

Architecture Exploration for Efficient IPSec encryption: A Case Study

Hanno Scharwächter, David Kammler,
Andreas Wieferink, Matthias Stiefelhagen,
Manuel Hohenauer

Agenda

- **Overview of Network Layer Protocol IPv6**
 - **Virtual Private Networks using IPSec**
- **Encryption in Communication Channels**
- **Architecture Exploration**
 - **Instructions Development**
 - **Simulation**
 - **Synthesis**
- **Summary**

Agenda

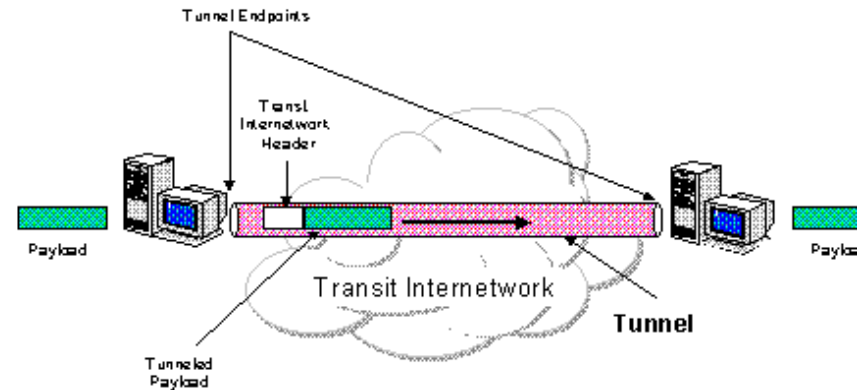
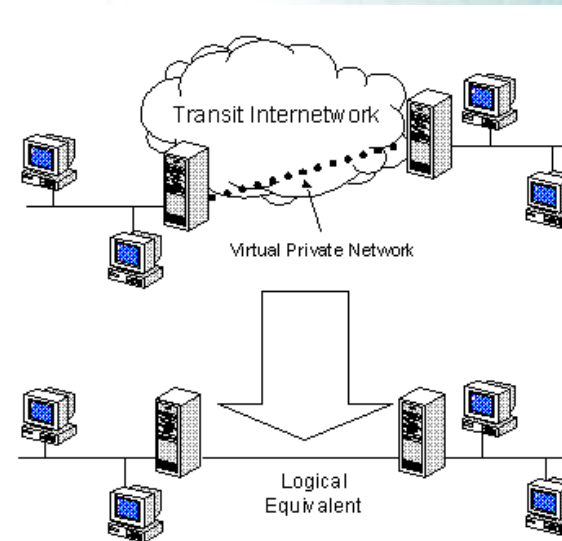
- **Overview of Network Layer Protocol IPv6**
 - Virtual Private Networks using IPsec
- Encryption in Communication Channels
- Architecture Exploration
 - Instructions Development
 - Simulation
 - Synthesis
- Summary

Network Layer

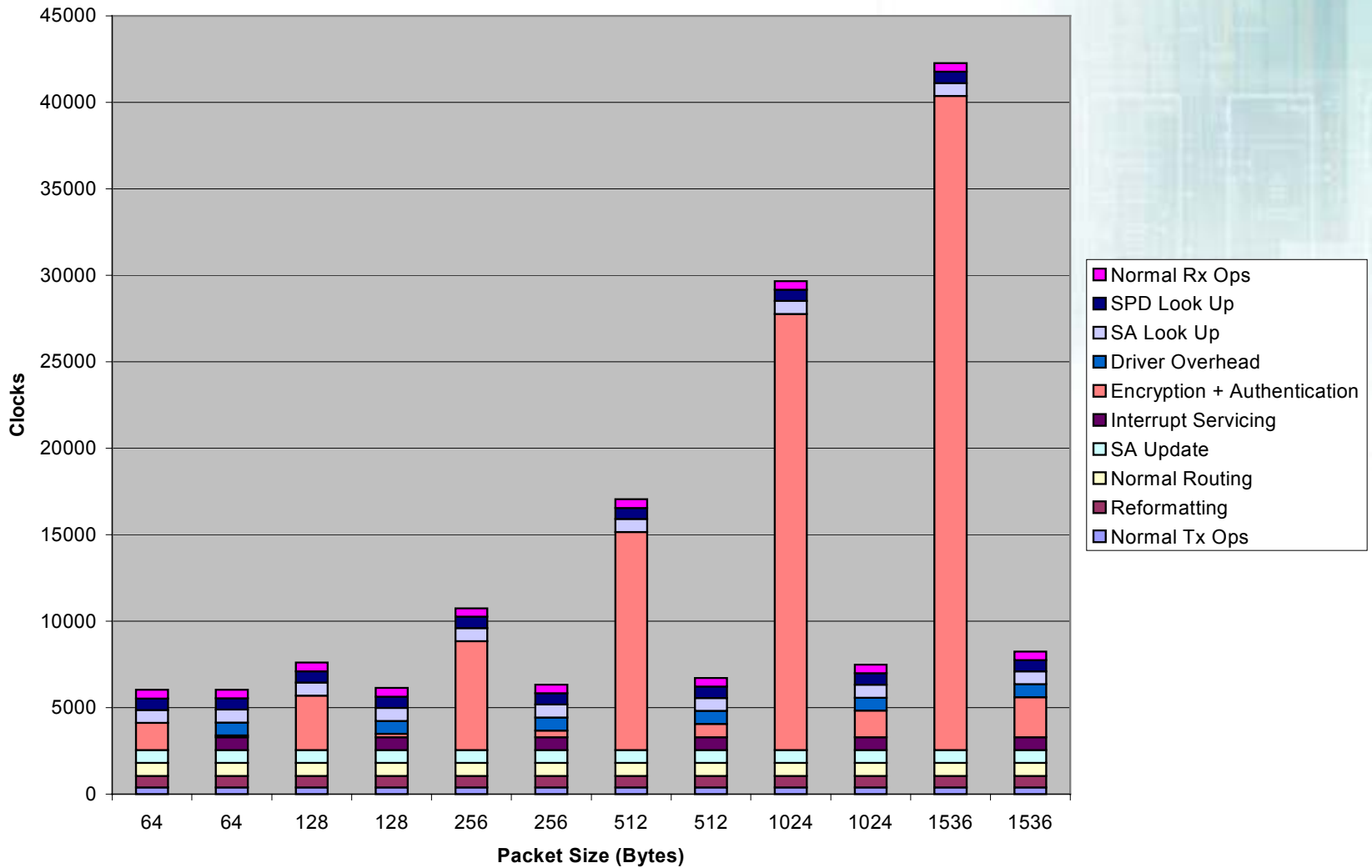
- Internet Protocol Version 6 (IPv6) is the next generation Internet protocol designed to overcome some of the limitations of the current protocol IPv4. The major differences are:
 - IPv6 uses 128-bit addresses (instead of 32-bit IPv4 addresses).
 - IPv6 does not calculate checksum.
 - IPv6 routers do not perform any fragmentation or reassembly.
 - IPv6 includes a security protocol, IPSec.

