knossosnet research group

System-On-Chip (SoC) Design Primitives for Wireless Networks Security

Nicolas Sklavos

KNOSSOSnet Research Group,
Technological Educational Institute of Western Greece
e-mail: nsklavos@ieee.org

Just a Few Seconds…

AT THE BEGGING !!!

Special Thanks to:

- Summer School Organizing Committee
- Professor Lejla Batina !

KNOSSOSnet Name Inspiration

KNOSSOSnet name is inspired from "Knossos": (Source Wikipedia)

Knossos (alternative spellings Knossus, Cnossus, Greek Κνωσσός, pronounced [kno'sos]), currently refers to the main Bronze Age archaeological site at Heraklion, a modern port city on the north central coast of Crete. Heraklion was formerly called Candia after the Saracen name for the place, Kandaiki, meaning the moat that was built around the then new settlement for defense. Kandaiki became Byzantine Chandax.

KNOSSOSnet Research Group

- Faculty Members
- Researchers & Research Associate Members
- Students
- Visitors from Other Institutes
- More Information: www.KNOSSOSnet.gr

You are More Than Welcome !!!

- Internships, Short Term Missions, ERASMUS+, etc.
- Research Collaborator: R & D Project(s).
- Visitor Researchers: Teaching, Research & Development Activities.
- Teaching opportunities: postgraduate and undergraduate courses.
- Great Time to Greece !!!

… Just Drop a Line : info @ KNOSOSSnet.gr

“I”, “O”, … LETS GO !

Lecture Outline

Part I: 10%

- INTRODUCTION TO CRYPTOGRAPHY

Part II: 40%

- ARCHITECTURES & SYSTEM ON CHIP (SoC) DESIGNS

Part III: 40%

- IMPEMENTATION PLATFORMS FOR SoC

Part IV: 5%

- PUBLISHED RESEARCH RESULTS

Part V: 5%

- CONCLUSIONS & OUTLOOK

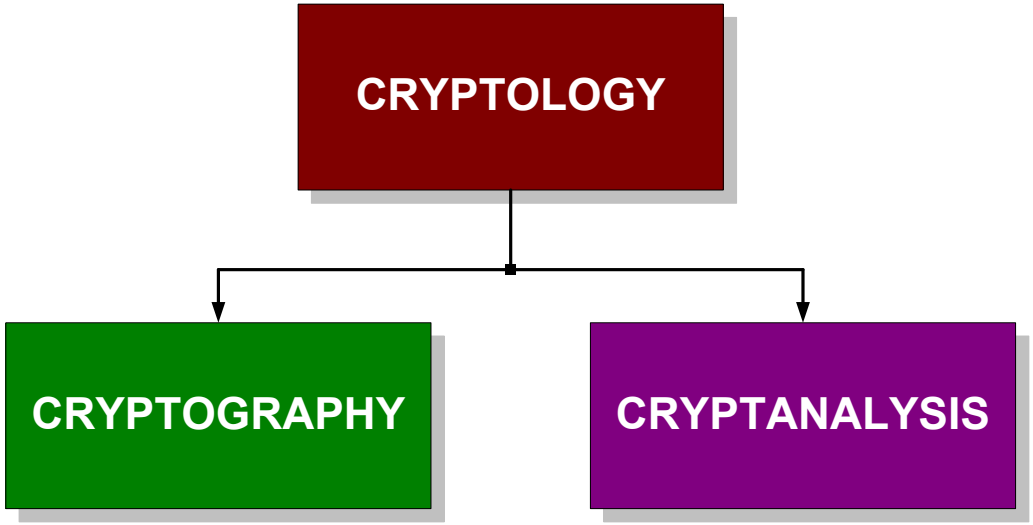
PART I

INTRODUCTION TO CRYPTOGRAPHY

Basic Terms

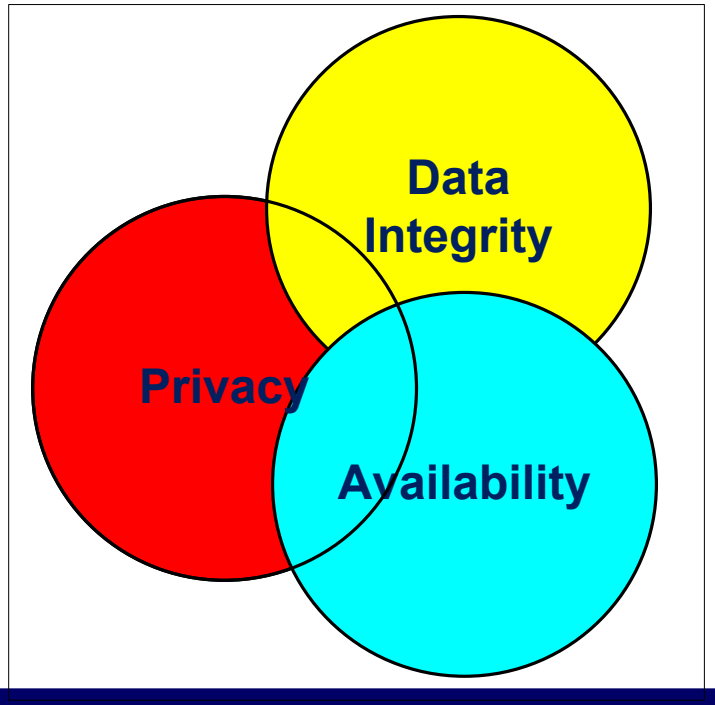
SECURITY & PRIVACY ISSUES

Introduction to Security



Basic Aims of Cryptography

- Confidentiality , Privacy
- Data Integrity
- Availability

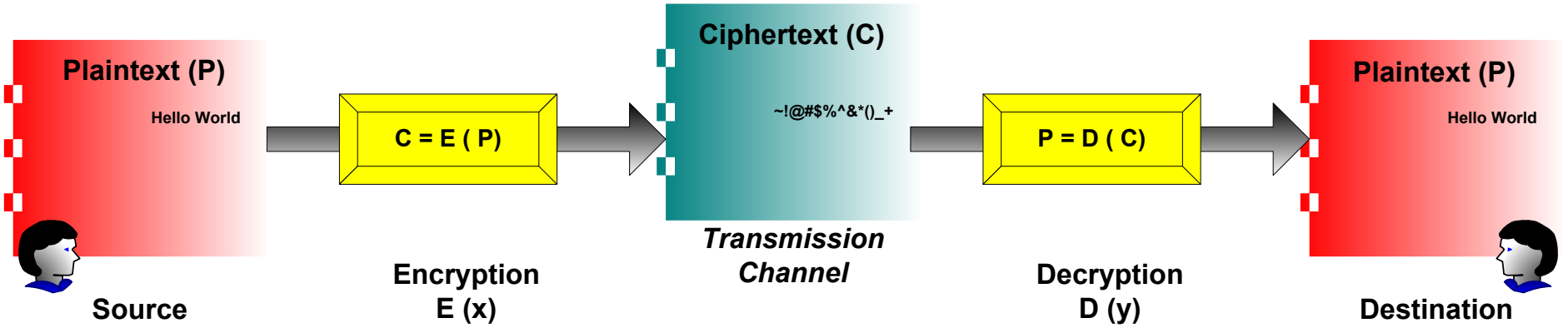


CRYTPOLOGY: distinguished chapter of Mathematics Science.

CRYPTOGRAPHY: ensures the privacy of data transmission.

CRYPTANALYSIS: recovers the information of the transmitted ciphertext.

Cryptography Schematic



Encryption Algorithms:

- ✓ mathematical function, based on also on other kinds of data transformation: digital, logic, S-BOXs.
- ✓ encryption/ decryption
- ✓ use of keys

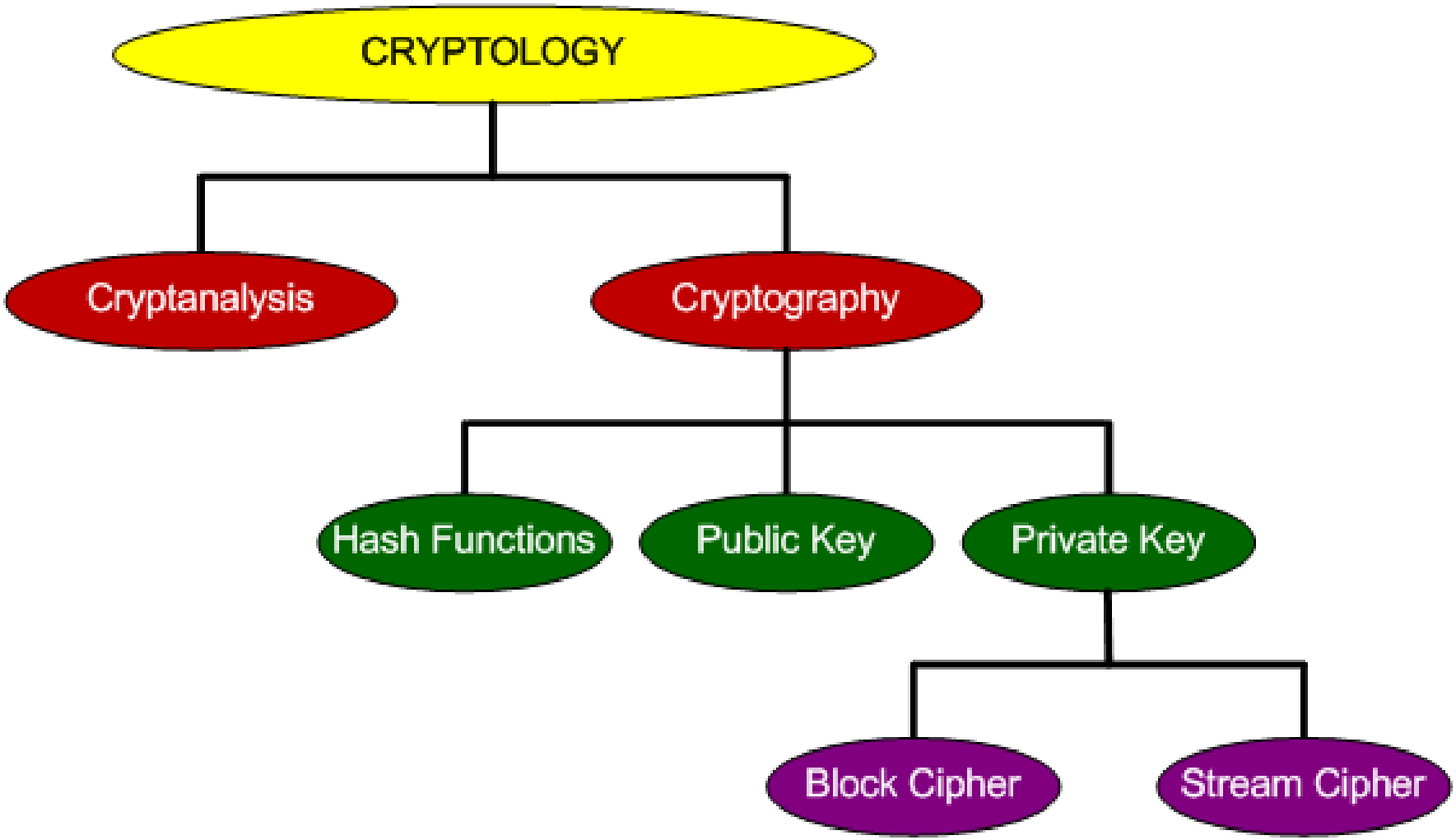
1. E {Key1, Plain} = Cipher
2. D {Key2, Cipher} = Plain

   D [Key2, E {Key1, Plain}] = Plain

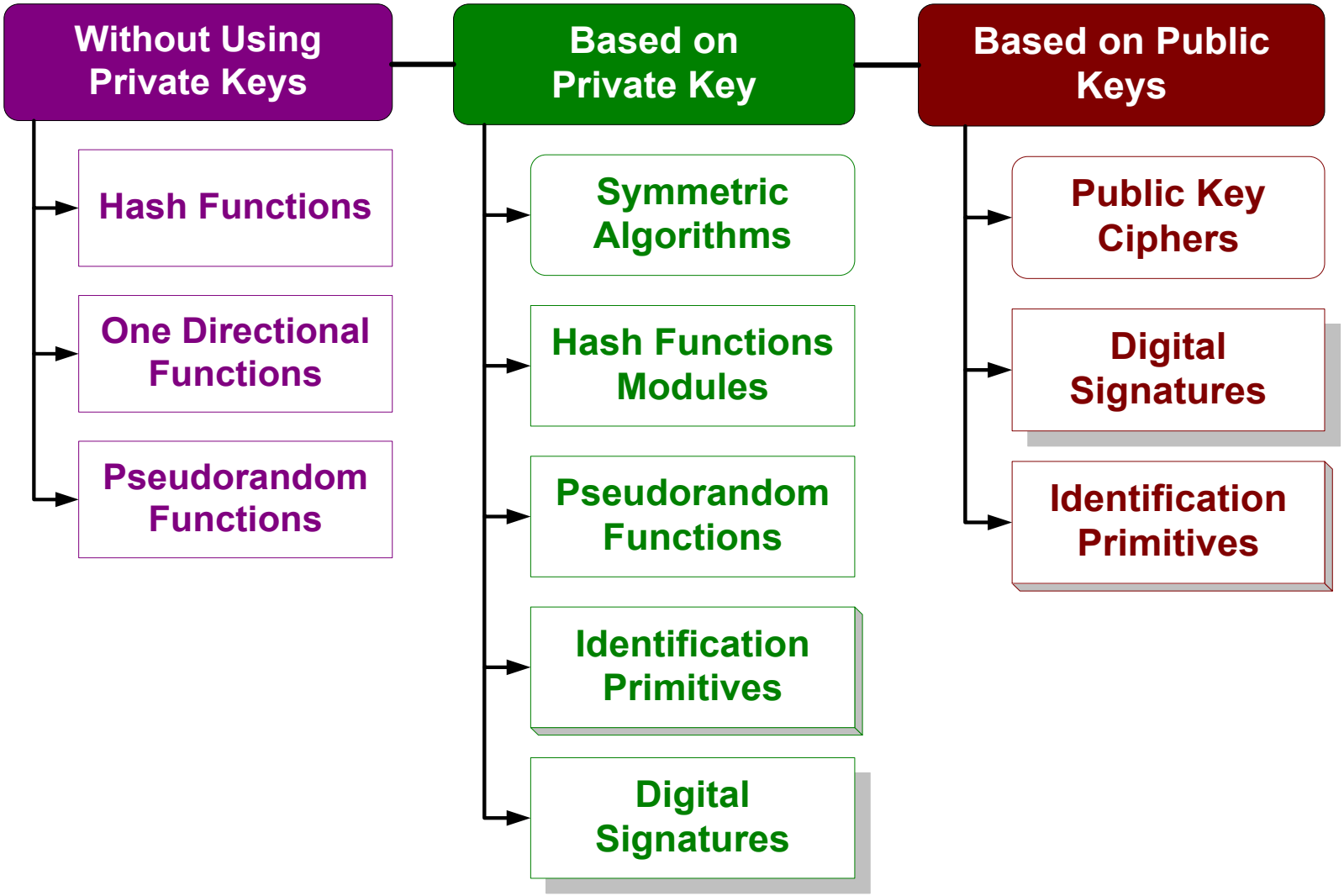
Examples of Encryption Algorithms

- Caesar's
- Enigma
- DES
- IDEA
- RSA
- AES

Cryptology & Ciphers



Security Schemes Categories



Modes of Operation

Cryptography Operation Modes

1. *Electronic Codebook* (**ECB**)
2. *Cipher Block Chaining Mode* (**CBC**)
3. *Cipher Feedback Mode* (**CFB**)
4. *Output Feedback Mode* (**OFB**)
5. *Counter Mode* (**CTR**)
6. *Cipher Block Chaining-Message Authentication Code Mode* (**CBC-MAC**)

Implementation Operation Modes:

- ✓ Non Feedback: **ECB** and **CTR**
- ✓ Feedback: **CBC**, **CFB**, **OFB**, **CBC-MAC**

Implementation Aspects

Security Services:

- **Bulk Encryption:** encryption of data, before enter the transmission channel, or network.

- **User Authentication:** usage of available sources only from the authenticated users.

- **Data Integrity:** protects from data modification during the message transmission through channel.

Implementation Issues :

1) **IP Copyrights – Permission of Use**
2) **Support of the requested Security Level of the Scheme**
3) **Easy to be used by the users**
4) **Flexibility of the Implementation**
   - ✓ ***Maximized of speed for the data transformation***
   - ✓ ***Minimized Allocated System Resources***

Six Layers Model

Layer 6	**Applications:** *e-mail, e-commerce, firewalls etc.*
Layer 5	**Authentications Protocols:** *SSL/TLS/WTLS, IPSEC, IEEE 802.11*
Layer 4	**Security Services:** *Confidentiality, Authentication, Integrity, Non-Repudiation*
Layer 3	**Cryptographic Primitives:** *Encryption, Decryption, Digital Signature*
Layer 2	**Public & Private Key Cryptography:** *Encryption Algorithms*
Layer 1	**Computer Arithmetic:** *Digital & Arithmetic Logic*

