt and I. Verbauwhede, "Hardware Designer's Guide to Fault Attacks," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 12, pp. 2295-2306, Dec. 2013.

[6] A. Sengupta, M. Rathor "IP Core Steganography for Protecting DSP Kernels used in CE Systems", IEEE Trans. on Consumer Electronics, Vol: 65, Issue: 4, pp. 506 – 515, Nov. 2019.

[7] M. Rostami, F. Koushanfar and R. Karri, "A Primer on Hardware Security: Models, Methods, and Metrics," Proceedings of the IEEE, vol. 102, no. 8, pp. 1283-1295, Aug. 2014.

[8] A. Sengupta and R. Chaurasia, "Securing IP Cores for DSP Applications Using Structural Obfuscation and Chromosomal DNA Impression," IEEE Access, vol. 10, pp. 50903-50913, 2022.

[9] F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," ACM Trans. Design Autom. Electron. Syst., vol. 10, no. 3, 523–545, Jul. 2005.

[10] A. Sengupta and M. Rathor, "Facial Biometric for Securing Hardware Accelerators," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 1, pp. 112-123, Jan. 2021.

[11] A. Sengupta, R. Chaurasia and T. Reddy, "Contact-Less Palmprint Biometric for Securing DSP Coprocessors Used in CE Systems," IEEE Trans. on Consumer Electronics, vol. 67, no. 3, pp. 202-213, 2021.

[12] J. Chen and B. C. Schafer, "Watermarking of Behavioral IPs: A Practical Approach," 2021 IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), France, pp. 1266-1271, 2021.

[13] OCL, 15 nm open cell library. [Online], Available: https://si2.org/open-cell-library/, last accessed on Feb. 2024.

[14] Express Benchmark Suite, University of California Santa Barbara (UCSB), Available: http://www.ece.ucsb.edu/EXPRESS/benchmark/, Accessed: Feb. 2025.

[15] W. Hu, C. Chang, A. Sengupta, S. Bhunia, R. Kastner, H. Li, "An Overview of Hardware Oriented Security and Trust: Threats, Countermeasures and Design Tools", IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., Vol. 40 (6), pp. 1010-1038, 2021.

[16] M. Potkonjak, “Methods and systems for the identification of circuits and circuit designs,” US Patent, US7017043B1, 2006.

[17] A. Sengupta, R. Sedaghat “Swarm Intelligence Driven Design Space Exploration of Optimal k-Cycle Transient Fault Secured Datapath during High Level Synthesis Based on User Power-Delay Budget", Elsevier Journal on Microelectronics Reliability, Volume 55, Issue 6, May 2015, pp. 990-1004, March 2015.

[18] A. Sengupta and D. Kachave, "Spatial and Temporal Redundancy for Transient Fault-Tolerant Datapath," IEEE Transactions on Aerospace and Electronic Systems, vol. 54, no. 3, pp. 1168-1183, June 2018. [19] “Single Event Upsets”, Intel, Feb. 2025. Available: https://www.intel.com/content/www/us/en/support/programmable/support-resources/quality/seu.html.

[20] O. Benot, Fault attack, in: H.C.A. van Tilborg, S. Jajodia (Eds.), Encyclopedia of Cryptography and Security, Springer, Boston, MA, 2011, https://doi.org/10.1007/ 978-1-4419-5906-5_505.

[21] S. Rai, A. Rupani, P. Nath and A. Kumar, “Hardware Watermarking Using Polymorphic Inverter Designs Based On Reconfigurable Nanotechnologies,” 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2019, pp. 663-669.

[22] R. Karmakar, S. S. Jana and S. Chattopadhyay, “A Cellular Automata Guided Finite-State-Machine Watermarking Strategy for IP Protection of Sequential Circuits,” IEEE Trans. Emerg. Topics Comput., vol. 10, no. 2, pp. 806-823, 1 April-June 2022.

[23] B. Colombier and L. Bossuet, "Survey of hardware protection of design data for integrated circuits and intellectual properties,' IET Comput. Digit. Techn., vol. 8, no. 6, pp. 274–287, 2015.

[24] A. Cui, C. Chang, S. Tahar and A. T. Abdel-Hamid, "A robust FSM watermarking scheme for IP Protection of sequential circuit design," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 30, no. 5, pp. 678-690, 2011.

[25] M. Yasin, J. J. Rajendran, O. Sinanoglu, and R. Karri, "On improving the security of logic locking," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 35, no. 9, 1411–1424, 2016.

[26] F. Koushanfar et al., "Can EDA combat the rise of electronic counterfeiting?," DAC Design Automation Conference 2012, San Francisco, CA, USA, 2012, pp. 133-138.

[27] S. M. Plaza and I. L. Markov, "Solving the Third-Shift Problem in IC Piracy With Test-Aware Logic Locking," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 6, pp. 961-971, June 2015.

[28] A. Sengupta, S. Mohanty, "Advanced Encryption Standard (AES) and its Hardware Watermarking for Ownership Protection", IET, IP Core Protection and Hardware-Assisted Security for Consumer Electronics, 2019, Book ISBN: 978-1-78561-799-7, e-ISBN: 978-1-78561-800-0.

[29] C. Paar, J. Pelzl “Understanding Cryptography - A Textbook for Students and Practitioners”, Springer-Verlag, eBook ISBN 978-3-642-04101-3, Number of Pages: XVIII, 372, Nov 2009.

THANK YOU