Virtual Private Network (VPN)

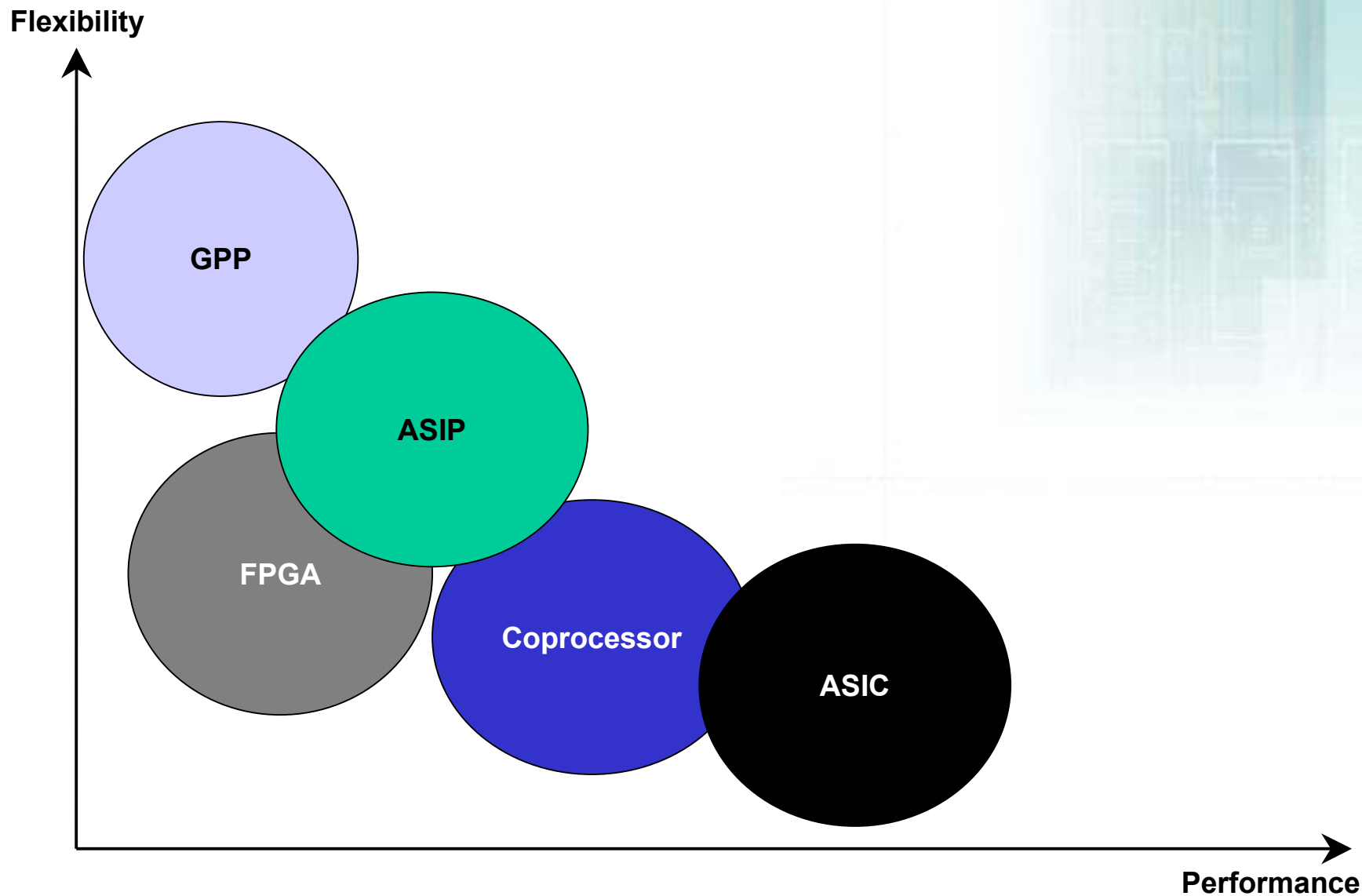
- VPN connect the components and resources of one network to another by tunneling data through public internet.
- Tunnel participants enjoy the same security and features available only in private networks.
- Tunneling means, that the transferred data packets are encapsulated inside another protocol packet as payload.