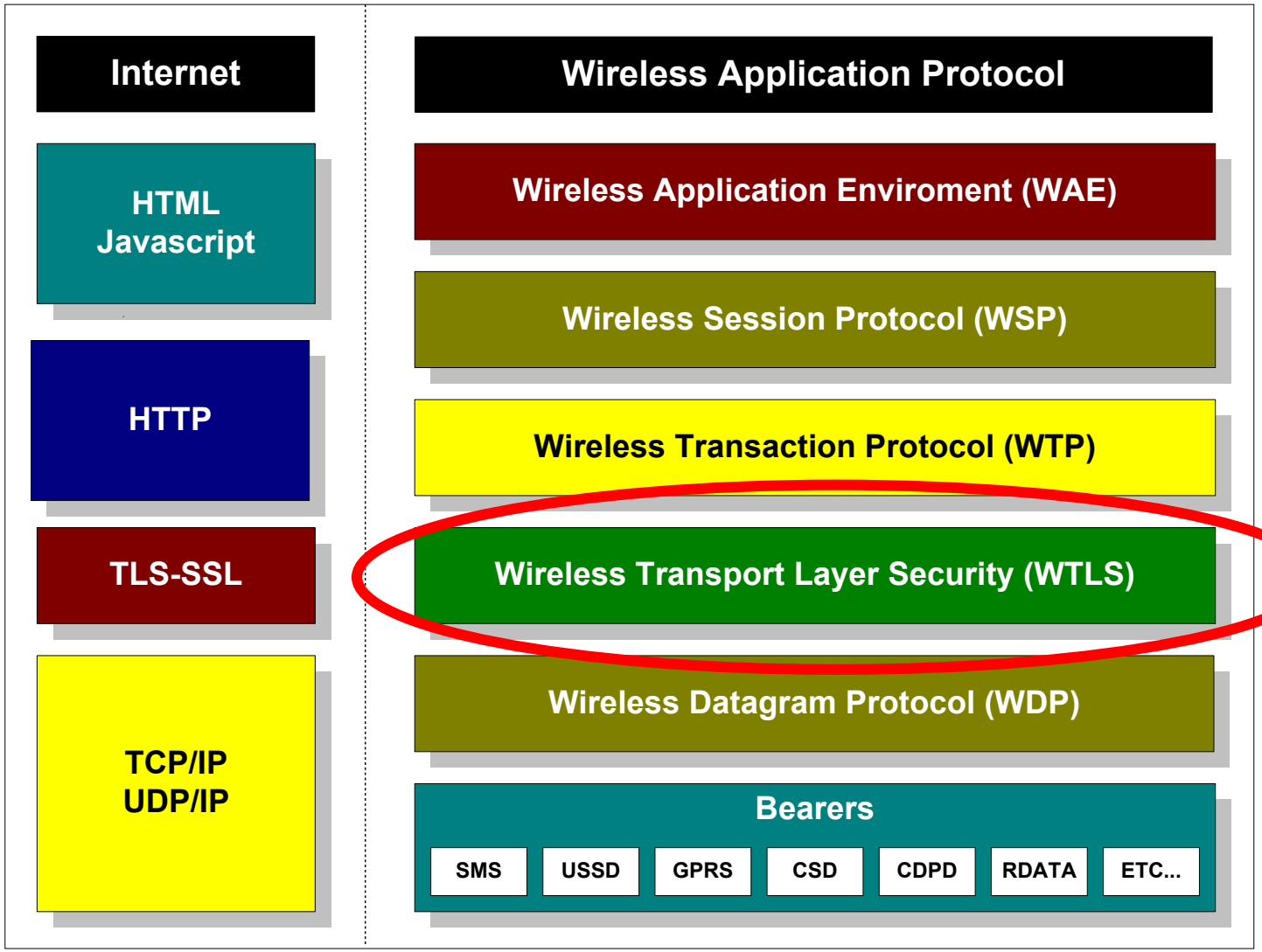
PART II

ARCHITECTURES FOR CRYPTO-ON-A-CHIP SYSTEMS DESIGN

Security Systems with High Complexity

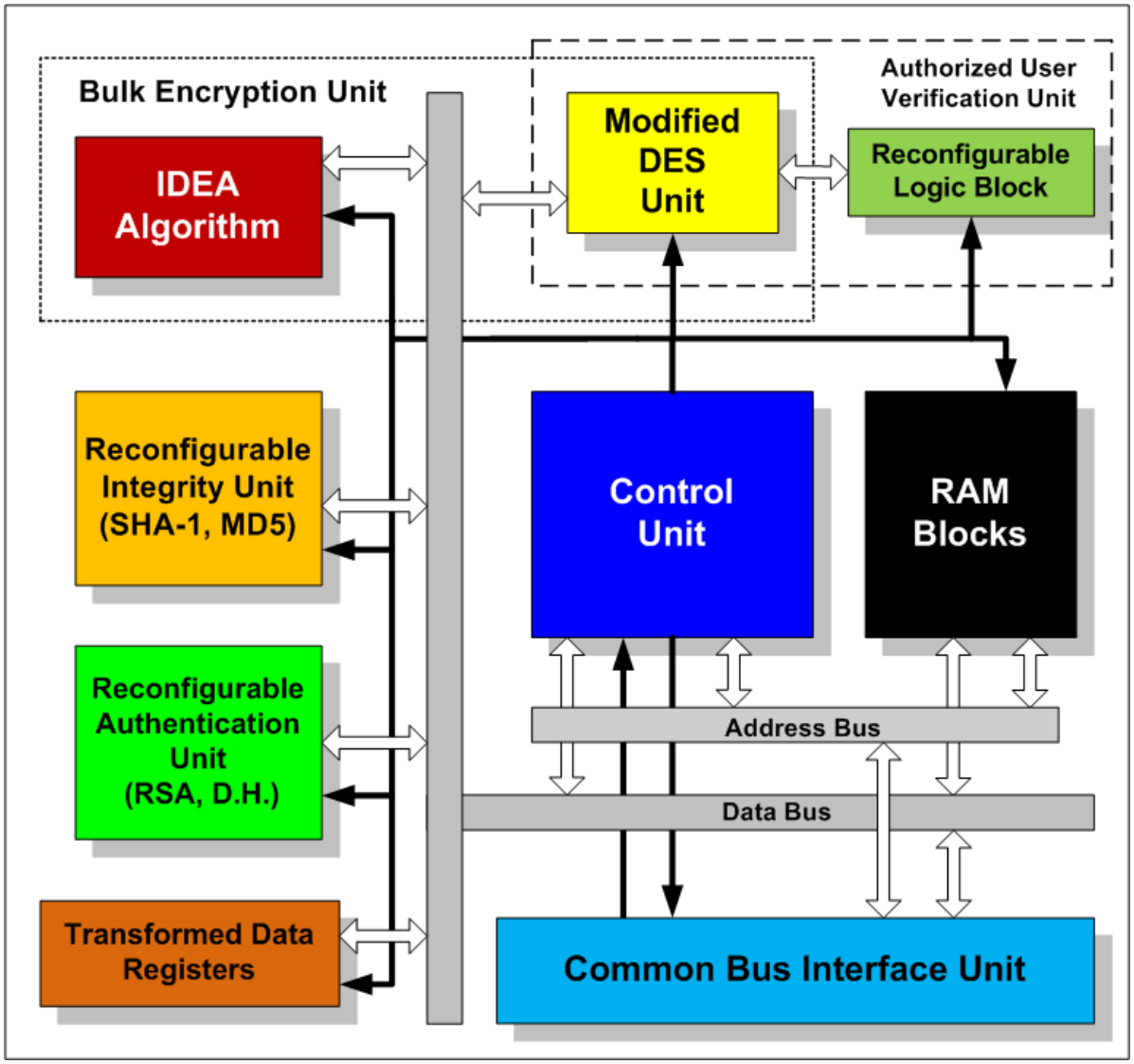
WIRELESS TRANSPORT LAYER SECURITY: DES, IDEA, SHA-1, RIPMEND, RC5 …

A "Huge" Security System

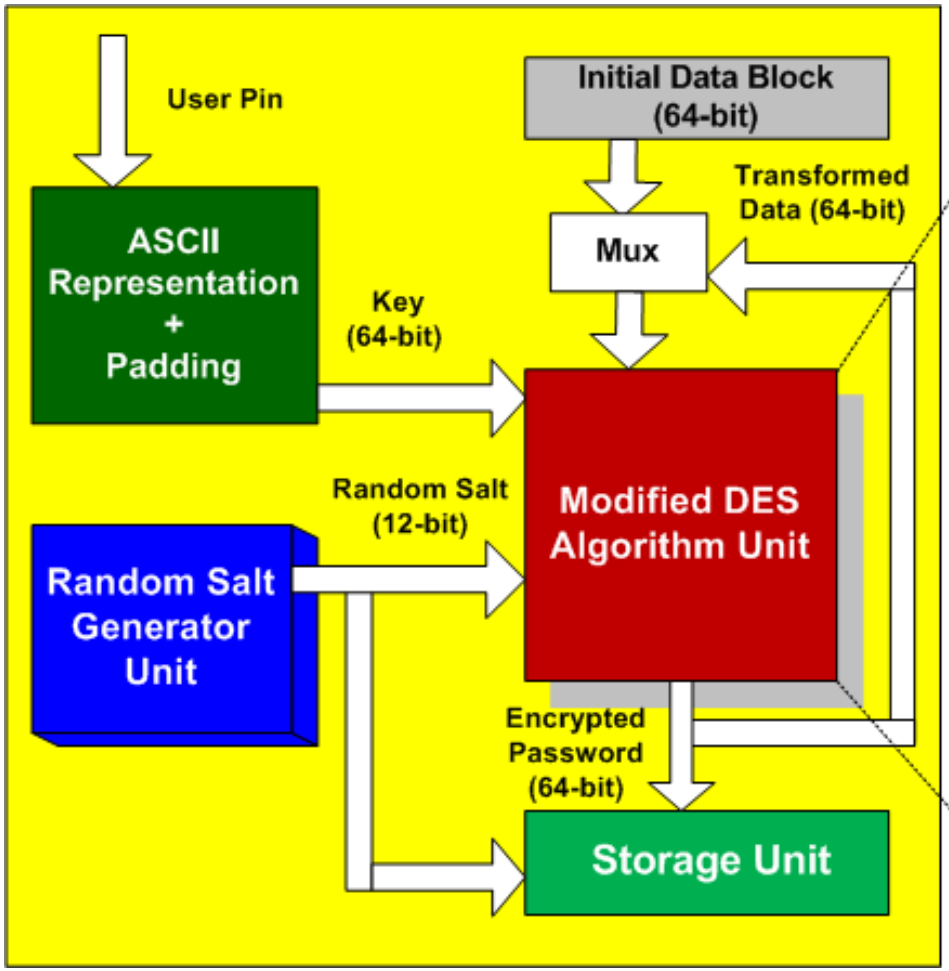


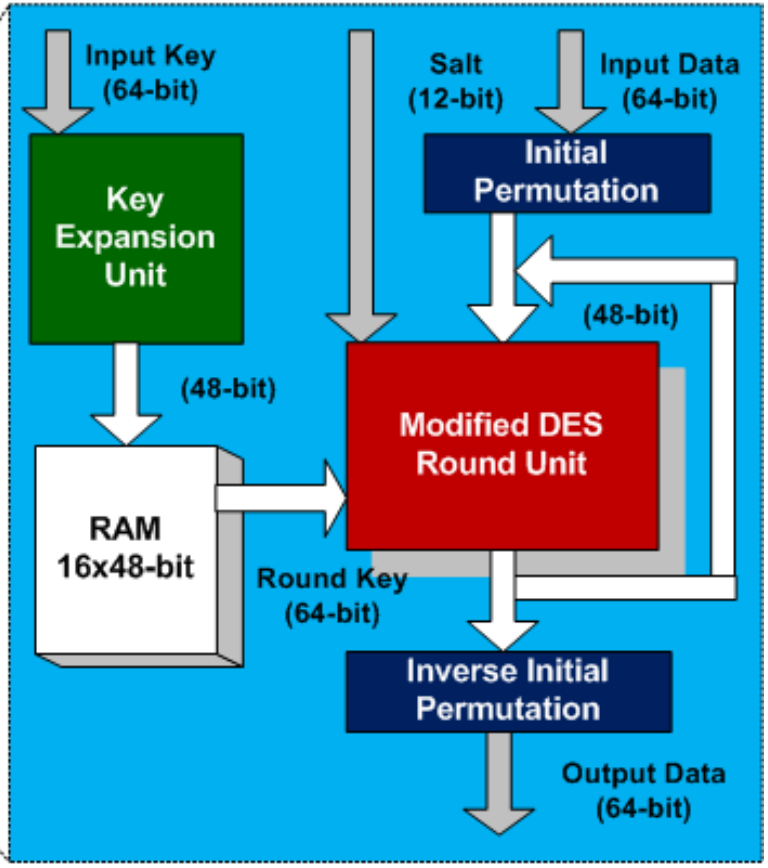
WTLS Proposed Co-Processor

- Proposed Privacy Unit:
  - i) IDEA Unit
  - ii) Authorization Unit
  - iii) RC5 Unit
- Authentication Unit
- Data Integrity Unit
- Supported Functions
  1) HMAC
  2) Digital Signature
  3) Random Number Generator



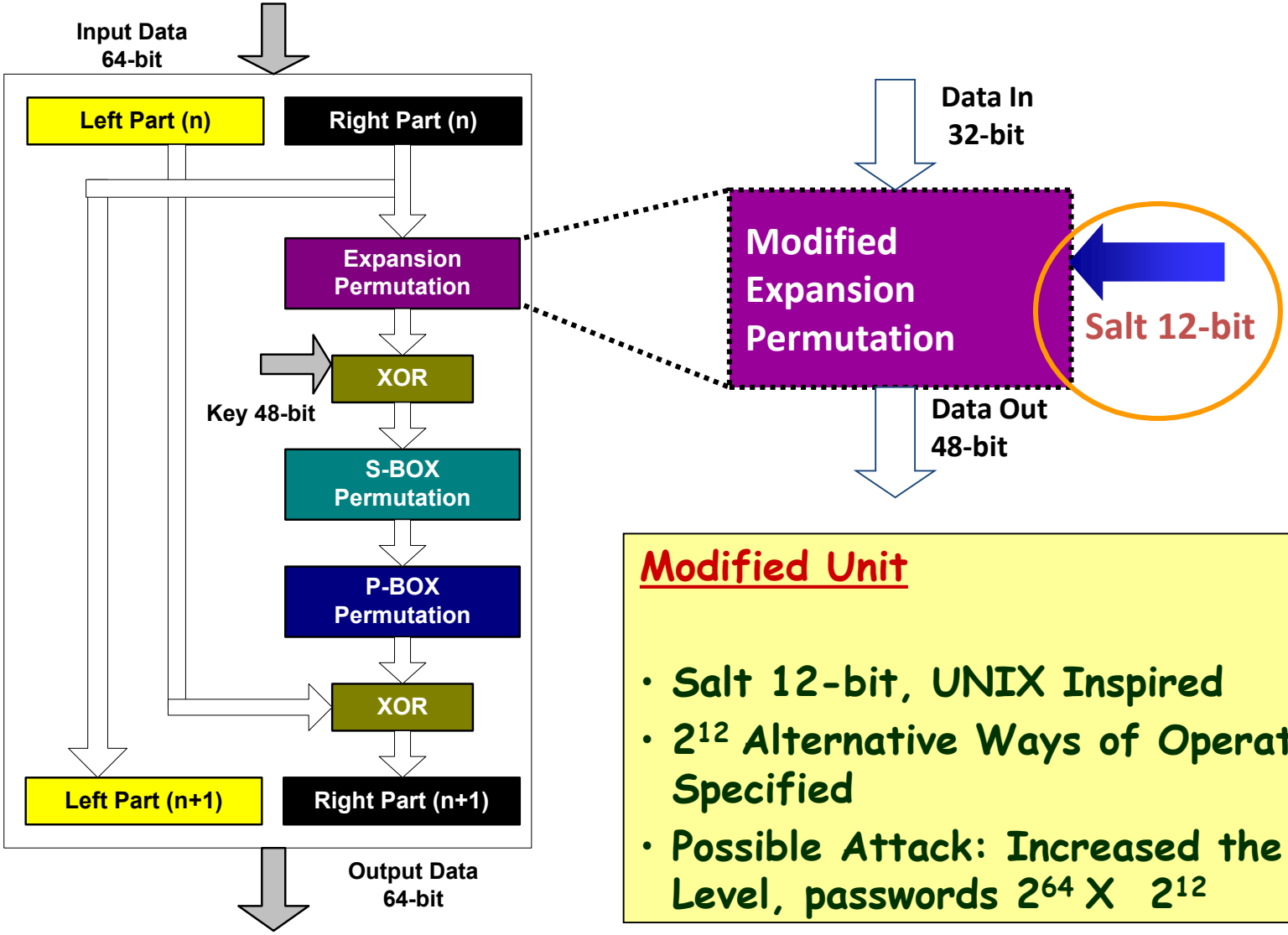
DES Algorithm Architecture Design





- Feedback Loop Design Technique: Performance, Area
- Co-Processor Design: Number of Clock Cycles, Routing

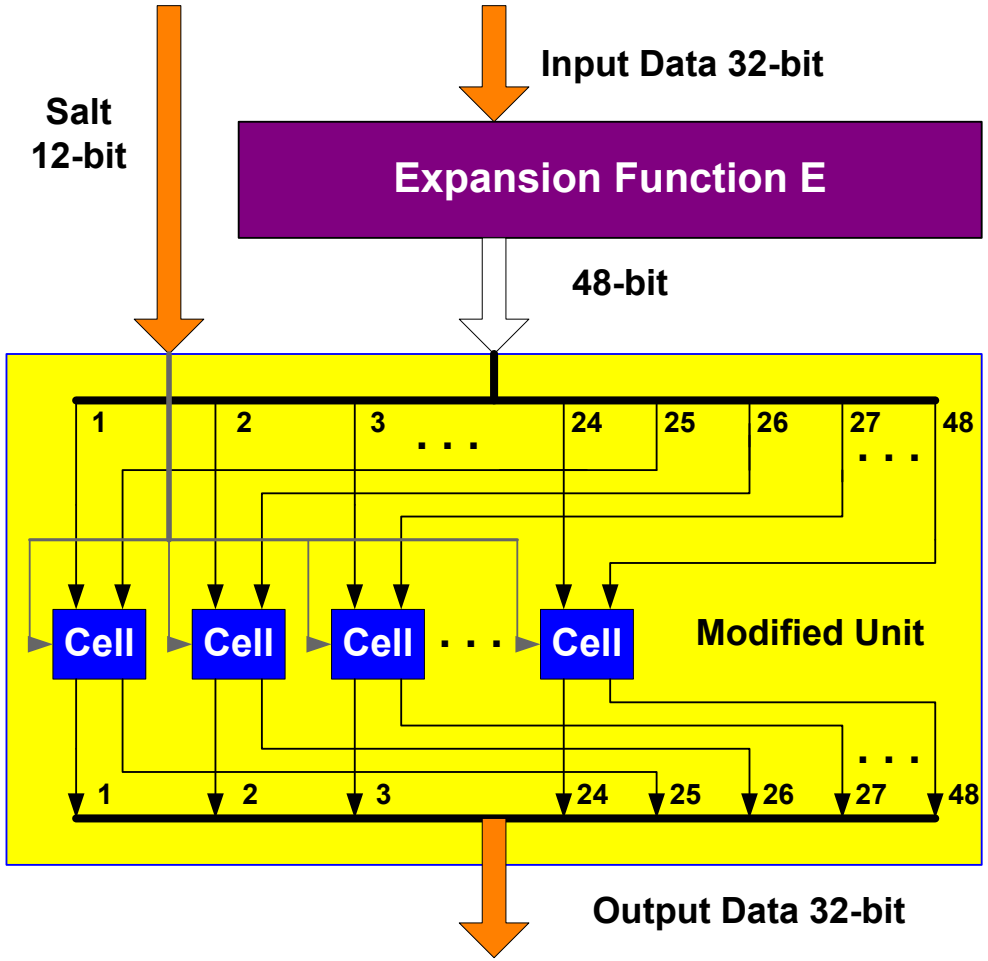
Modified DES Unit Design



Modified Unit

- Salt 12-bit, UNIX Inspired
- 2^{12} Alternative Ways of Operation are Specified
- Possible Attack: Increased the Security Level, passwords $2^{64} \times 2^{12}$

Modified Expansion Unit



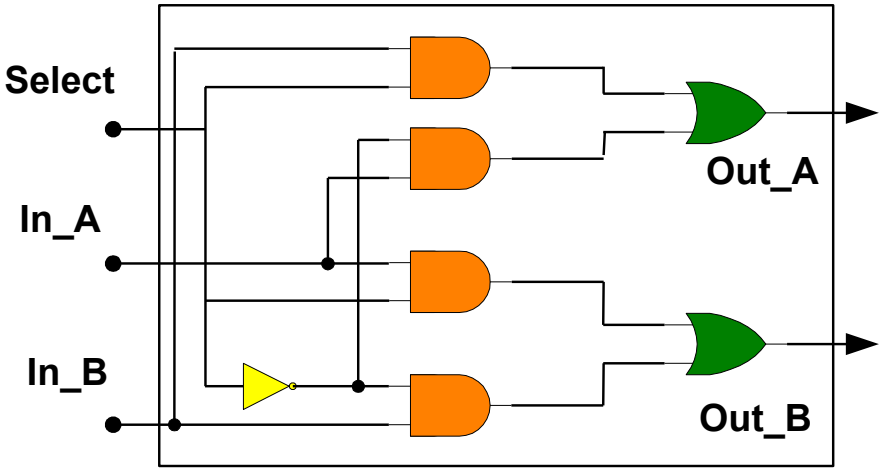
Cells

i) Combination of Two Bits of the Output for Function E

ii) 7 Gates: 4xAND, 2xOR, 1xInv

iii) 24 Used Cells in total

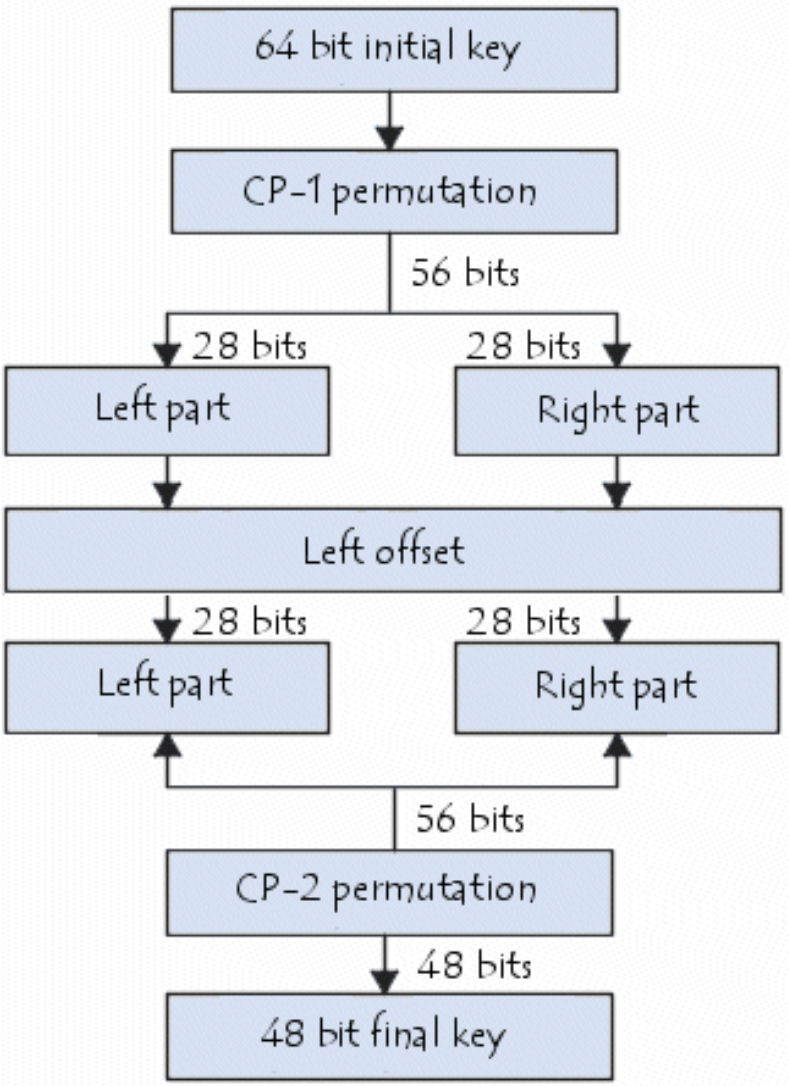
iv) Vector Salt, 12-bit



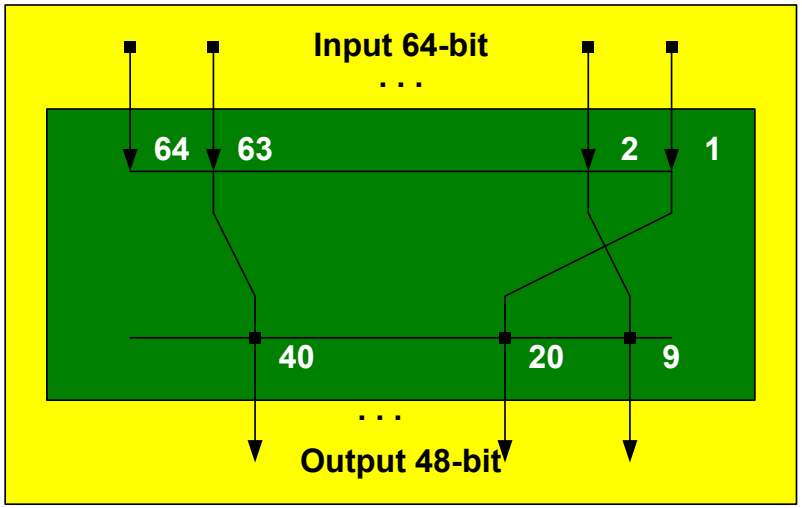
Key Generation Specifications

Specifications:

1. 64-bit Initial Key
2. CP First Permutation
3. Key Partitioning
4. Left Offset
5. Left & Right Parts
6. CP Last Permutation
7. Final Key

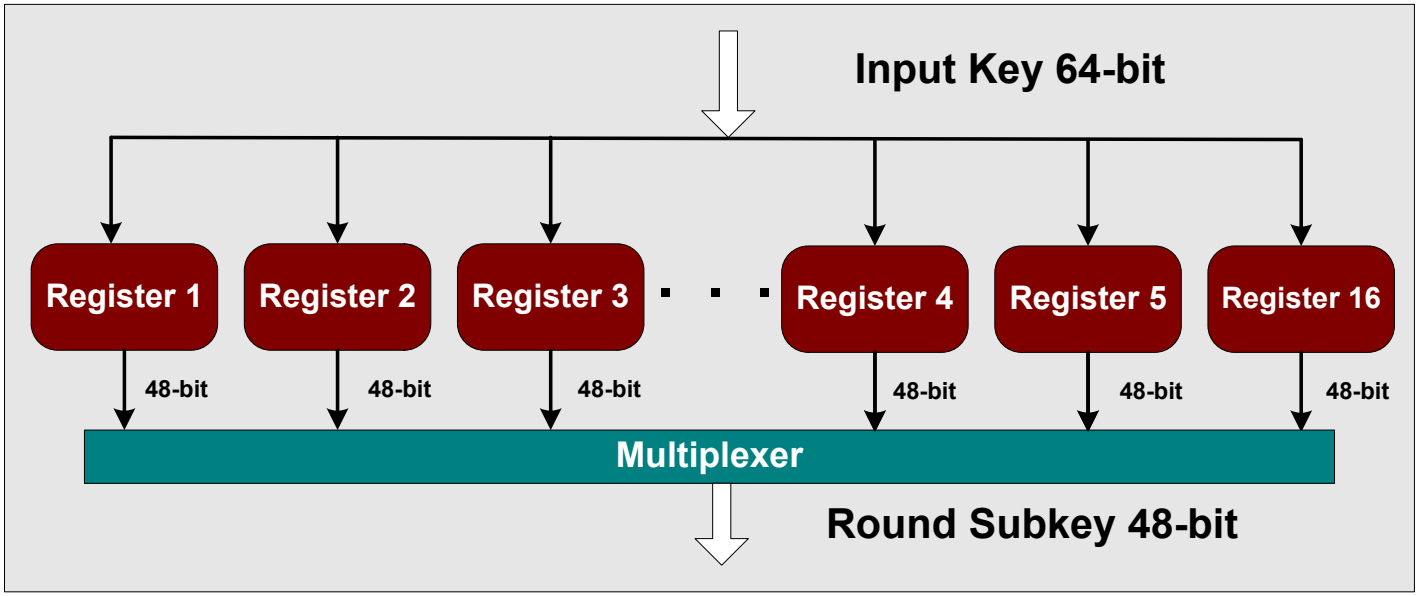


Proposed Key Generation

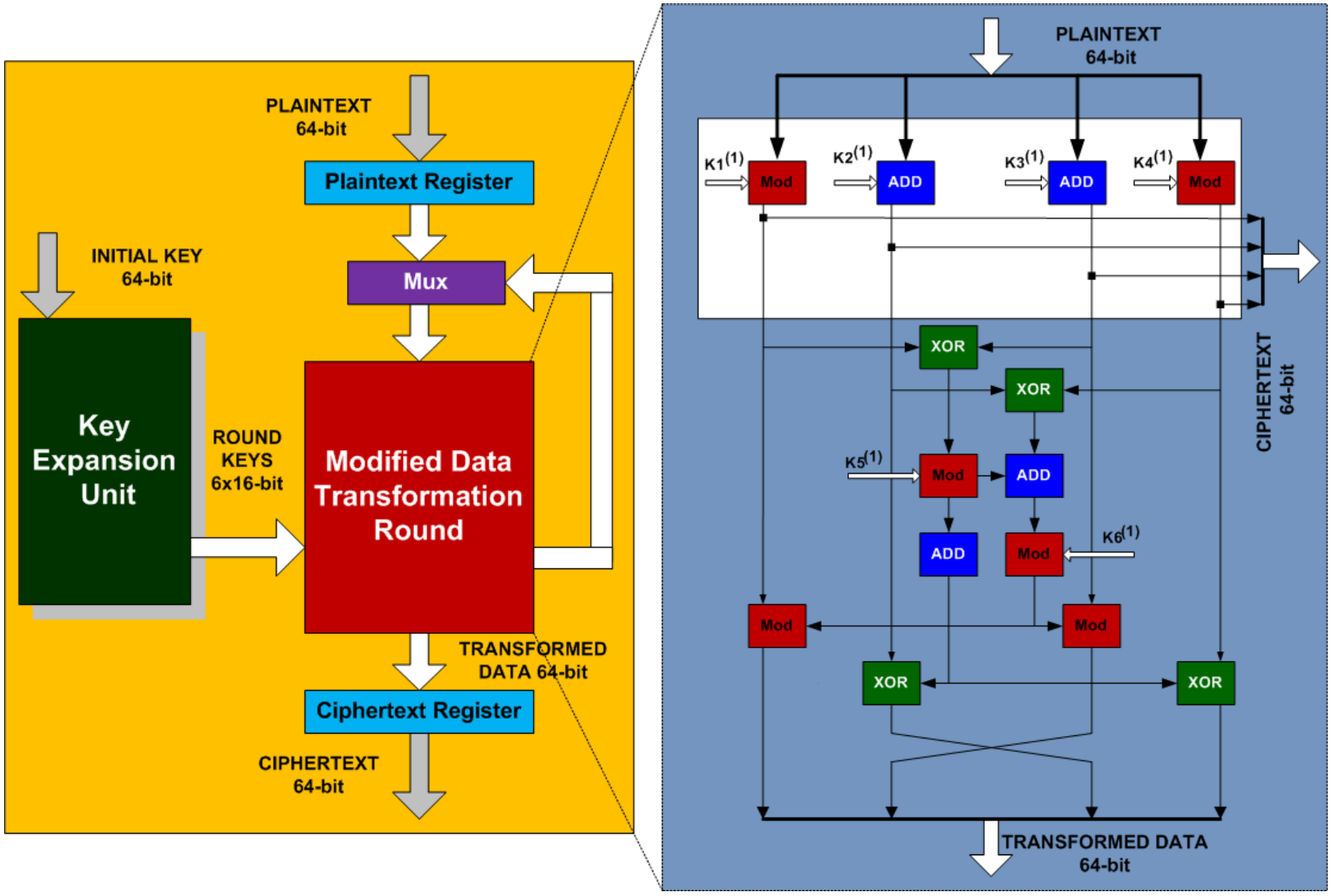


Proposed Architecture

- Specifications Analysis => 16 Registers/Multiplexers
- Area Increase at 5%
- Increase Performance at 70%



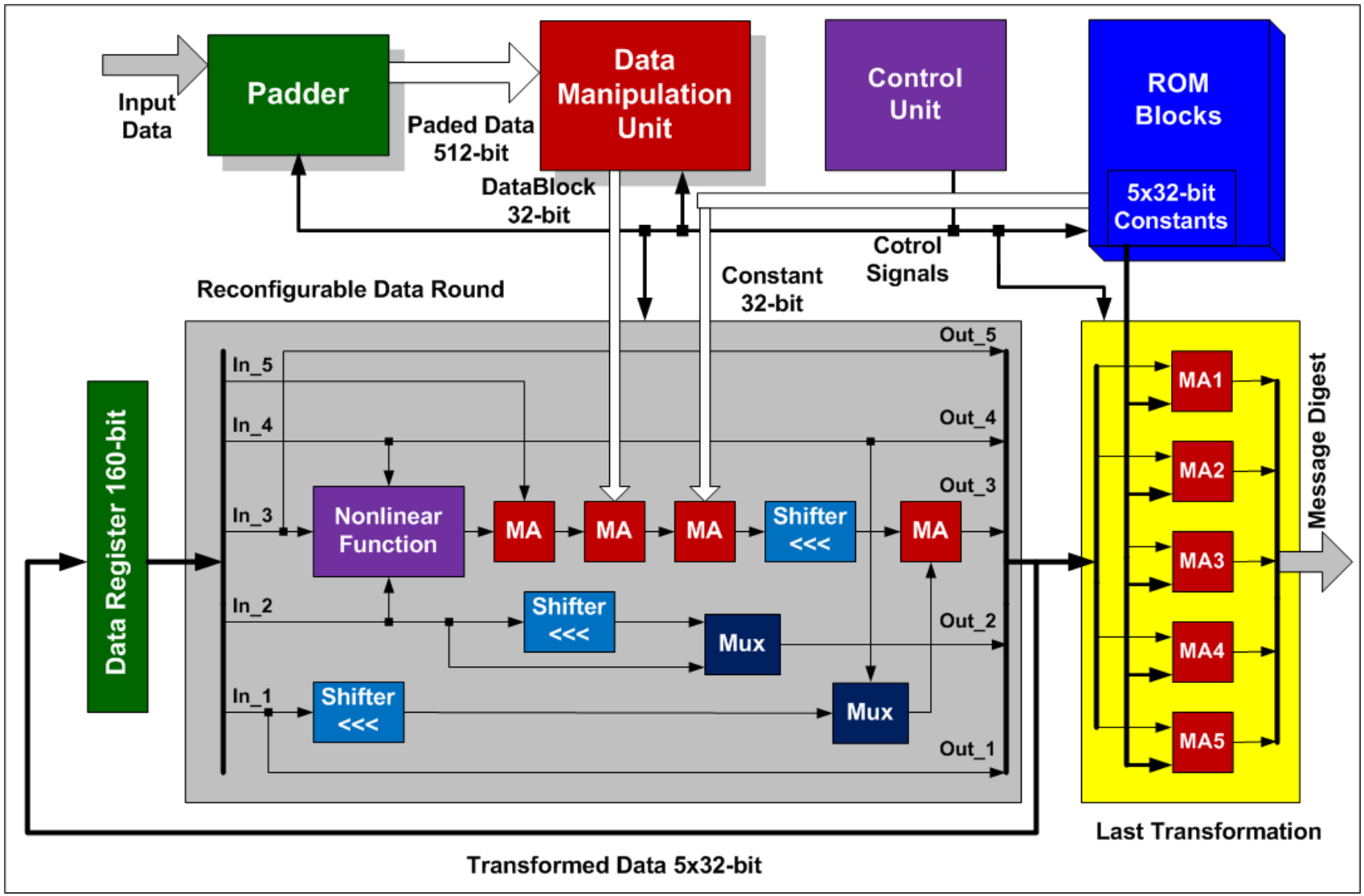
IDEA Encryption Algorithm



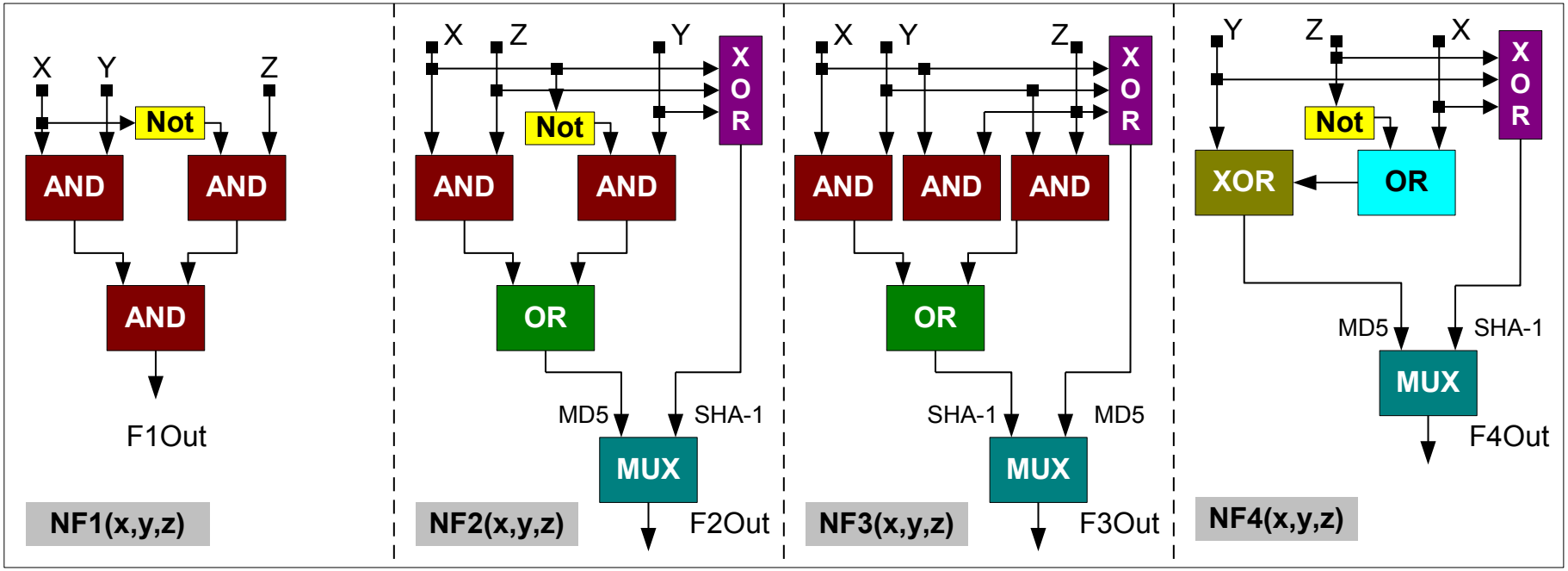
Implementation Comparison

Implementation Parameters	Conventional	Proposed	Comparison
Used Units	6 x MM 2^{16}+1 8 x MA 2^{16} 4 x XOR 16-bit -	4 x MM 2^{16}+1 6 x MA 2^{16} 4 x XOR 16-bit 1 x Regist. 64-bit	- (2 x MM 2^{16}+1) - (2 x MA 2^{16}) - 1 x Regist. 64-bit
Area	A	B	B=(0.6-0.7) A -(30-40 %)
T_{delay}	{ 3 x T_{delay}(MM) + 2 x T_{delay}(MA) + 2 x T_{delay}(XOR}	2 x T_{delay}(MM) + 2 x T_{delay}(MA) + 1 x T_{delay}(XOR)	-{1 x T_{delay}(MM) + 1 x T_{delay}(XOR)}
Frequency	F1	F2	F2=1.3 x F1 (30%)
Throughput	64 bit / 9 clock cycles	2x64 bit / 10 clock cycles	+ 45 %