Break-up of Tasks in Typical VPN Traffic



Space of System Implementation



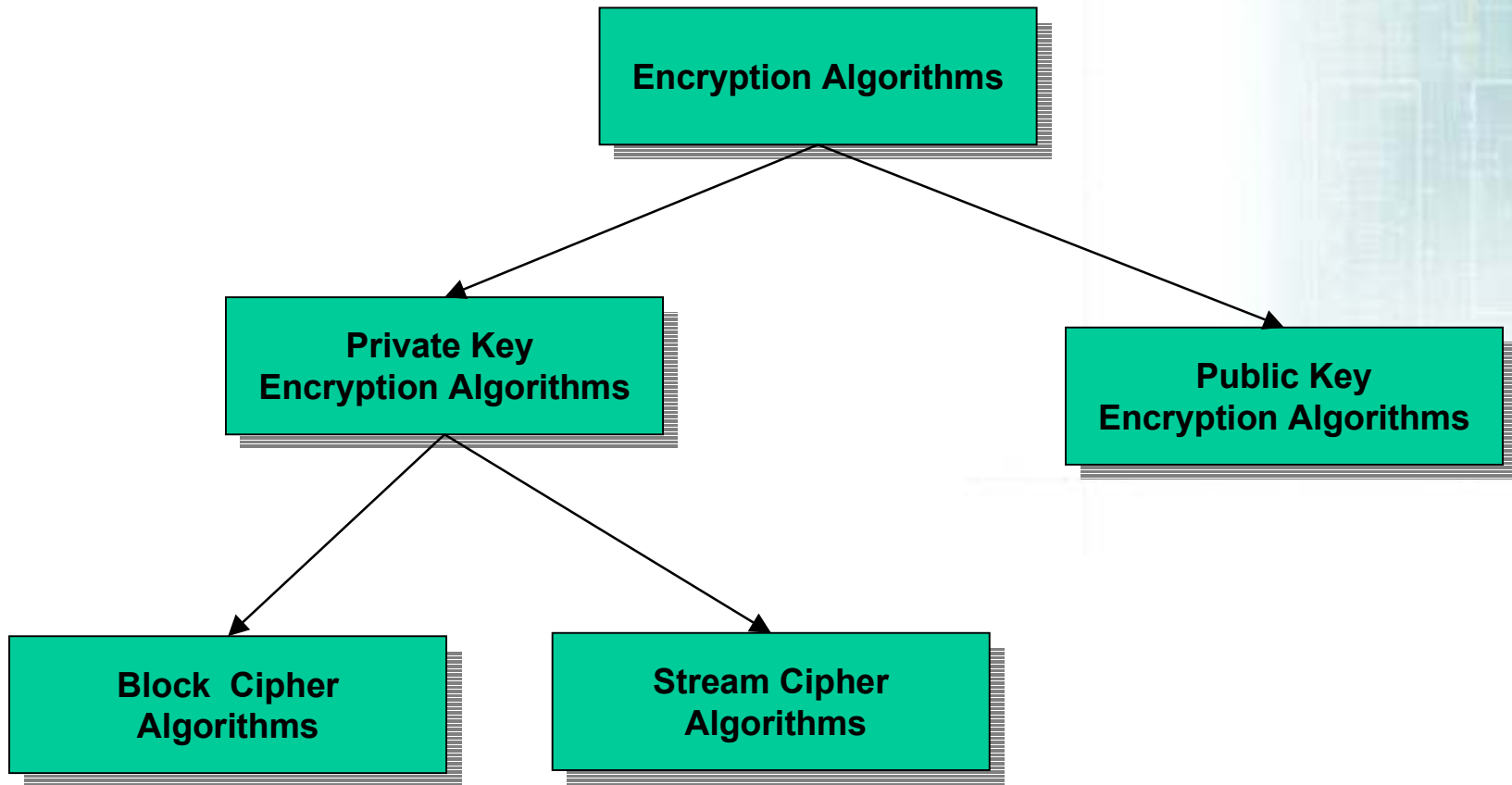
Agenda

- **Overview of Network Layer Protocol IPv6**
 - **Virtual Private Networks using IPsec**
- **Encryption in Communication Channels**
- **Architecture Exploration**
 - **Instructions Development**
 - **Simulation**
 - **Synthesis**
- **Summary**