Data Integrity Unit: SHA-1, MD5

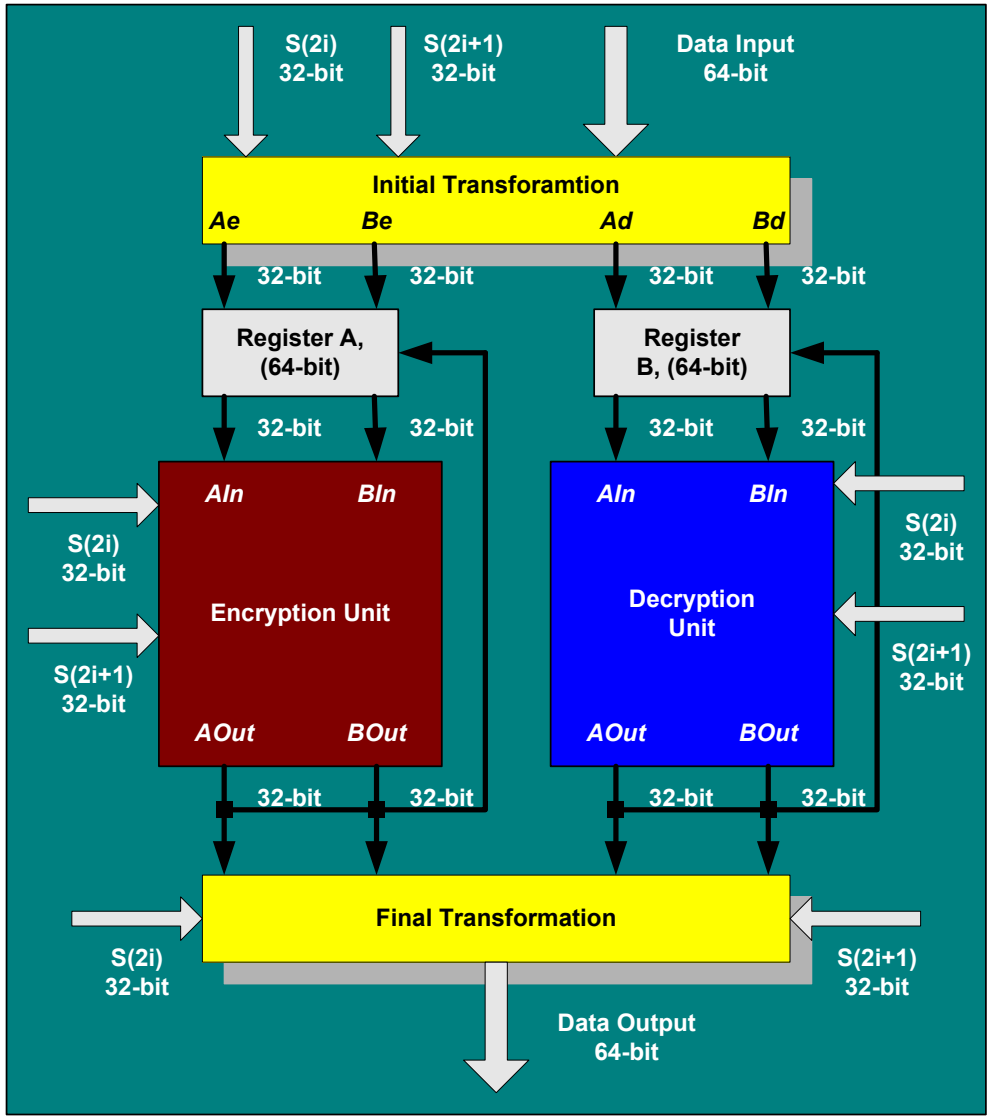


Nonlinear Functions Fi

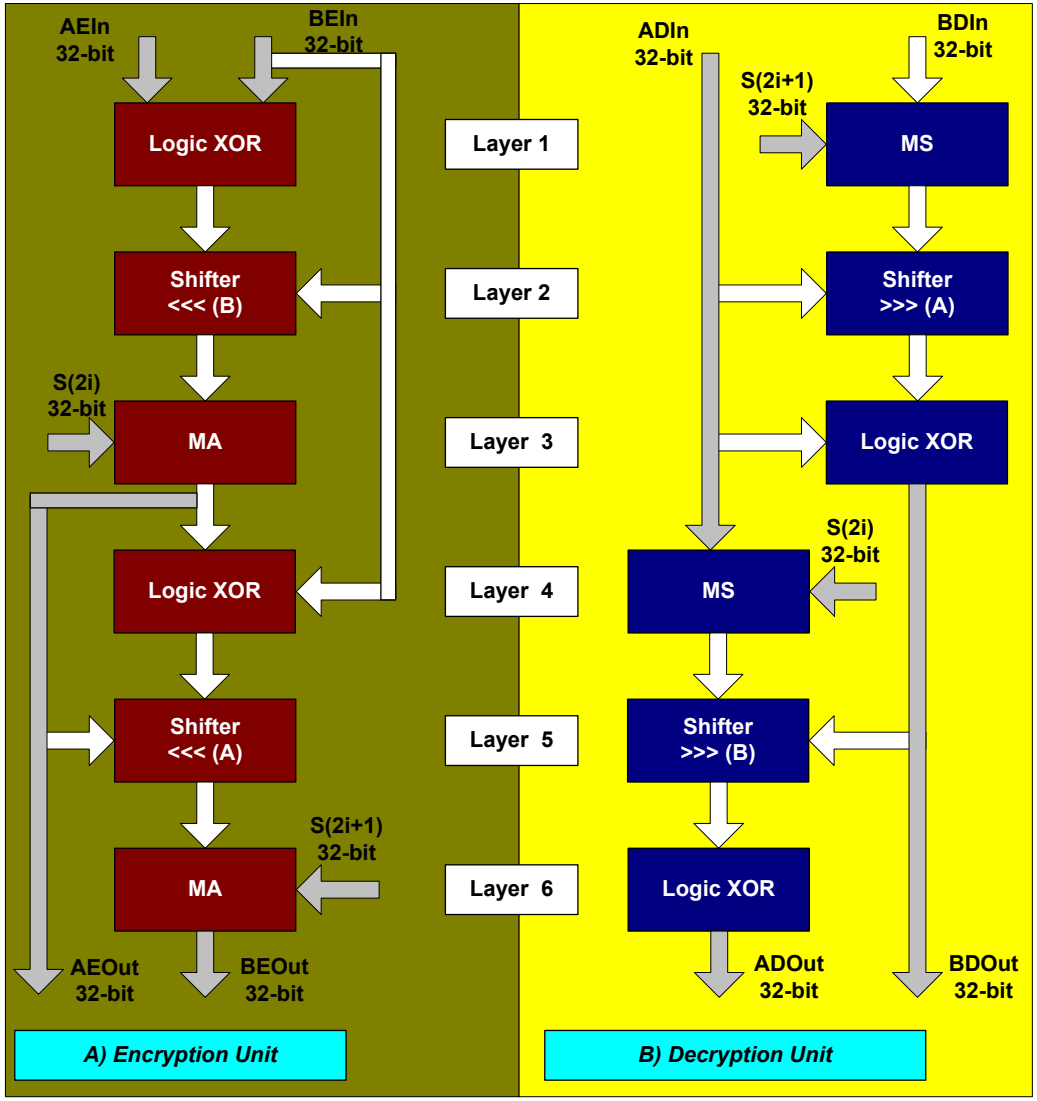


- NF1: Common Operation
- NFi, 2…4: Double Operation, SHA-1 and MD5

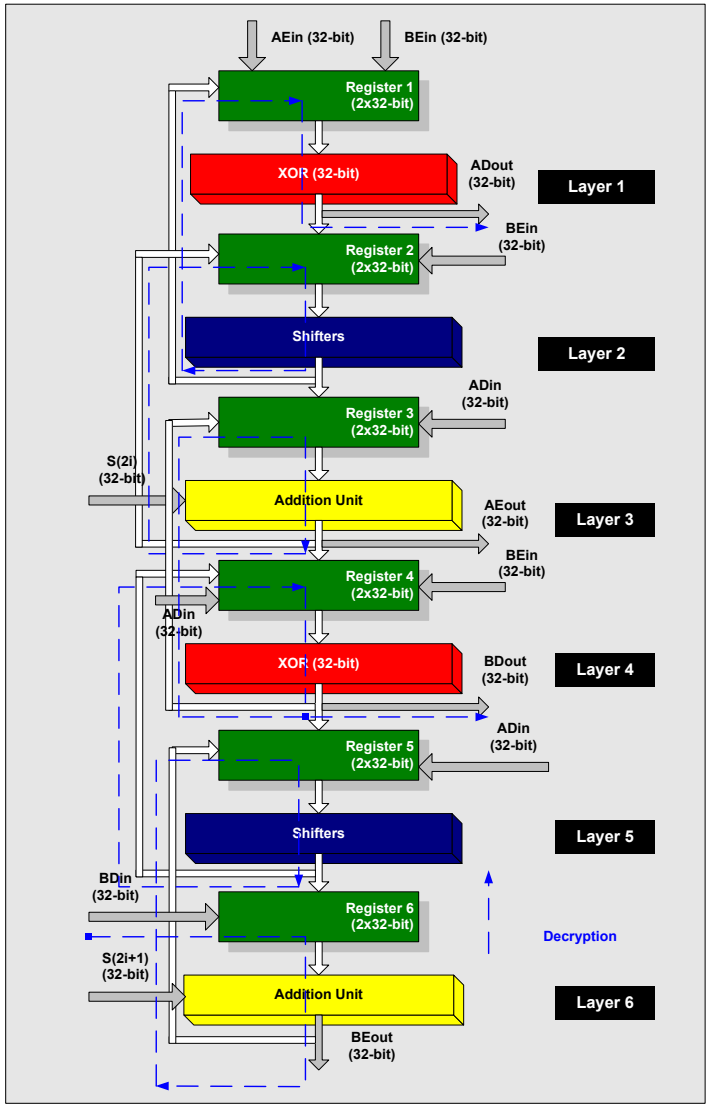
RC5 Block Cipher



Encryption/Decryption Processes

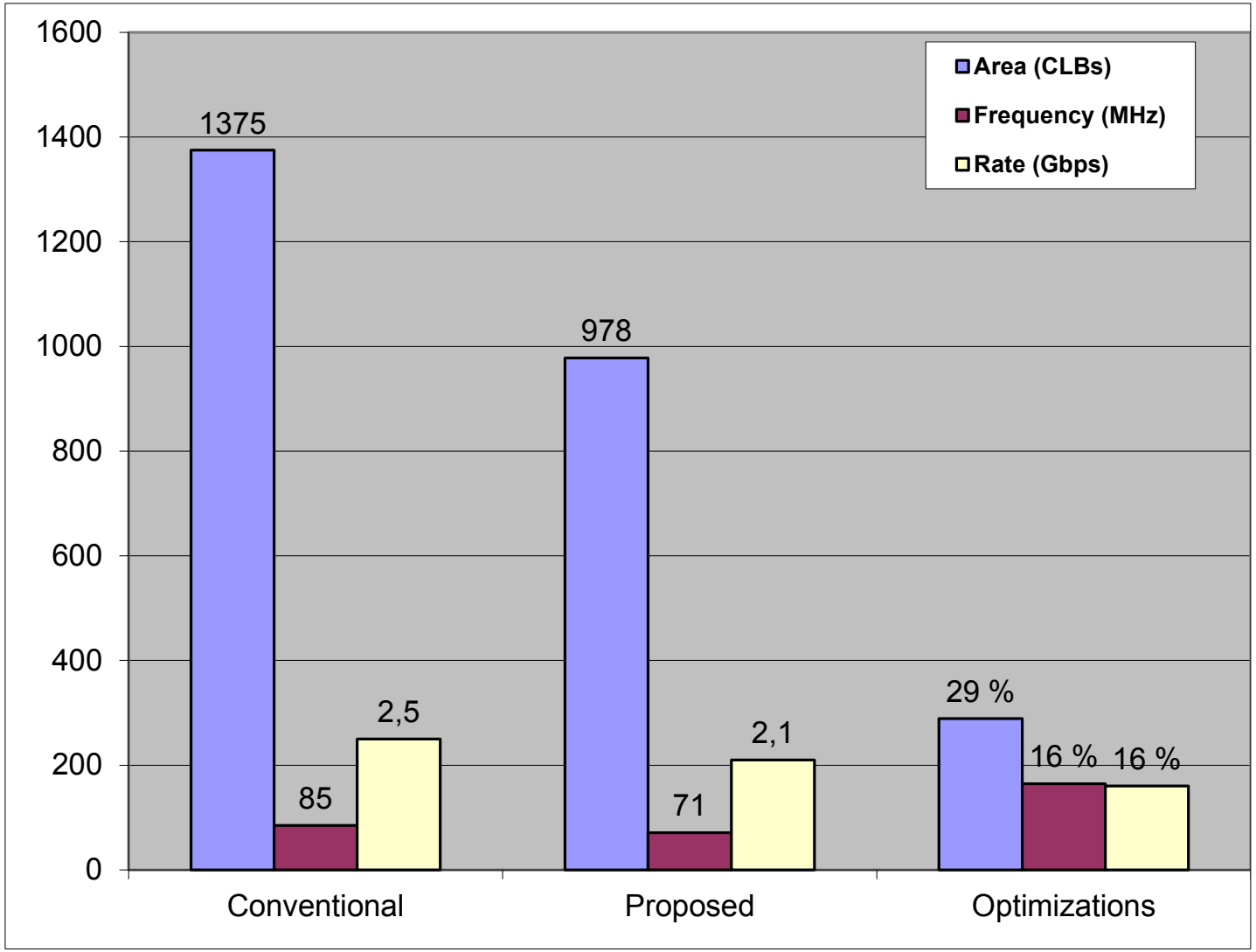


Proposed Design



- Encryption: Forward
- Decryption: Backwards
- Cascade of Registers
- Units: Forward and Backward Operation

Comparison Aspects



Proposed Architectures and Designs for

SECURE HASH FUNCTIONS FAMILY VERSION 2

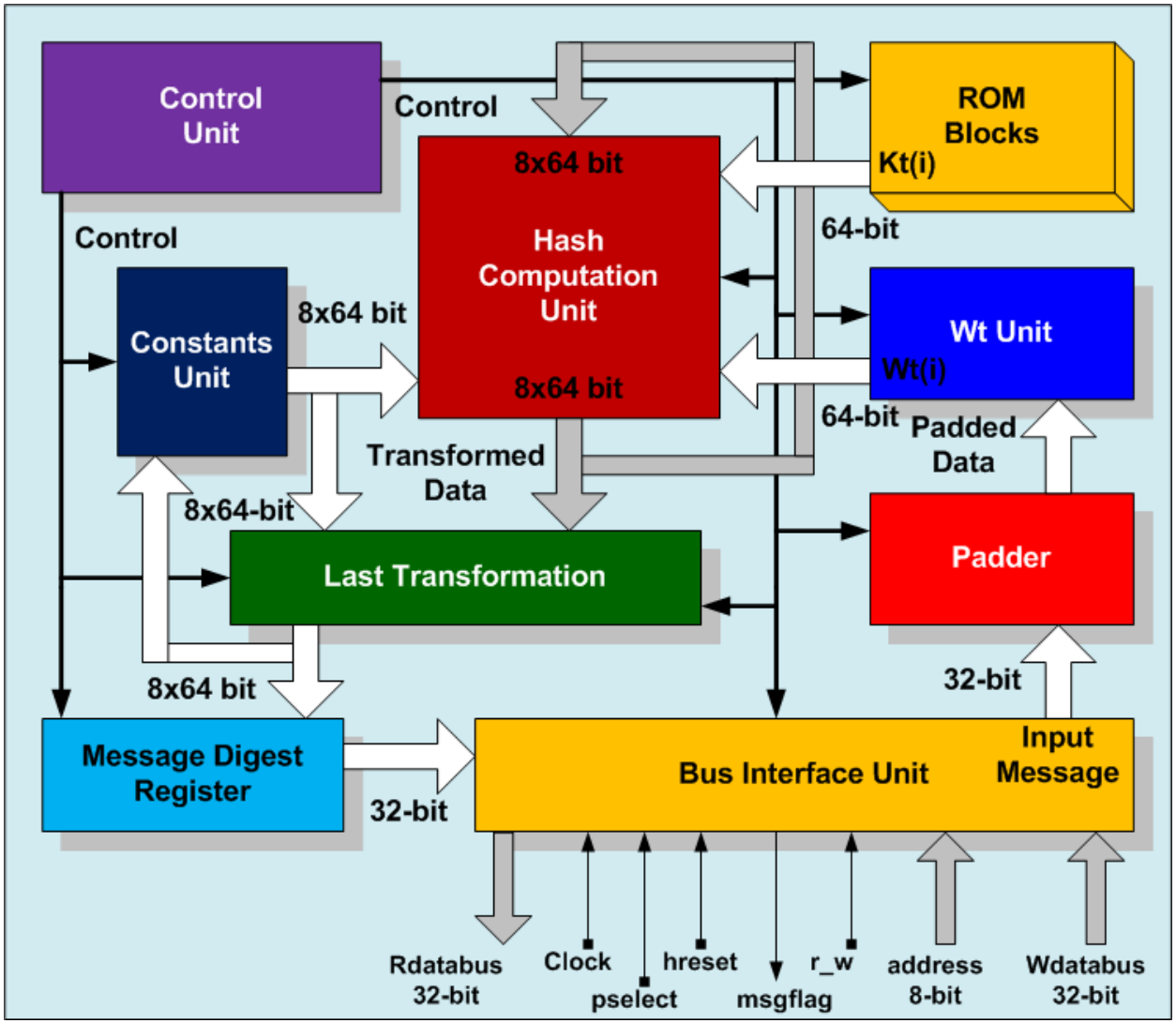
SHA-2 Hash Family

- Contains three Hash Functions:
  SHA-2(256), *SHA-2(384)*, and *SHA-2(512)*
- Hash Value: Output Value *256-* , *384-* & *512-* *bit vector*
- Operation with three phases of Data Transformation:
  - *Preprocessing (Padding)*
  - *Hash Computations (Data Transformations)*
  - *Last Modification*

Functions Specifications

TERMS (-bits)	SHA-2 (256)	SHA-2 (384)	SHA-2 (512)
Message Size	< 2^{64}-	< 2^{128}-	< 2^{128}-
Block Size Input	512 -	1024 -	1024 -
Word Size	32 -	64 -	64 -
Rounds	*64*	*80*	*80*
Hash Value Output	256 -	384 -	512 -

SHA-2 Family Proposed Architecture

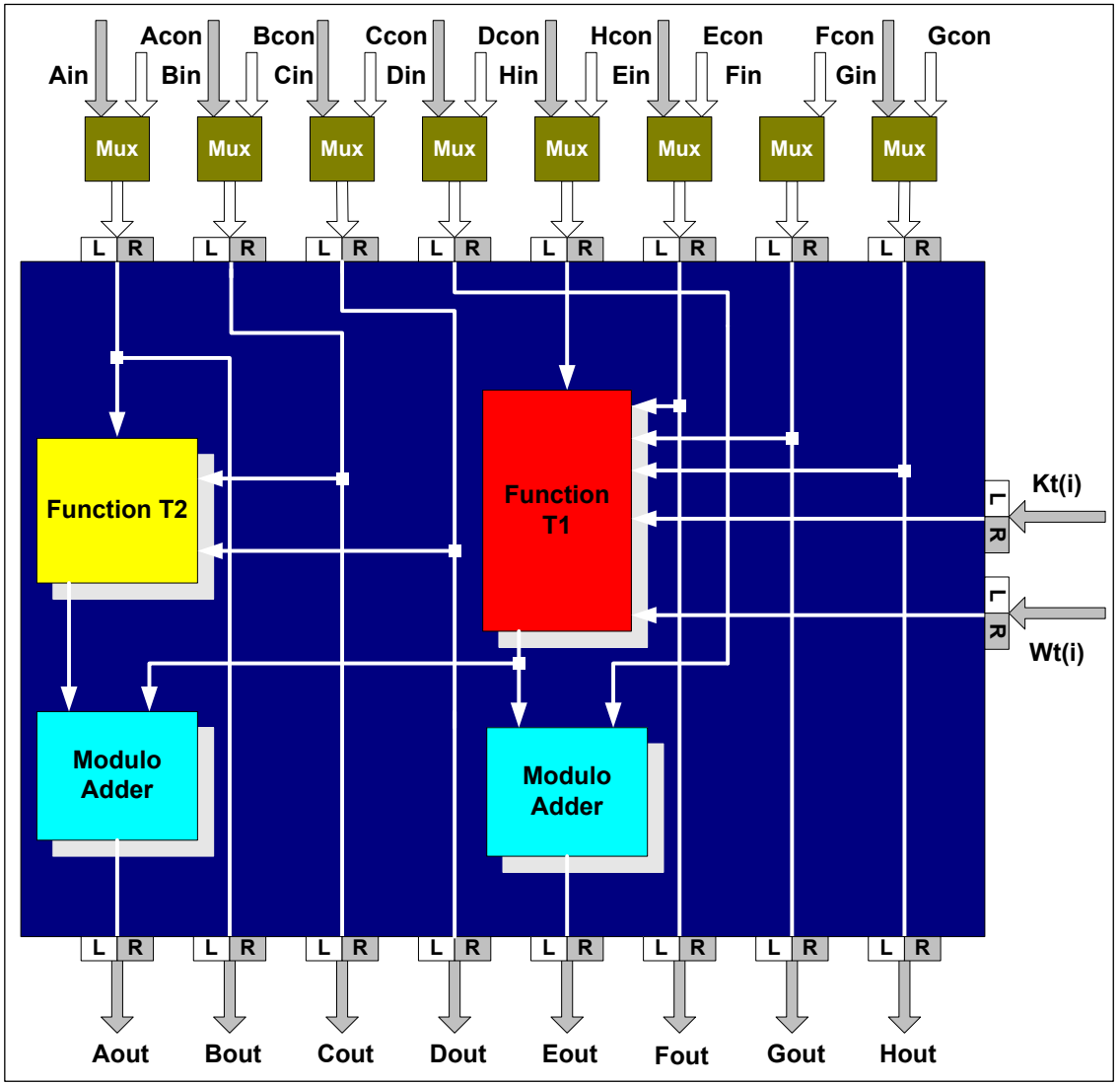


Data Transformation Unit

- Eight Basic IN/OUTs
- Variables in Length: In/Outs (L||R), (32 II 32)-bit :

  A) 32- B), C) 64-bit
- Rounds of Transformation:

  A) : 64 B), C) : 80
- Functions T1, T2,
- Modulo Adders Units



Generic Architecture of Function Ti

Shifters Units:

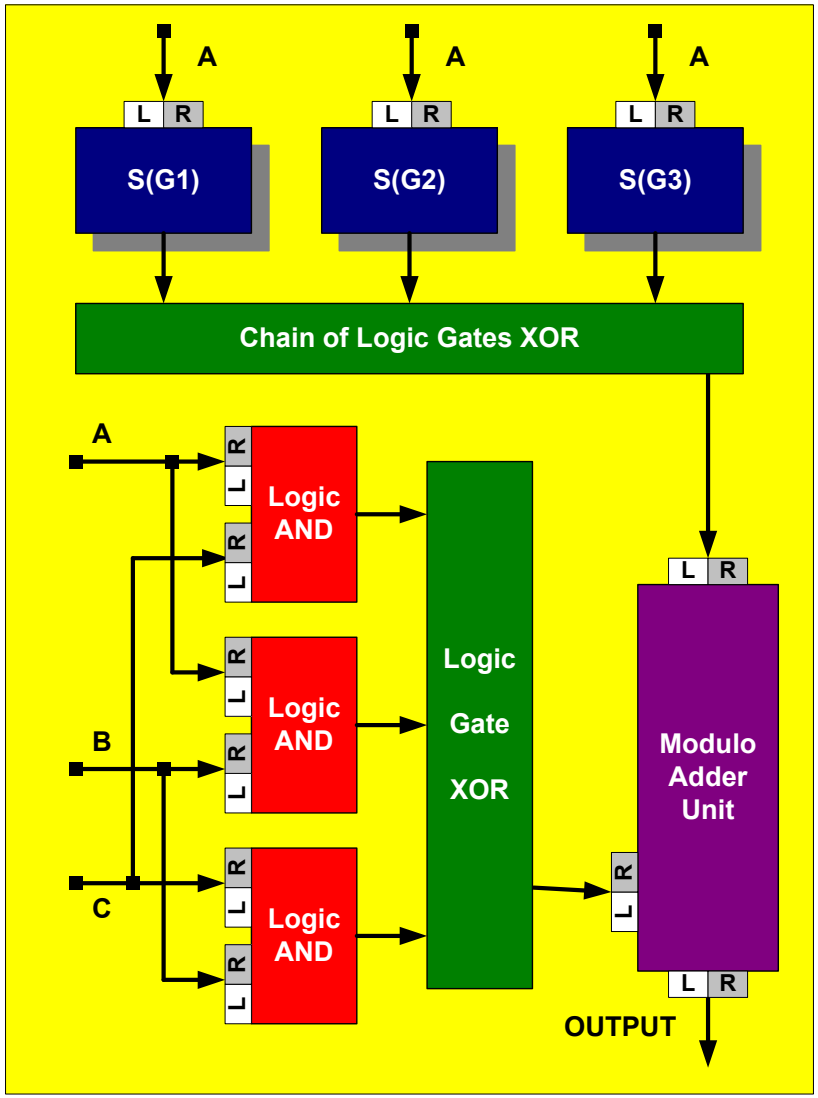
Shifting by a different constant value each time

Adders:

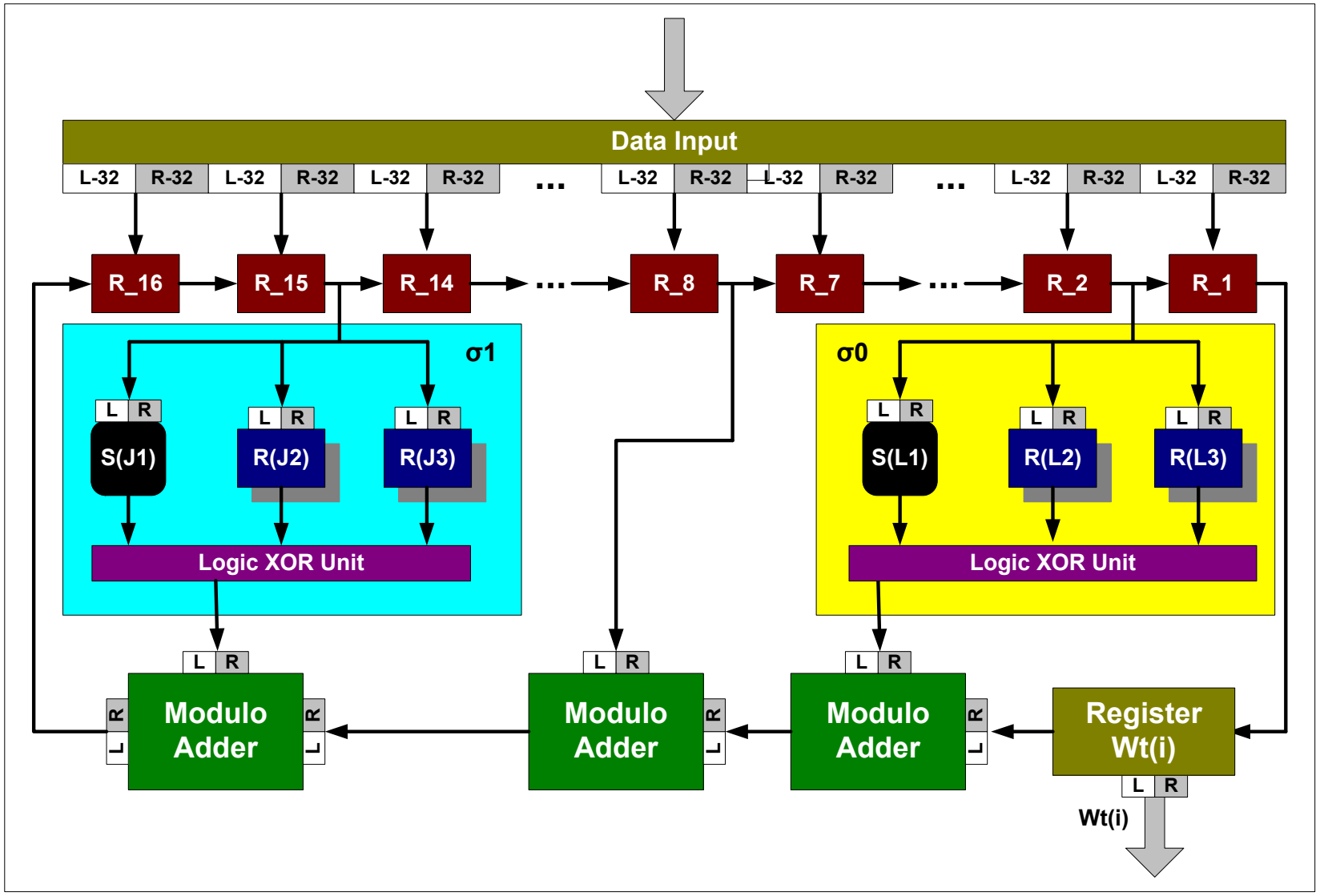
Modulo Adders 2^{32} - and 2^{64} - bit

Logic Gates:

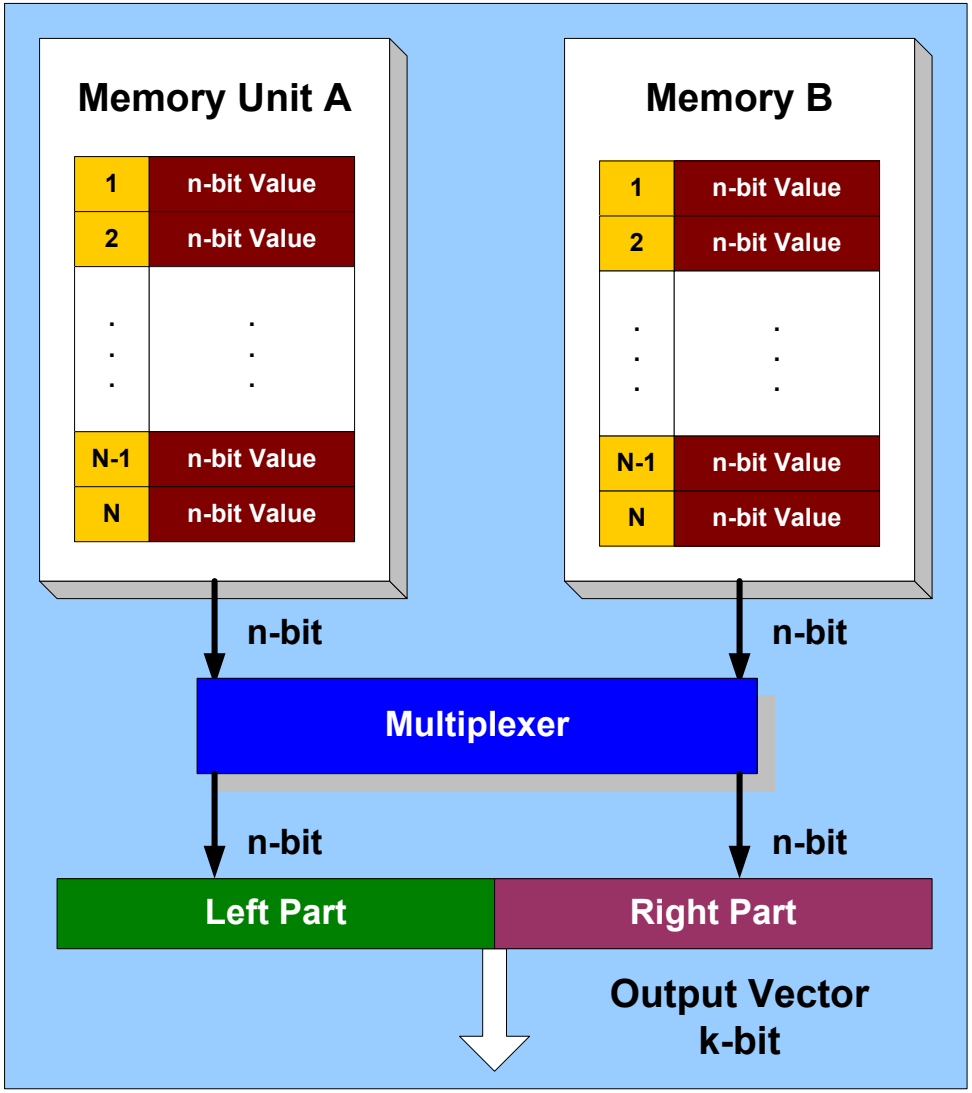
Cascade of Logic Gates AND & XOR



Data Transformation Units



Memory Approach RAM/ROM



Specifications:

- Design of Memory Use
  Based on rows RAM or ROM
- Memory Unit Size:
  N X n-bit
- Vector Creation
  (L || R) = (n-bit || n bit)

SECURE HASH FUNCTIONS FAMILY VERSION 3: SHA-3

Secure Hash Functions Standard V3

- 1995: Secure Hash Function Algorithm Version 1
  - *Limited security level to an 80-bit block cipher.*
- 2002: Secure Hash Function Algorithm Version 2
  - *Family of Three Hash Functions: SHA-256, SHA-384, and SHA-512 .*
  - *Longest message digest: 256-, 384-, and 512- bit.*
- 2010 – 12: Secure Hash Function Algorithm Version 3
  - *The competition received **64 proposed candidates**, while **51 of them progressed** to the next round.*
  - *SHA-3 Competition Current State: 2nd Candidate Conference, University of Santa Barbara, California, August 23-24, 2010: **14 second round candidates**.*

And the Winner Is … ?

- NIST selected **the finalists of SHA-3**, on December 9, 2010:
  1. BLAKE, 2. Grostl, 3. JH, 4. Keccak, 5. Skein
- *A **one-year public comment period*** was planned for these candidates.
- The Third **SHA-3 Candidate Conference** will be held at **Washington, DC on March 22-23, 2012**. The purpose of the conference is to discuss the SHA-3 finalist algorithms, and to solicit public feedback.
- The ***selection of the SHA-3 winner***, was announced on October ***2012: Keccak !***

Motivation

- **Performance in hardware** has been a key factor in the selection of the winner.
- Many hardware implementations of the SHA-3 finalists have been published: targeted to design **criteria of performance**.
- **Straight forward implementations**, of the functions using the specifications, described.
- Developed using **ASICs & FPGAs**.

This Work is About

- Designed for ***resource constrained environment***, on a wide variety of platforms: PDAs, smart phones, Satellite Communications etc.
- Cryptographic algorithms, may not be the main application but ***a great part of a huge complex system***.
- Many functions have to be included in addition to security, for ***System on Chip devices***.
- ***Area and allocated resources*** are the majority constraints.
- ***Multi Mode Operation***, for all the four functions of BLAKE family.
- Comparisons with straight forward and stand alone ones, in order to have a **fair and detailed comparison** regarding throughput, and allocated resources.

BLAKE Hash Functions Family

- One of the *4 third round finalists* and one of the most good proposed hash functions, since it meets all NIST criteria for SHA-3 standard:
  - *produced message digests of final size equal to 224-, 256-, 384-, and 512-, bit*
  - *similar parameter sizes to SHA-2*
  - *maximum message length of at least (2^64-1) - bit*
  - *streaming mode of one pass*

But what is called "BLAKE" ?

- A family of four alternative functions:
  - *BLAKE-28, BLAKE-32, BLAKE-48, & BLAKE-64*

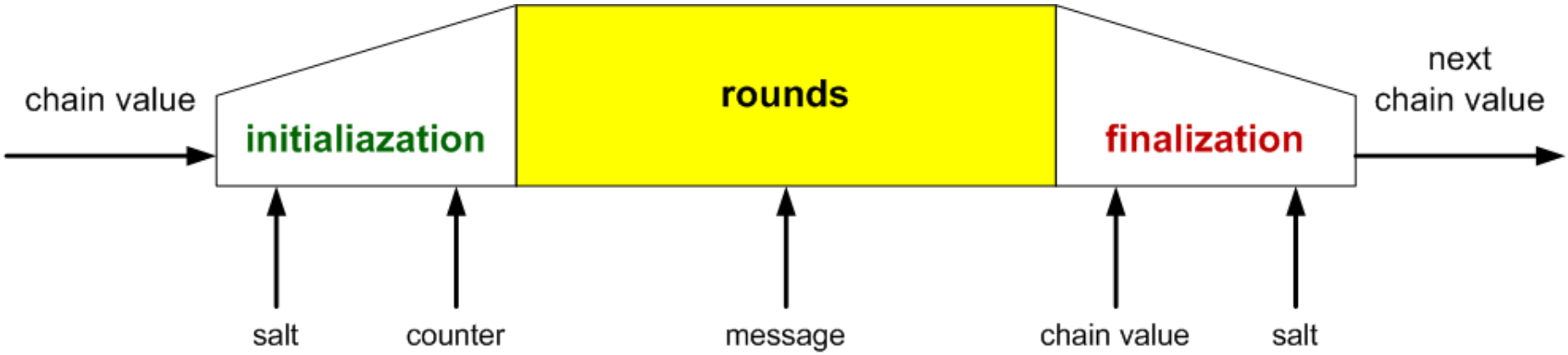
- These functions are basically different in:

1. *word length they use for data transformation*
2. *length of applied message block*
3. *produced message digest*
4. *used "salt" (fundamental used vector, randomly generated)*

BLAKE Family Specifications

Function	Word	Message	Block	Digest	Salt
BLAKE-28	*32-bit*	*< 2^64-bit*	*512-bit*	*224-bit*	*128-bit*
BLAKE-32	*32-bit*	*< 2^64-bit*	*512-bit*	*256-bit*	*128-bit*
BLAKE-48	*64-bit*	*< 2^128-bit*	*1024-bit*	*384-bit*	*256-bit*
BLAKE-64	*64-bit*	*< 2^128-bit*	*1024-bit*	*512-bit*	*256-bit*

BLAKE Compression Function



- Operation philosophy is inherited from HAIFA & LAKE.
  - **Initialization:** *chain value, salt, counter.*
  - **Rounds:** *message dependent, compress.*
  - **Compression:** *representation as a 4x4 matrix.*
  - **Finalization:** *chain value, salt.*

Design Philosophy

- Generic Architecture for the three BLAKE Hash Functions
- Similar Data Transformations but not (!!!) the same
- Parameterized Proposed System
- Selection of the desired operation in pre-programmable way of use.
- Future-Current Work : Multi-operation architecture.

Proposed Generic Architecture

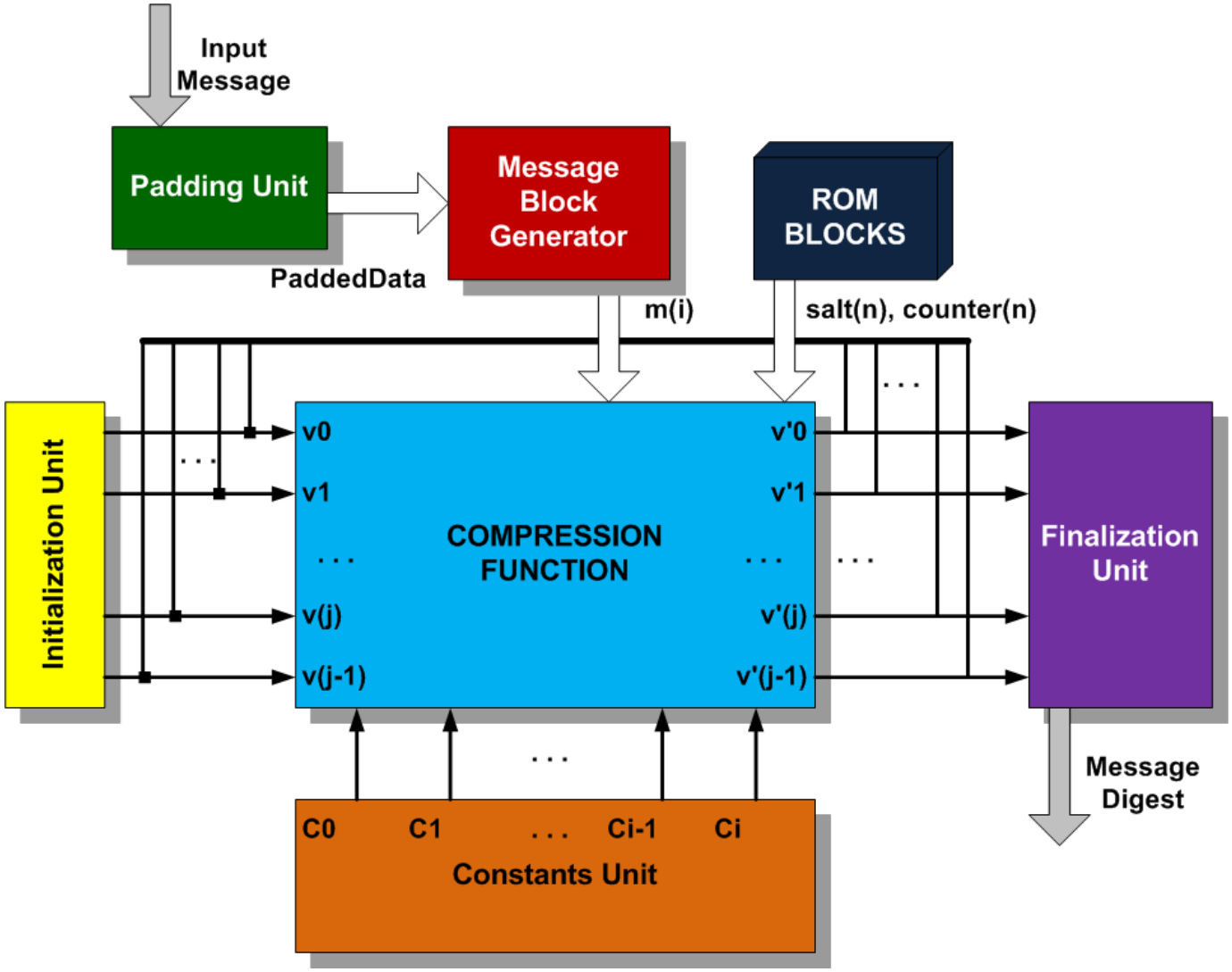
 # Rounds:

10: BLAKE-28

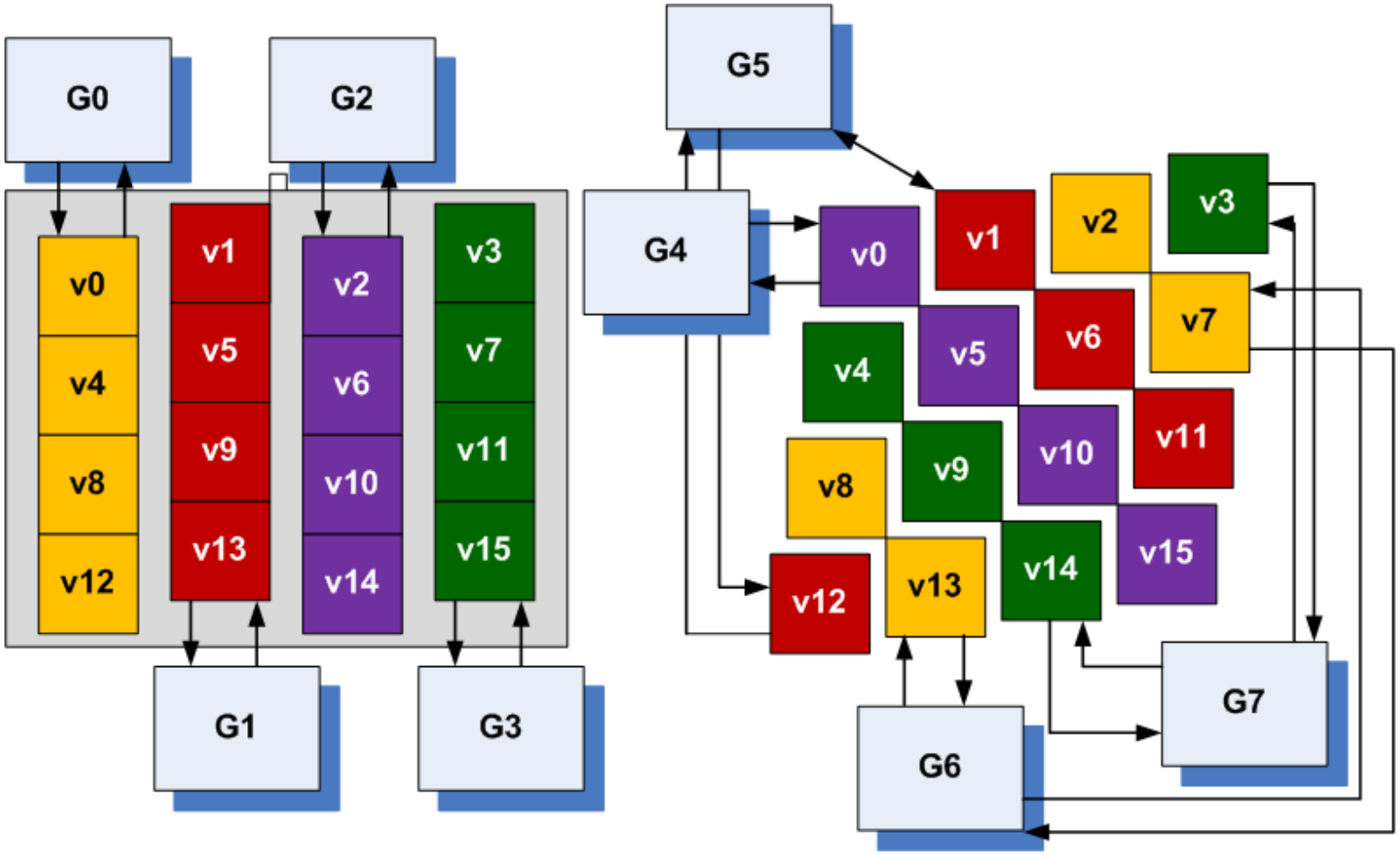
10: BLAKE-32

14: BLAKE-48

14: BLAKE-64

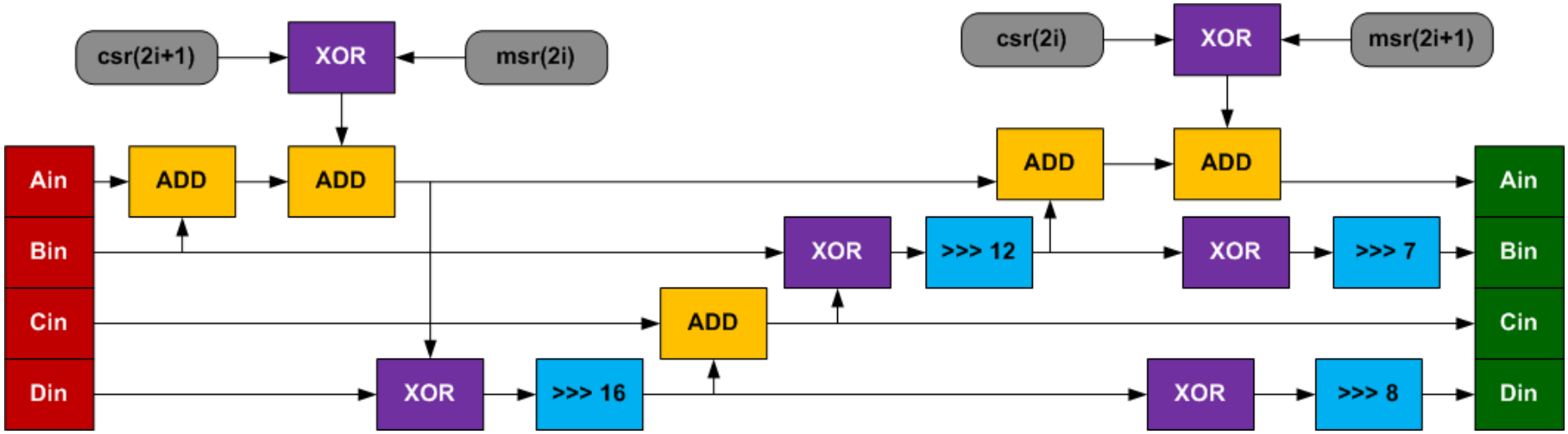


G(i) Functions Architecture, i=0,…,7



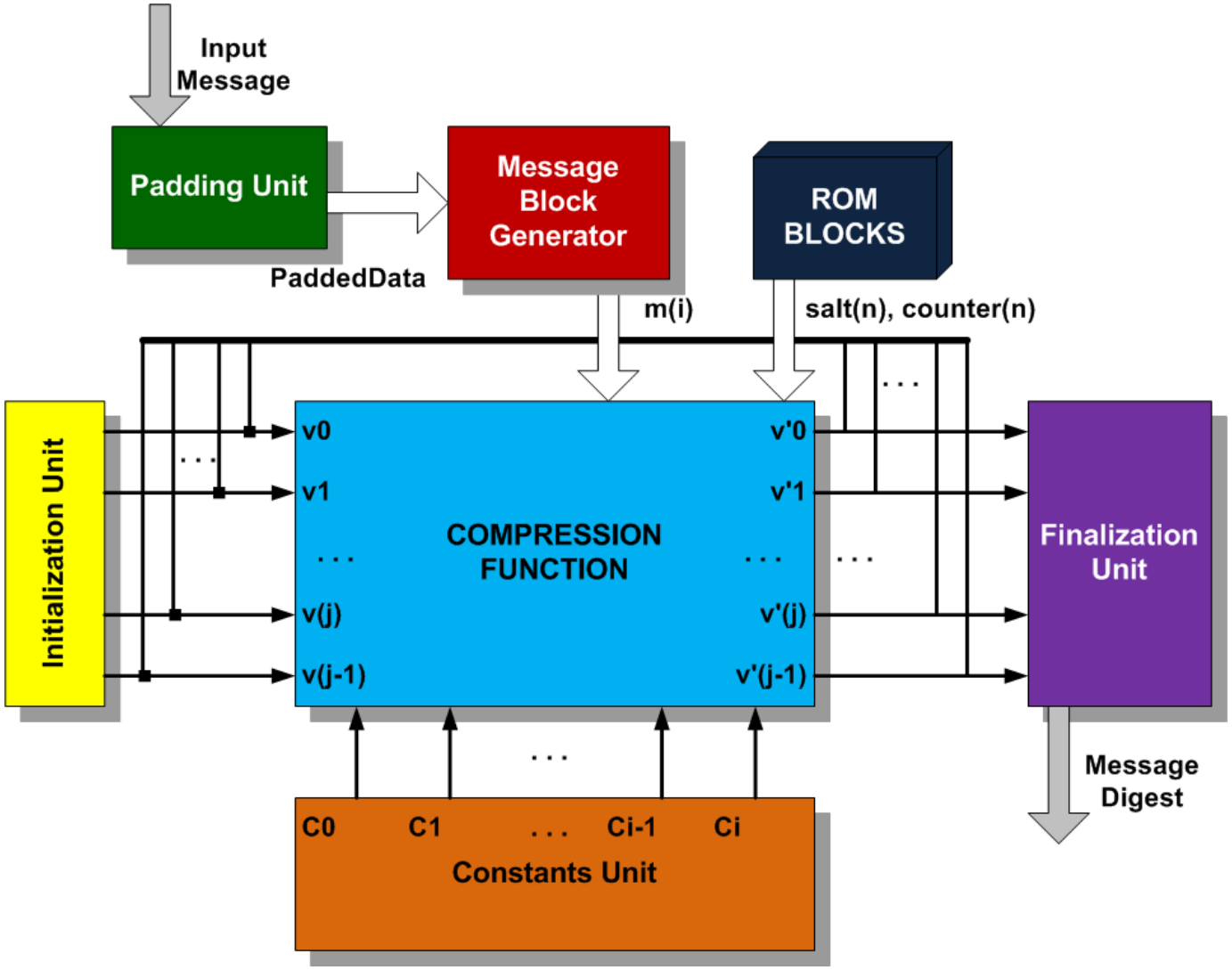
Type of Transformation: a) Column, b) Diagonal Step

G(i) Functions Architecture



- Chain Blocks of : **XOR, Modulo ADD,** & **Rotation**
- Message blocks: Msr(i), where r=round, i : function G(i)
- Counter data: Csr(i), where i : function G(i)

Multi-Mode System



 # Rounds:

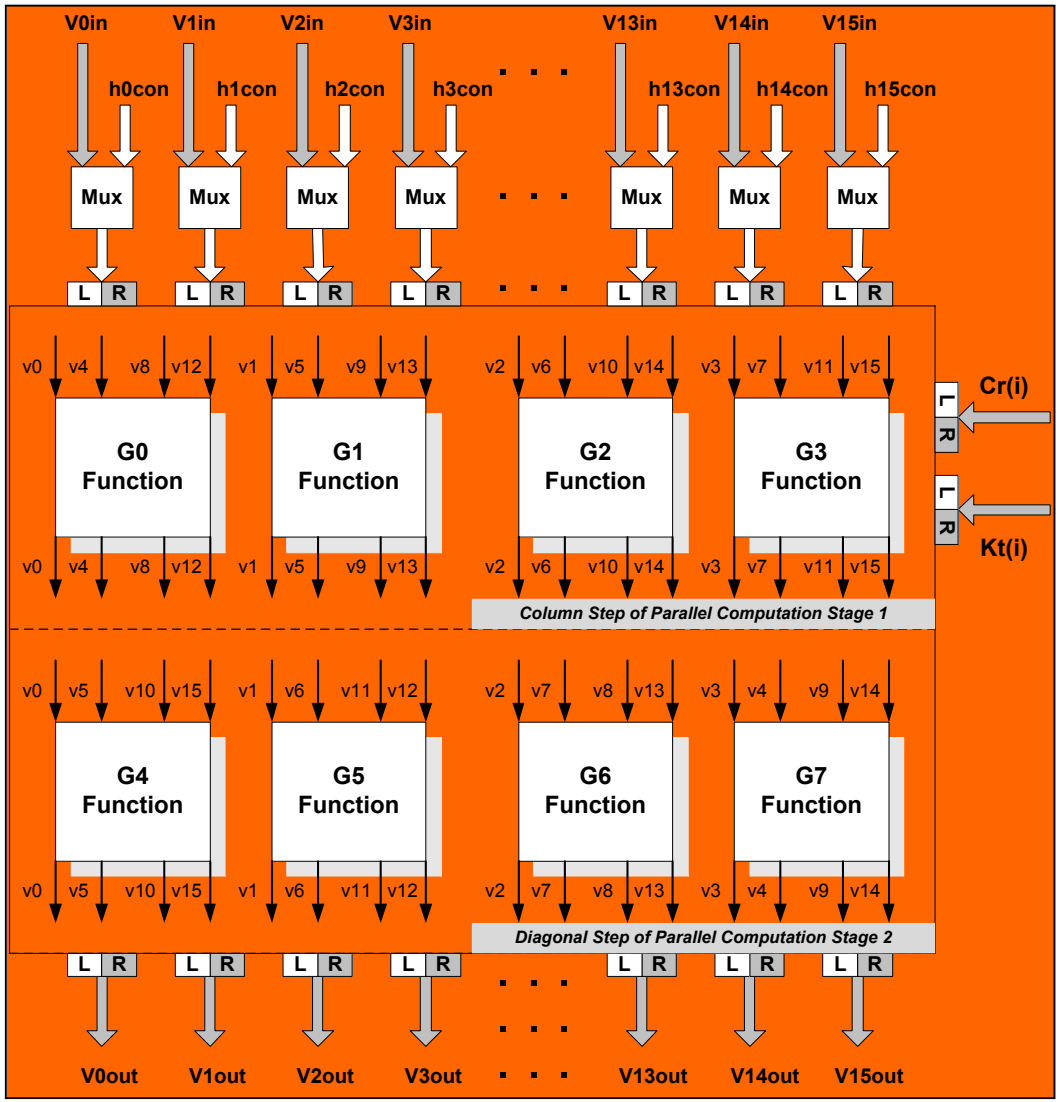
10: BLAKE-28

10: BLAKE-32

14: BLAKE-48

14: BLAKE-64

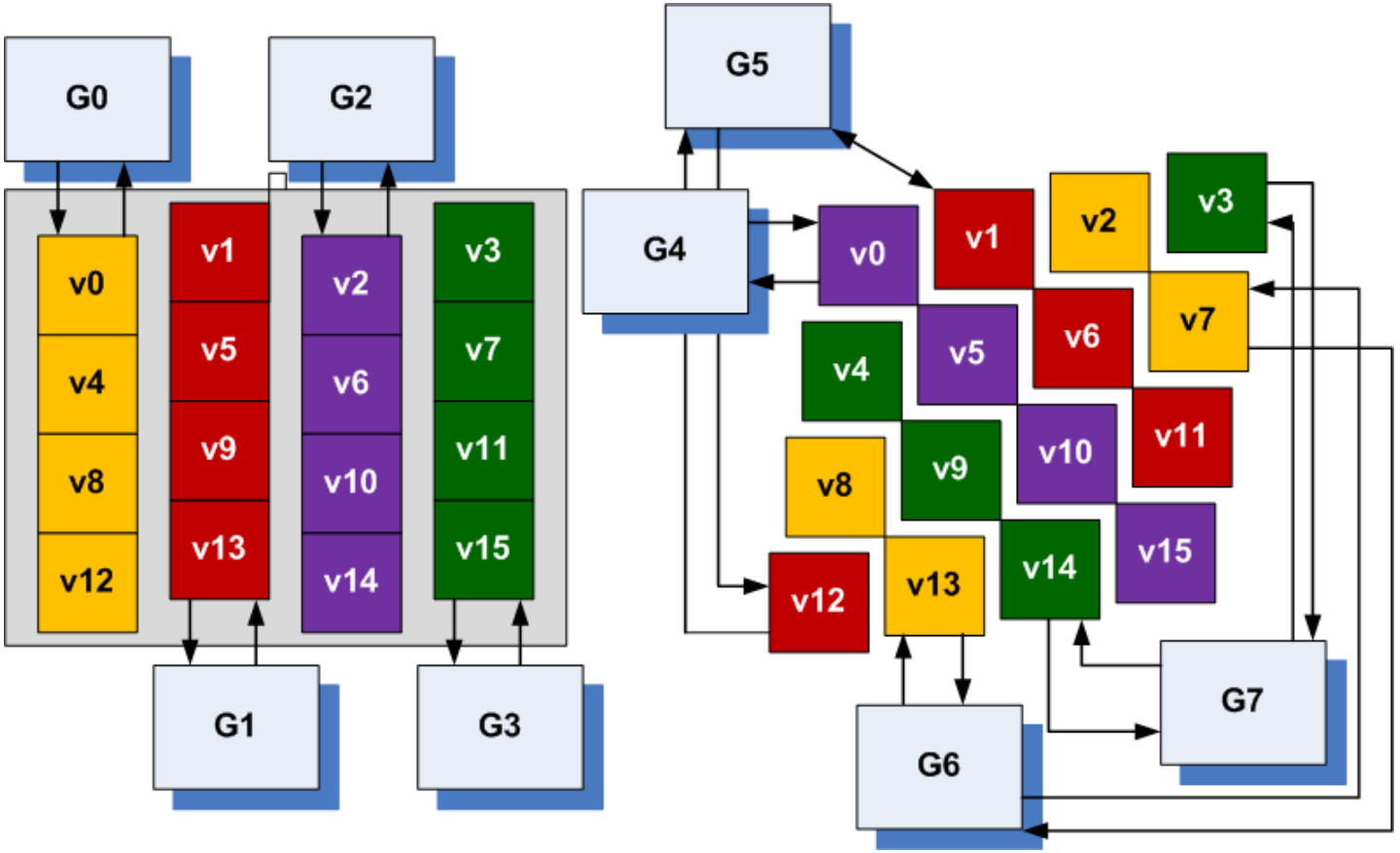
Compression Function



Compression Function:

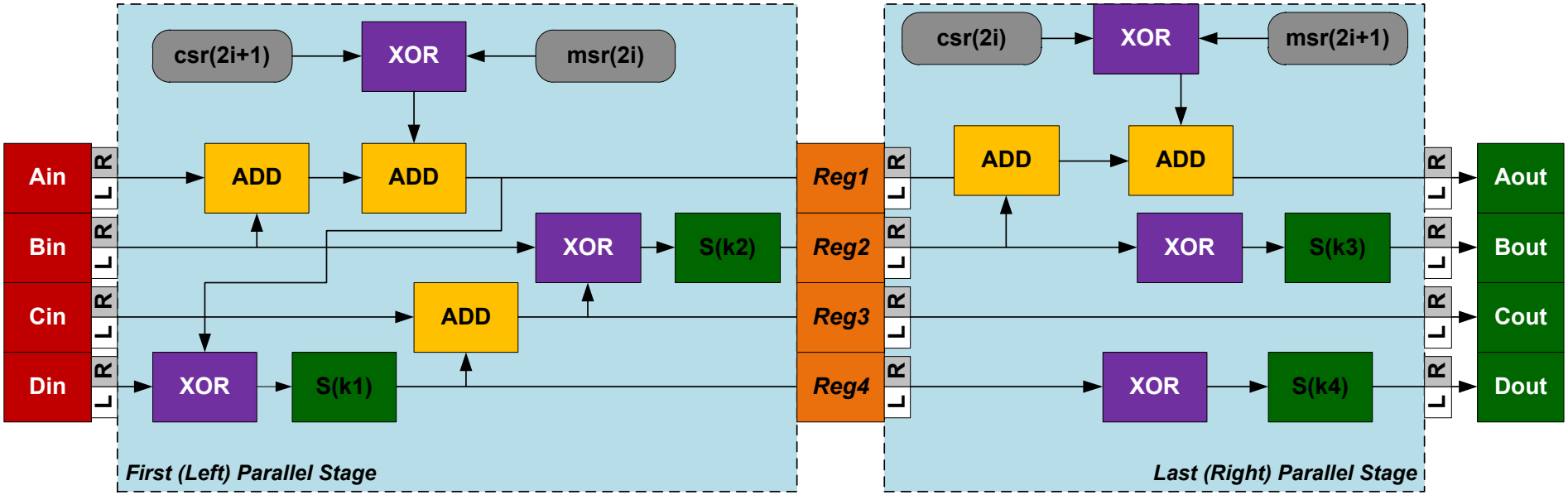
- G(i) Functions, i=0,…,7
- 8 Initial Values Set
- Left & Right Part of Data Inputs: L & R
- 32-bit || 32-bit , (64-bit)
- Idle or Active || Active
- Idle may means stack at Hex OO…OO