Agenda

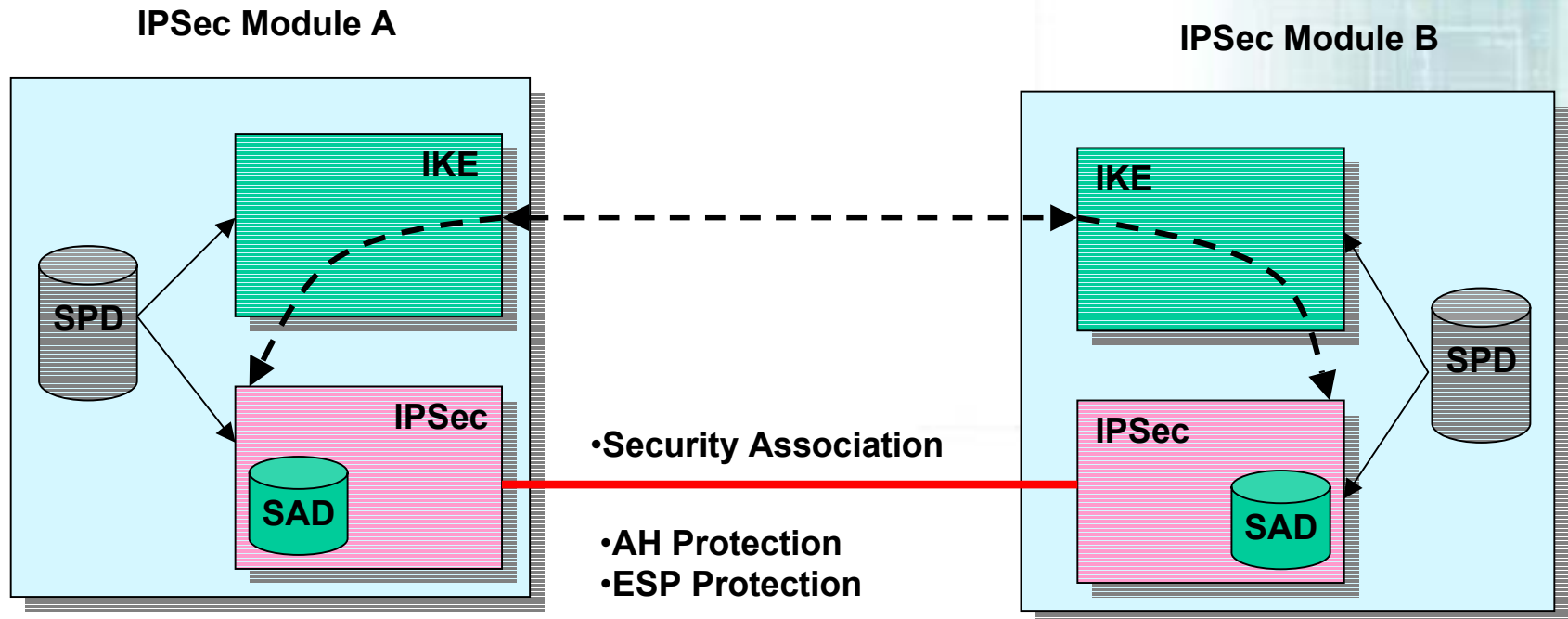
- **Overview of Network Layer Protocol IPv6**
 - Virtual Private Networks using IPsec
- **Encryption in Communication Channels**
- **Architecture Exploration**
 - Instructions Development
 - Simulation
 - Synthesis
- **Summary**

Encryption Algorithms



- DES, AES
- Blowfish
- IDEA

IPSec-Concepts



SA = Security Association
AH = Authentication Header
IKE = Internet Key Exchange

SPD = Security Policy Database
SAD = Security Association Database
ESP = Encapsulated Security Payload

IPSec Encryption Requirements

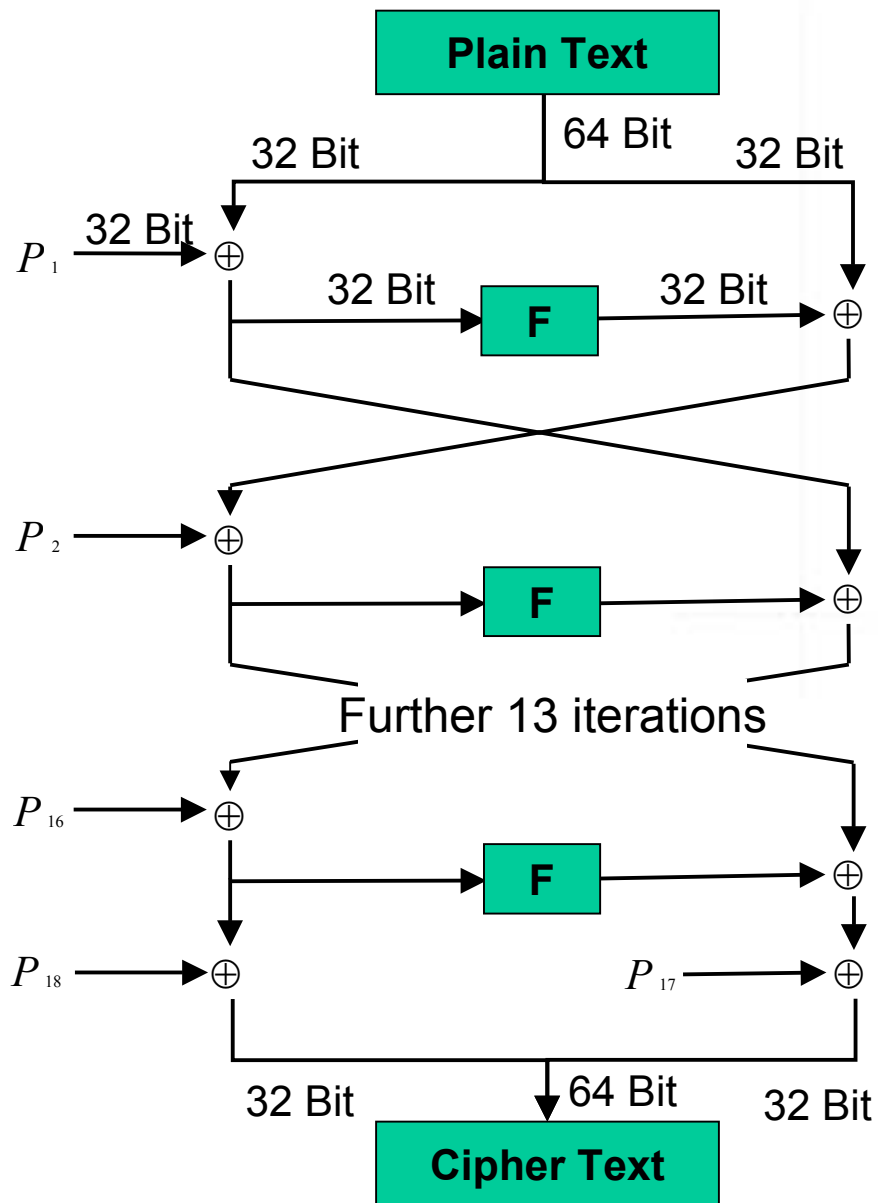
- 64-Bit symmetric block cipher algorithm in Cipher-Block-Chaining (CBC) mode
- DES is standard
- Alternative algorithms may be used
- For IPSec encryption, you can select a public algorithm, assuming that a publicly known algorithm is well examined by many cryptographers: If nobody has been able to crack it yet, it might be a quite good algorithm.

Blowfish Encryption Algorithm

- Designed to encrypt data very efficiently on 32 bit processors
- Significantly faster than DES
- Needs less than 5 KB memory
- It is publicly available
- No attacks known against it for 16 rounds

- Blowfish only uses simple operations like ADD, XOR or memory access. Since the design is easy to analyse, implementation errors can be avoided.
- **Blowfish's design is a Feistel-Network, which is common to most of the block cipher algorithms.**

Blowfish Encryption Data Flow



Legend:

- P designates subkeys
- F is a substitution
- \oplus designates XOR

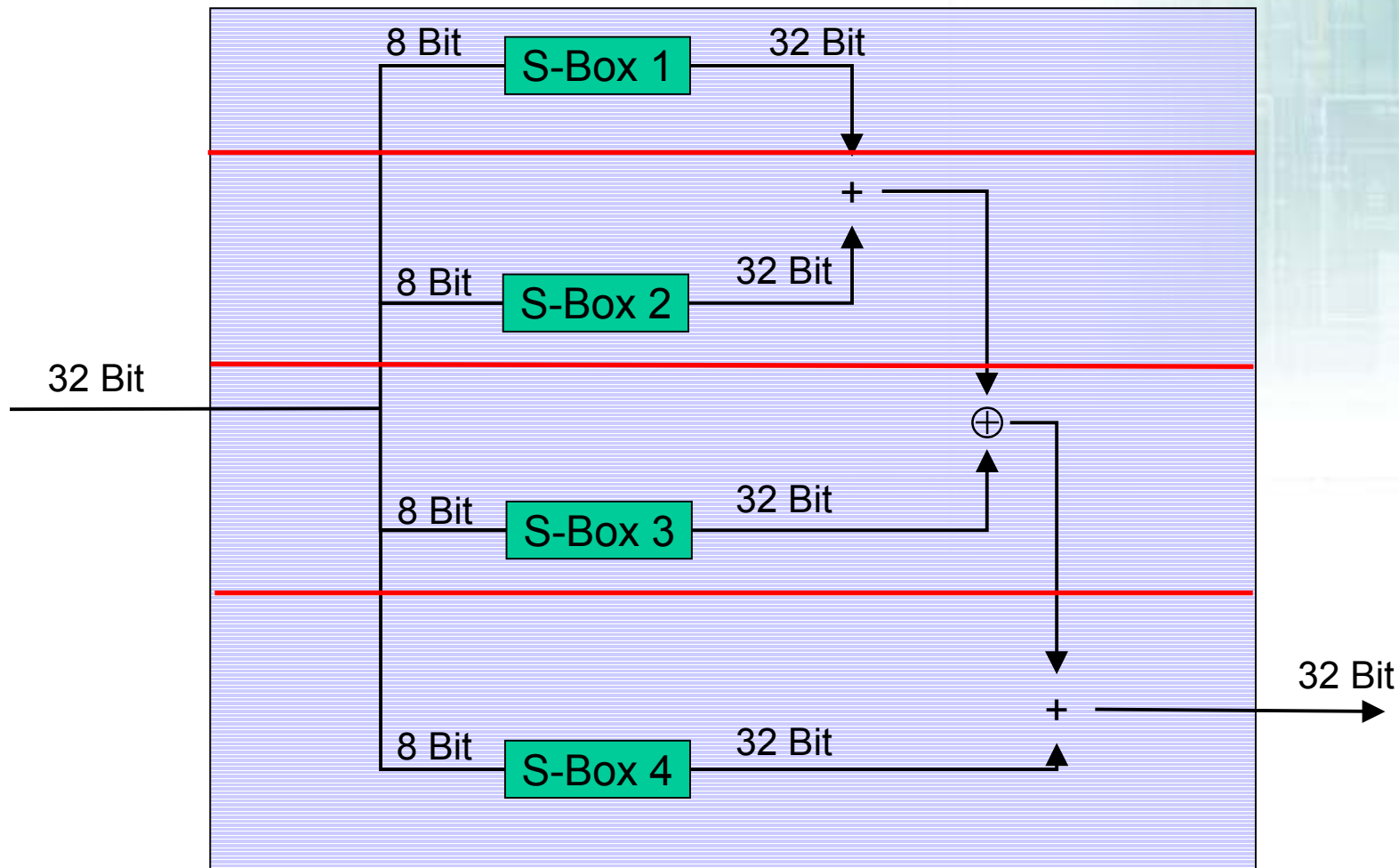
Agenda

- **Overview of Network Layer Protocol IPv6**
 - **Virtual Private Networks using IPsec**
- **Encryption in Communication Channels**
- **Architecture Exploration**
 - **Instructions Development**
 - **Simulation**
 - **Synthesis**
- **Summary**

Agenda

- **Overview of Network Layer Protocol IPv6**
 - Virtual Private Networks using IPsec
- **Encryption in Communication Channels**
- **Architecture Exploration**
 - Instructions Development
 - Simulation
 - Synthesis
- **Summary**

Function F (Exploration Iteration 1)