G(i=1, 7) Functions Architectures



Type of Transformation: a) Column, b) Diagonal Step

G(i) Functions: Multi-Mode



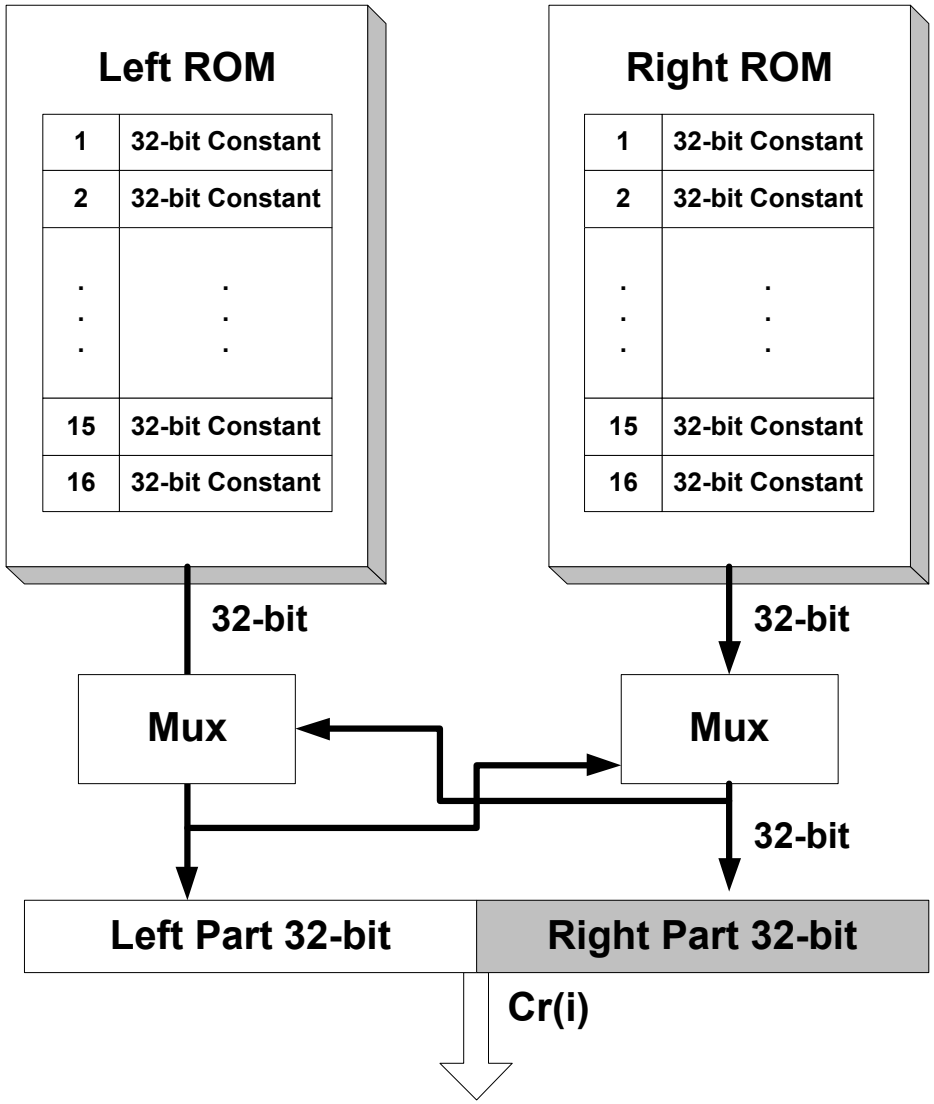
- Four functions of BLAKE hash family
- Two Parallel Stages
- "Left" and "Right" parts, active of idle
- Defined number of data rotations S(ki)
- District constants sets, Number of Rounds

Constants Unit

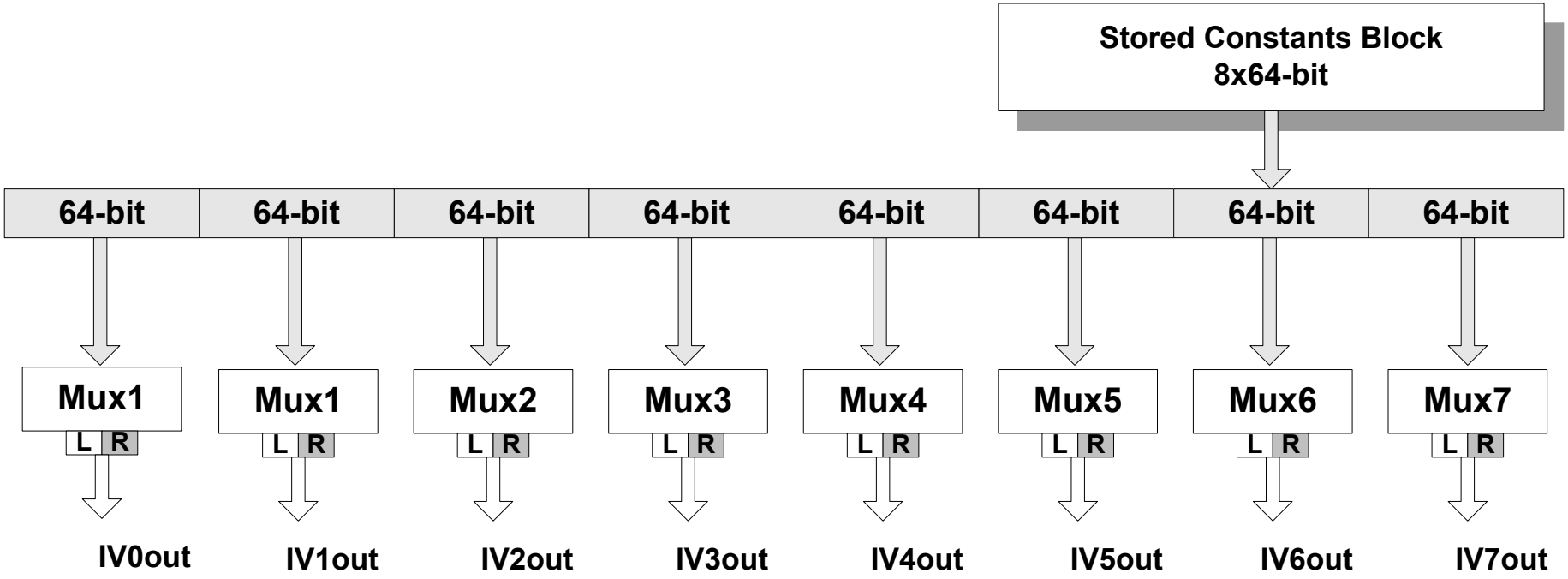
Used ROM Blocks:

- **BLAKE-28: (768-bit in total)**
  - Eight Initial Values of SHA-2(224) : 8 x 32-bit , Sixteen Constants: 16 x 32-bit
- **BLAKE-32: (768-bit in total)**
  - Eight Initial Values of SHA-2(256) : 8 x 32-bit, Sixteen Constants: 16 x 32-bit
- **BLAKE-48: (1536-bit in total)**
  - Eight Initial Values of SHA-2(384): 8 x 64-bit, Sixteen Constants: 16 x 64-bit
- **BLAKE-64: (1536-bit in total)**
  - Eight Initial Values of SHA-2(512) : 8 x 64-bit, Sixteen Constants: 16 x 64-bit

Constants Unit Memory Block



Initial Values Generator



BLAKE-28: 8 Initial Values of SHA-2(224) : 8 x 32-bit , 16 Constants: 16 x 32-bit

BLAKE-32: 8 Initial Values of SHA-2(256) : 8 x 32-bit, 16 Constants: 16 x 32-bit

BLAKE-48: 8 Initial Values of SHA-2(384): 8 x 64-bit, 16 Constants: 16 x 64-bit

BLAKE-64: 8 Initial Values of SHA-2(512) : 8 x 64-bit, 16 Constants: 16 x 64-bit

Finalization Unit Architecture

- Operates on the last step
- It has been implemented as a chain of XOR blocks, between the initial chain value h, and the salt s.
- For example:

$$h'0 = h0 \text{ XOR } s0 \text{ XOR } v0 \text{ XOR } v8$$

$$\ldots$$

$$h'7 = h7 \text{ XOR } s3 \text{ XOR } v7 \text{ XOR } v15$$

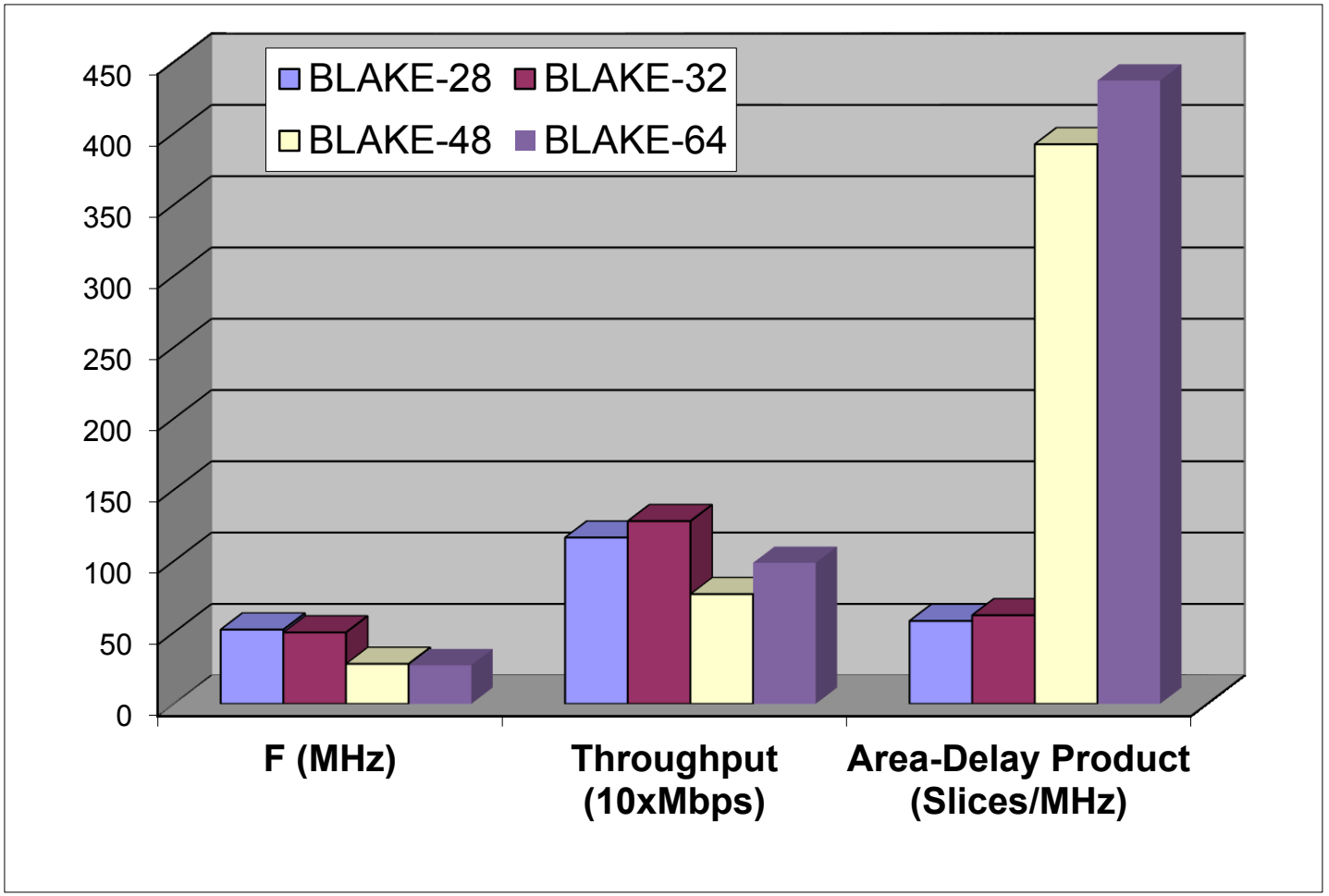
- 24 XOR Gates of 64-bit are used in total

FPGA BLAKE Implementations

	F (MHz)	Throughput (10xMbps)	Area (Slices)
BLAKE-28	*52*	*116,48*	*3017*
BLAKE-32	*50*	*128,00*	*3101*
BLAKE-48	*28*	*76,80*	*10986*
BLAKE-64	*27*	*98,74*	*11800*

Throughput = (# Bits of Message Block X Frequency) / (# Clock Cycles of Message Block)

BLAKE Family Comparison Graph



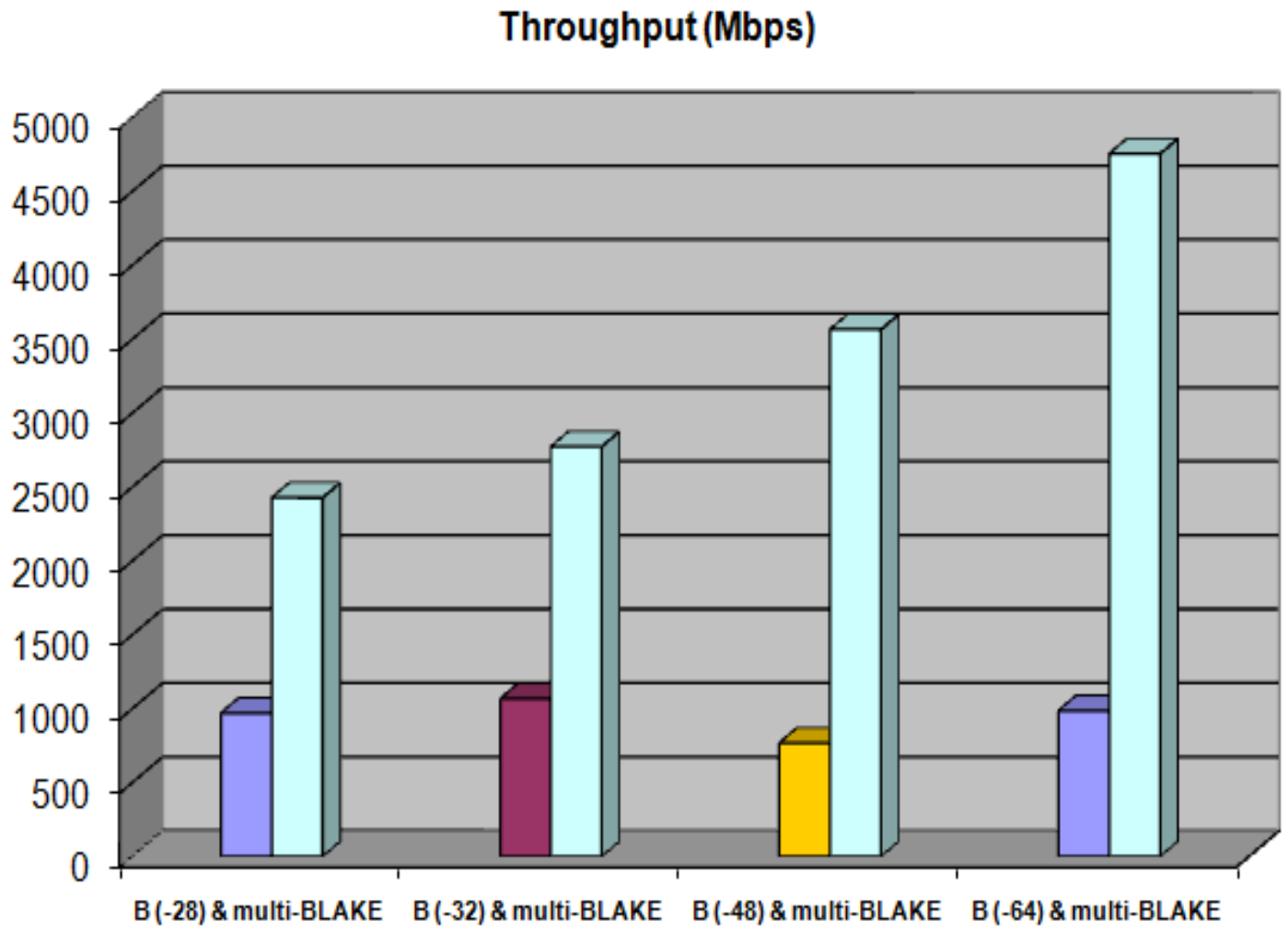
Comparisons with Other Works

IMPLEMENTATIONS	AREA	F (MHz)	RATE (Mbps)
SHA-2	*1060, 1966, 2237*	*83, 74, 74*	*326, 350, 480*
CUBEHASH	*1178*	*167*	*0,160 Gbps*
CROSTL	*5878*	*128*	*3,28 Gbps*
LANE	*3442*	*133*	*1,38 Gbps*
SHABAL	*2307*	*222*	*1,33 Gbps*
SPECTRAL HASH	*951*	*125*	*0,18*
Str. Forw. BLAKE-28	*3017*	*52*	*116,48*
Str. Forw. BLAKE-32	*3101*	*50*	*128,00*
Str. Forw. BLAKE-48	*10986*	*28*	*76,80*
Str. Forw. BLAKE-64	*11800*	*27*	*98,74*

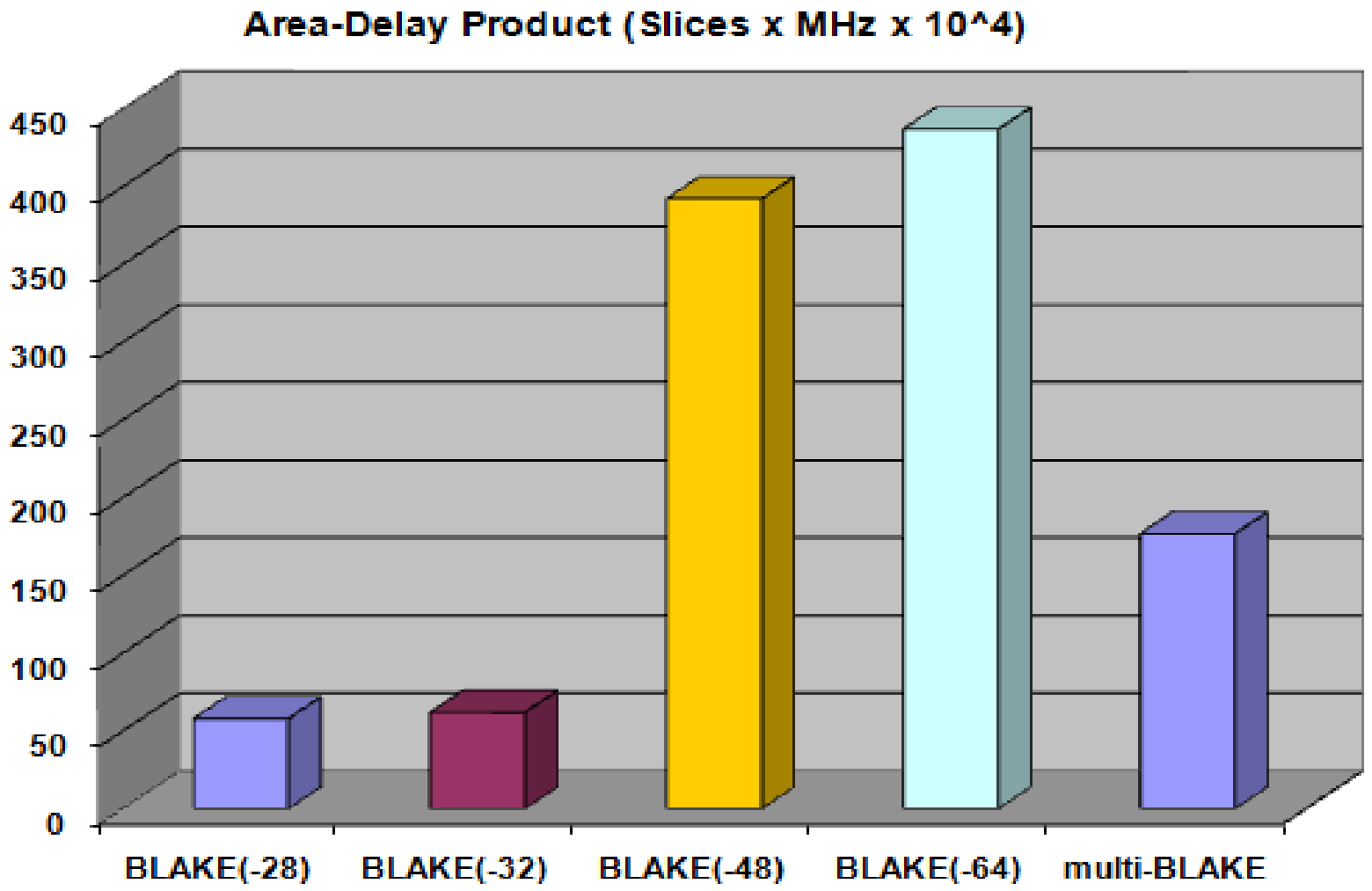
Multi-Mode System

	F (MHz)	ROM Blocks Used	Area (Slices)
BLAKE-28	*52*	-	*3017*
BLAKE-32	*50*	-	*3101*
BLAKE-48	*28*	-	*10986*
BLAKE-64	*27*	-	*11800*
Multi Mode System	*65*	*16x-64-bit ROM*	*11500*

Throughput of Multi-Mode System



Area-Delay Product Comparison



PART III

IMPLEMENTATION PLATFORMS

Alternative Directions of Implementation

SOFTWARE VS HARDWARE

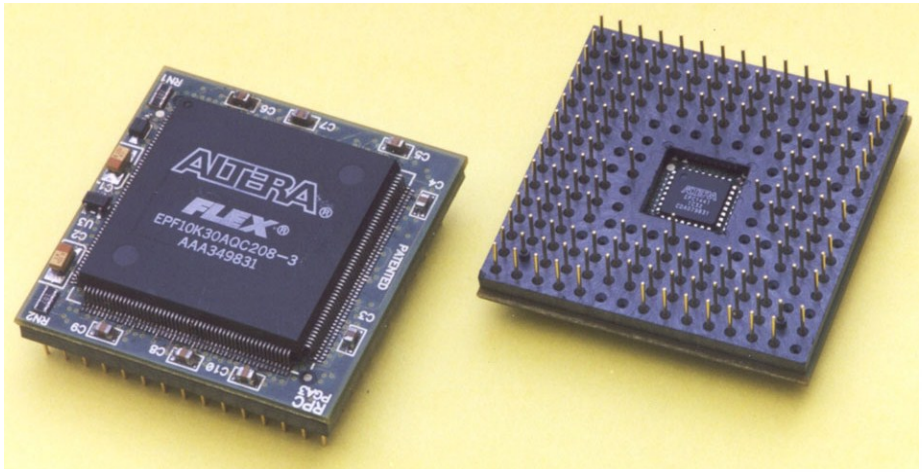
Software Approaches

- Software implementation usually **satisfies user requirements.**
- **Widely used** especially the previous years.
- The cryptography algorithms **demands high processing capabilities**.
- Software implementations are **slow.**
- A software implementation is **easy to be broken or attacked**.
- The implementation lead to a **large code size**.
- In critical applications and processing constrained devices, **the software is not preferable**.