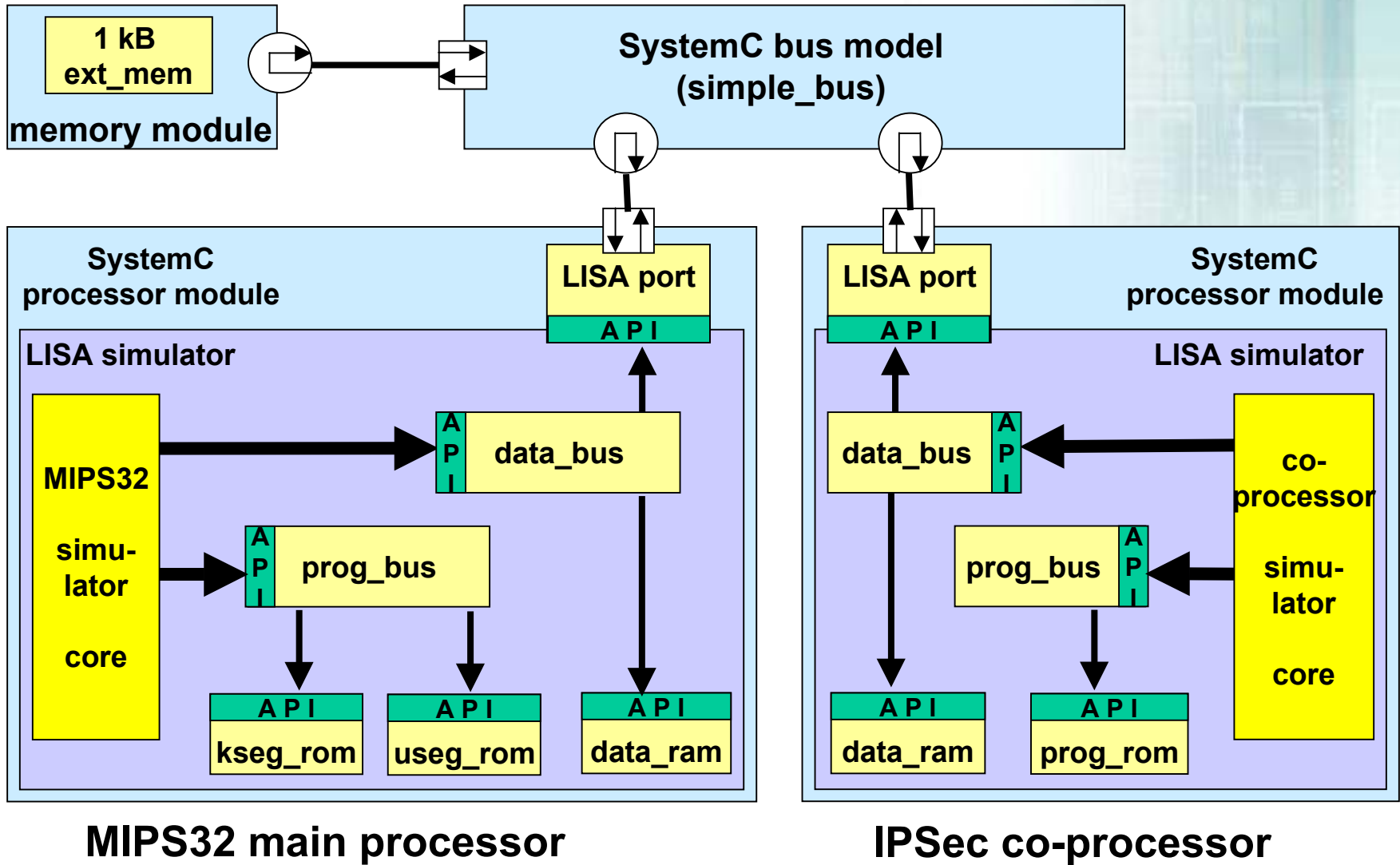
Platform Requirements

- Implementing dedicated hardware instructions on a coprocessor, results in greater flexibility for system integration.
- This approach requires:
 - Multi-Processor Simulation Platform
 - Hardware Synthesis Tools
 - C Compiler
 - Assembler, Linker etc.

Related Work

- A. Halambi, P. Grun, V: Ganesh, A. Khare, N.Dutt and A. Nicolau. EXPRESSION: A Language for Architecture Exploration through Compiler/Simulator Retargetability.
- D. Lanner, J. van Praet, A. Kifli, K. Schoofs, W. Geurts, F. Thoen and G. Goessens. Chess: Retargetable Code Generation for Embedded DSP Processors.
- S. Kobayashi, Y. Takeuchi, A. Kitajima and M. Imai. Compiler Generation in PEAS-III: an ASIP Development System.
- A. Hoffmann, H. Meyr and R. Leupers. Architecture Exploration for Embedded Processors with LISA.