Hardware Integrations

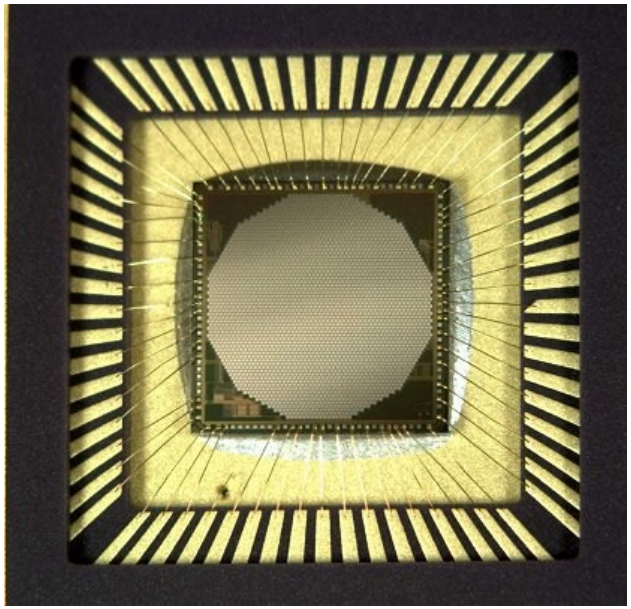
- Hardware implementation **fruitful requirements** in most of the cases.
- **Not so widely used** but they have rapid growth the last years.
- The satisfy the cryptography algorithms demands for **high processing capabilities.**
- Hardware integrations achieve **high speed performance.**
- A hardware implementation is **difficult to be broken or attacked**.
- **Time consumed**.
- Not available for all users.
- **Cost** a lot.

Hardware Integration Platforms



Field Programmable Gate Arrays (FPGAs)

Application Specific Integrated Circuits (ASICs)



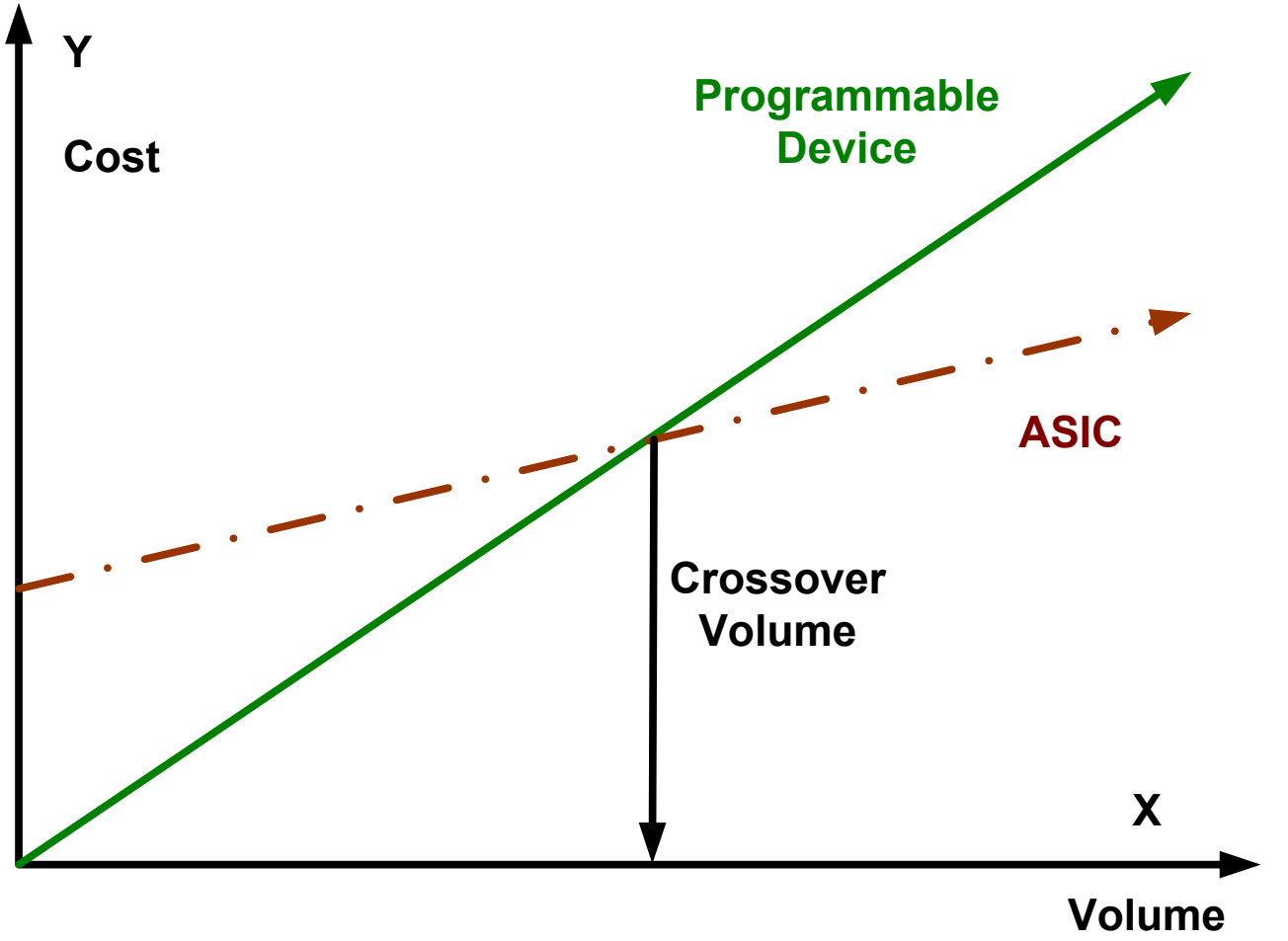
FPGAs & ASICs Comparison

CHARACTERISTICS	FPGAs	ASICs
Performance	*Low*	*High*
Power Consumption	*Very High*	*Low*
Logic Integration	*Low*	*High*
Tools Cost	*Low*	*Low*
Test Development Complexity	*Very Low*	*Very Low*
Density	*Very Low*	*High*
Design Effort	*Low-Medium*	*Low-Medium*
Time Consumed	*Low*	*High*

Software Vs Hardware

CHARACTERISTICS	SOFTWARE	HARDWARE	
		FPGAs	*ASICs*
Performance	*Low*	*Low*	*High*
Power Consumption	*Depends*	*Very High*	*Low*
Logic Integration	*Low*	*Low*	*High*
Tool Cost	*Low*	*Low*	*Low*
Test Development Complexity	*Very Low*	*Very Low*	*High*
Density	*High*	*Very Low*	*High*
Design Efforts	*Low-Medium*	*Low-Medium*	*High*
Time Consumed	*Short*	*Short*	*High*
Size	*Small-Medium*	*Small*	*Large*
Memory	*Fine*	*Fine*	*Fine*
Flexibility	*High*	*High*	*-*
Time to Market	*Short*	*Short*	*High*
Run Time Configuration	*-*	*High*	*-*

Cost of the Device



- **Field Programmable Gate Arrays:** low cost for minimized systems
- **ASICs:** low cost only huge systems, with mass production line

Working With

FIELD PROGRAMMABLE GATE ARRAYS (FPGAS)

FPGA Devices

- But what is an FPGA:
  - ✓ an **integrated circuit**…
  - ✓ …which belongs to the family of **PLDs**.
  - ➢ contains **thousands of building blocks**…
  - ➢ …called Configurable Logic Blocks (**CLBs**).
    - CLBs could be **reconfigured**…
    - …**functionality**, digital **circuit**, almost any kind of **operation**…

Main Advantages

- Reconfigurability:
  - different purposes
  - alternative stages of computing
  - reprogrammed: a) before run, b) on run-time

- Available to the market:
  - XILINX, http://www.xilinx.com/
  - ALTERA, http://www.altera.com/
  - …both of them share over 70% of the market !
  - LATTICE, ACTEL, QUICK LOGIC, ATMEL, ACHRONIX.

Applications of FPGAs

Almost any kind of algorithm, or circuit can be integrated using FPGAs:

- Embedded Systems,
- Network Processors,
- Real Time Systems,
- Digital Signal Processing,
- Multimedia, Computer Graphics,
- Robotics,
- …and Security Systems !!!

Basic Stages of Integration

- **System Development:**
  - *VHDL (VHSIC Hardware Description Language)*
- **Code Simulation :**
  - *use of data for operation scenarios: test vectors*
- **Code Synthesis:**
  - *device of certain technology (FPGA)*
- **Circuit Simulation:**
  - *test of system operation*
- **Results- Optimizations:**
  - *system, components*
  - *throughput, area, allocated resources, power consumption*

VHDL Code: Logic Gate AND 64x64

```
--------------------------------------------------------------------------------
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
USE IEEE.std_logic_arith.all;

ENTITY AND64x64 IS
Port (A, B  :IN     std_logic_vector(63 downto 0);
      C  :OUT    std_logic_vector(63 downto 0));
END AND64x64 ;

ARCHITECTURE Structural of AND64x64 IS
BEGIN
BlockGen: FOR i IN 0 to 63 GENERATE

                C(i) <=  A(i)  AND  B(i);

      END GENERATE;
 END Structural ;
--------------------------------------------------------------------------------
```

Feedback-Loop Designs

Performance:

- Not very good values of throughput.
- The value of operating frequency is defined by the data transformation round unit.
- Time critical component is the data transformation unit.
- No delays for data communications.

Area Resources:

- Required Area Minimization.
- Minimized of the System Resources.

Power Consumption:

- Less Power Consumption.
- Reduced of the Critical Power Consumption Components.

Processor-Loop Designs

Performance:

- Not very good values of throughput.
- The value of operating frequency is defined by the data transformation round unit.
- Time critical component is the data transformation unit.
- Delays maybe be caused from data communications.

Area Resources:

- Minimization of the system resources.

Power Consumption:

- Good power consumption.
- One or critical components for power consumption.

Reconfigurable Computing Designs

"Reconfigurable computing, in the abstract sense, refers to any information processing system in which blocks of hardware can be reorganized or repurposed to adapt to changing data flows or hardware."
-Ron Wilson, EE Times

"A reconfigurable computer is a device which computes by using post fabrication spatial connections of compute elements."
-Andre DeHon

"Reconfigurable devices contain an array of computational elements whose functionality is determined through multiple programmable configuration bits."
-Katherine Compton and Scott Hauck

"General purpose custom hardware."
- Seth Copen Goldstein

Reconfigurable Computing Design

Performance:

- Good values of throughput.
- The value of operating frequency is defined by the time critical transformation round unit.
- No delays for data communications.

Area Resources:

- Required Area Minimization.
- Minimized of the System Resources.

Power Consumption:

- Almost Good power consumption.

PART IV

BIBLIOGRAPHY

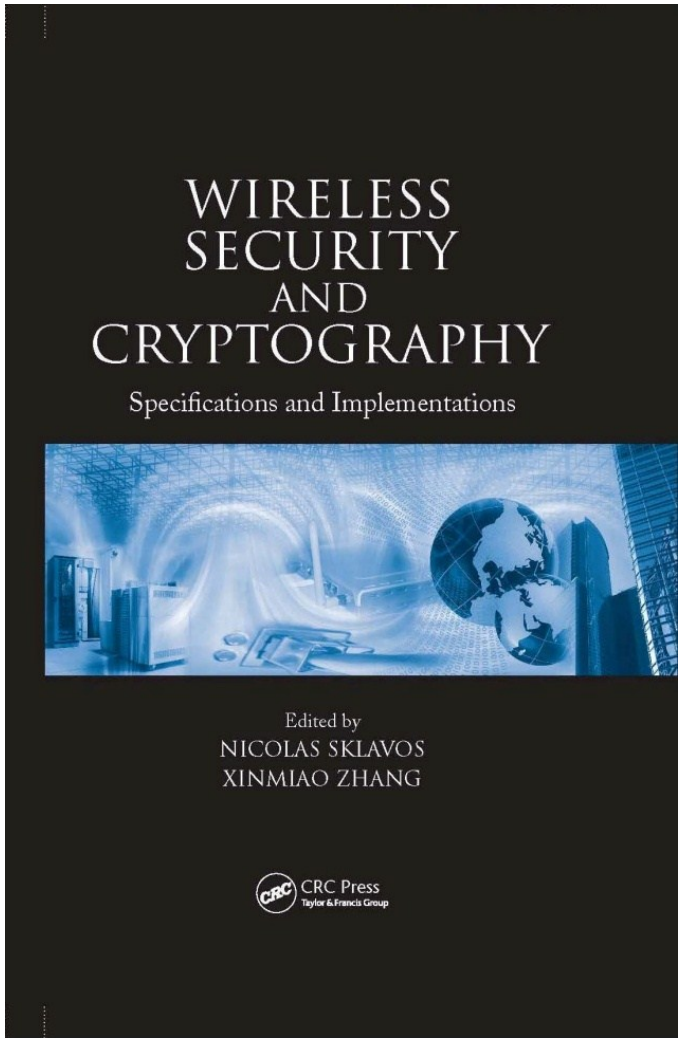
…In Detail:

SELECTIVE PUBLICATIONS - FURTHER DISCUSSION

...Some References: More Study !

- A. Mpikos, N. Sklavos, **"LTE/SAE Security Issues on 4G Wireless Networks"**, IEEE Security and Privacy, Issue March/April, Vol. 11, No. 2, pp. 55-62, 2013.
- N. Sklavos, O. Koufopavlou, **"Architectures and VLSI Implementations of the AES-Proposal Rijndael"**, IEEE Transactions on Computers, Vol. 51, Issue 12, pp. 1454-1459, 2002.
- P. Kitsos, N. Sklavos, K. Papadomanolakis, O. Koufopavlou, **"Hardware Implementation of the Bluetooth Security"**, IEEE Pervasive Computing, Mobile and Ubiquitous Systems, Volume 2, Number 1, pp. 21-29, 2003.
- N. Sklavos, O. Koufopavlou, **"Implementation of the SHA-2 Hash Family Standard Using FPGAs"**, Journal of Supercomputing, Springer-Verlag, Vol. 31, No 3, pp. 227-248, 2005.
- N. Sklavos, P. Kitsos, K. Papadopoulos, O. Koufopavlou, **"Design, Architecture and Performance Evaluation of the Wireless Transport Layer Security (WTLS)"**, Journal of Supercomputing, Springer-Verlag, Vol. 36, No 1, 2006.
- Nicolas Sklavos, Keynote Speaker, **"Towards to SHA-3 Hashing Standard for Secure Communications: On the Hardware Evaluation Development"**, 10th International Information and Telecommunication Technologies Symposium (I2TS'11), Brazil, December 19-21, 2011, IEEE Latin America Transactions, Volume 2012, Issue 1, 2012.

More Issues Could be Found:



N. Sklavos, X. Zhang,

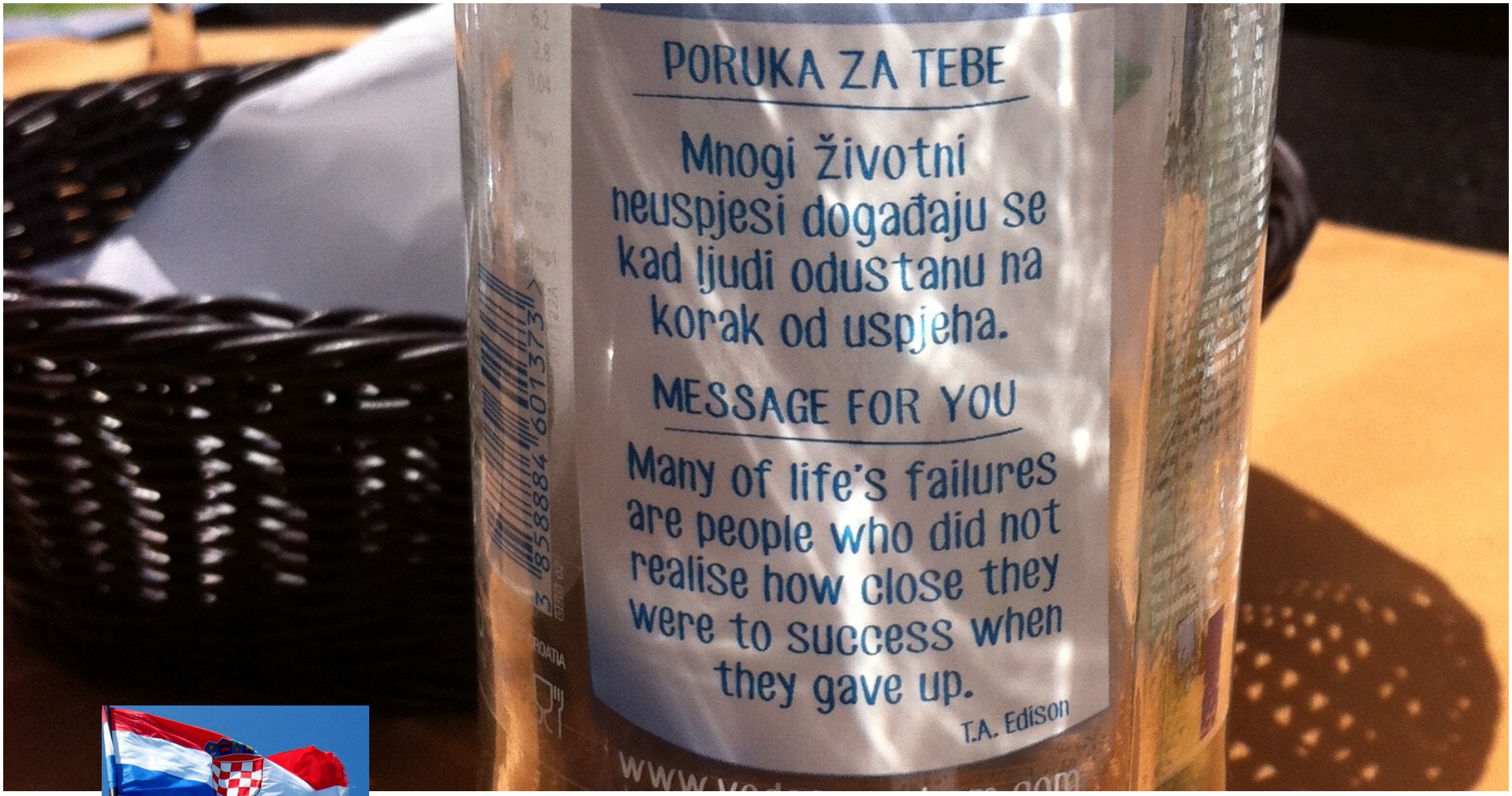
Wireless Security & Cryptography: Specifications and Implementations,

CRC-Press,
A Taylor and Francis Group,
ISBN: 084938771X, 2007.

PART V

CONCLUSIONS & OUTLOOK

… Message in a Croatian Bottle !



Questions & Further Discussion

Homepage :

www.KNOSSOSnet.gr/NiSk

For More Information Please Contact :

Nicolas Sklavos,

e-mail: nsklavos@ieee.org , nsklavos@ceid.upatras.gr