LISA Simulator Structure

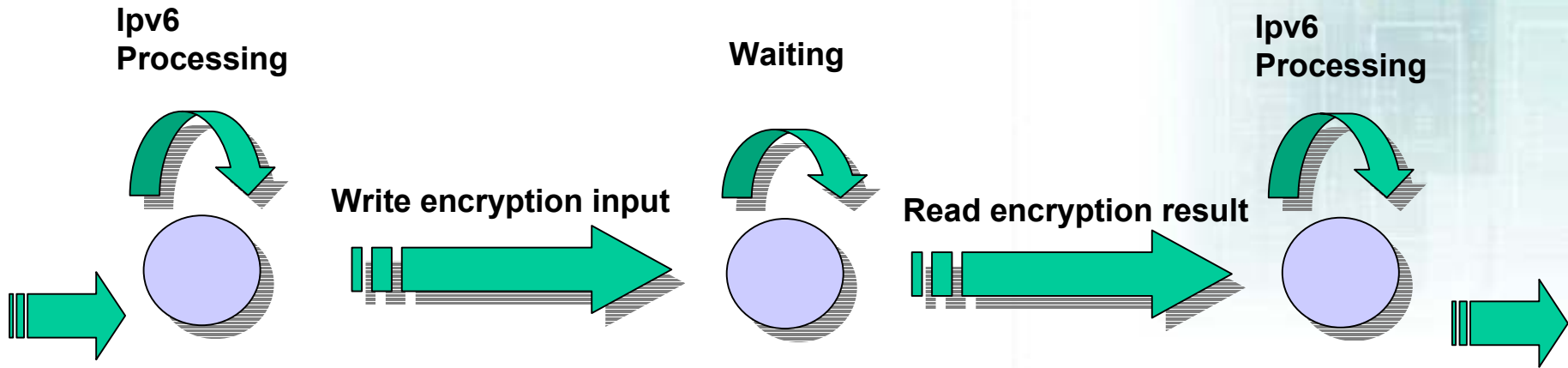


MIPS32 main processor

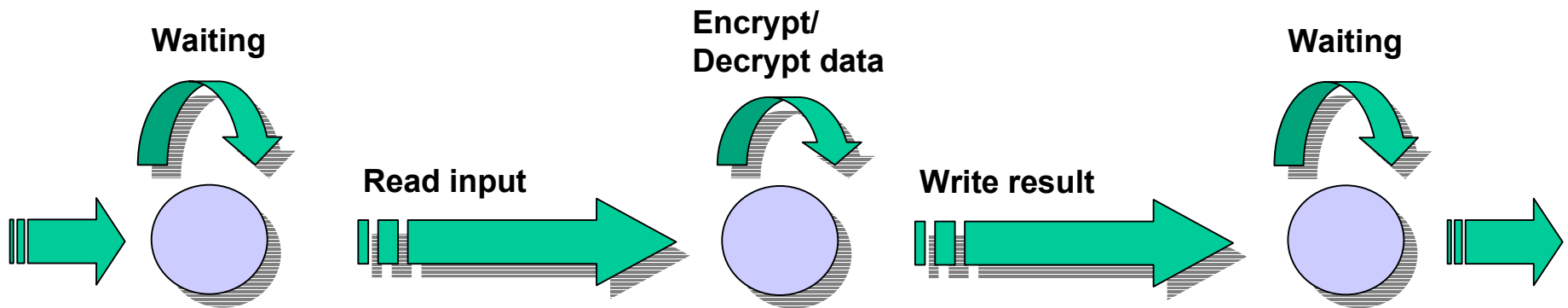
IPSec co-processor

Coprocessor/Processor States

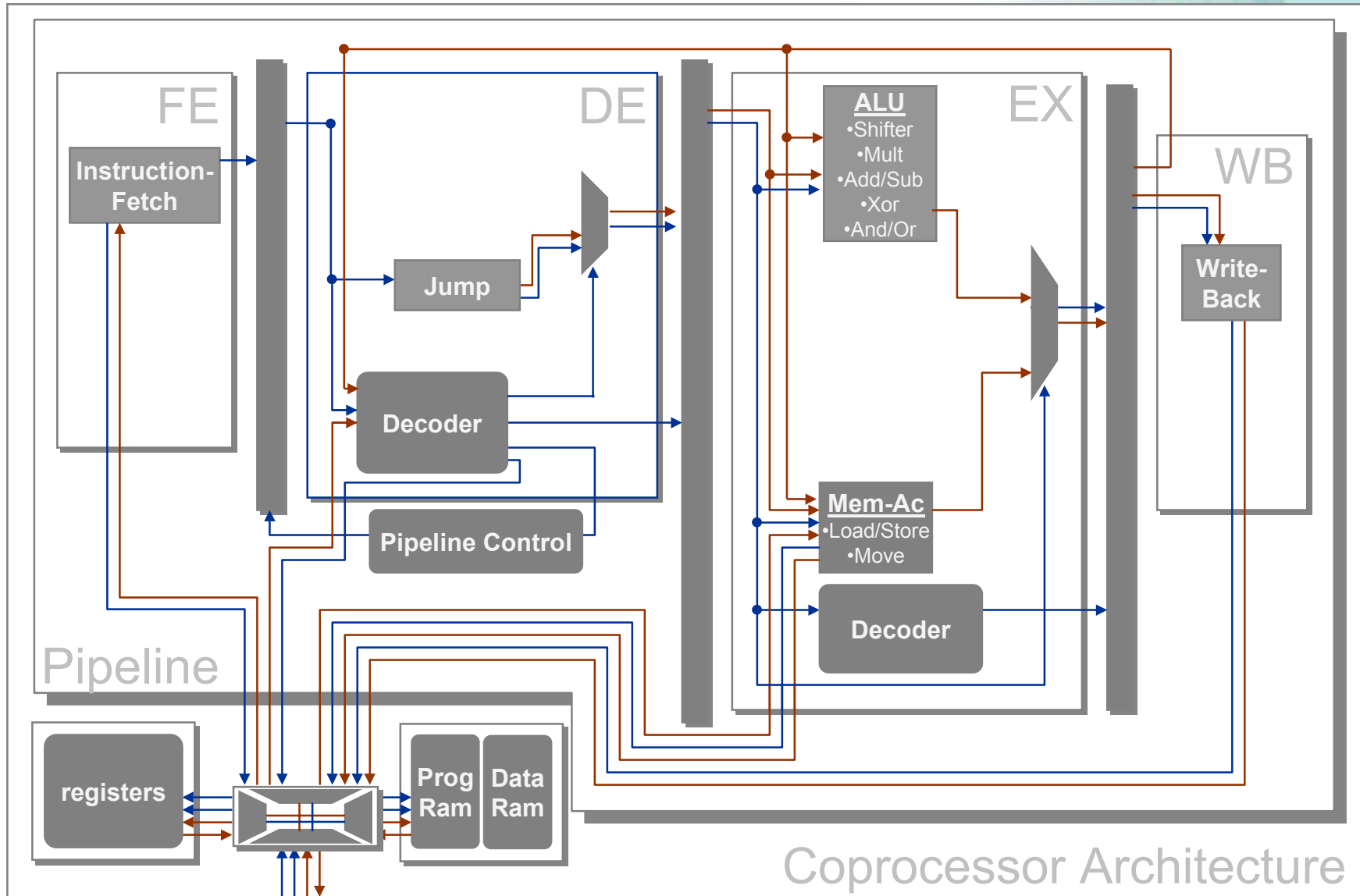
MIPS processing states:



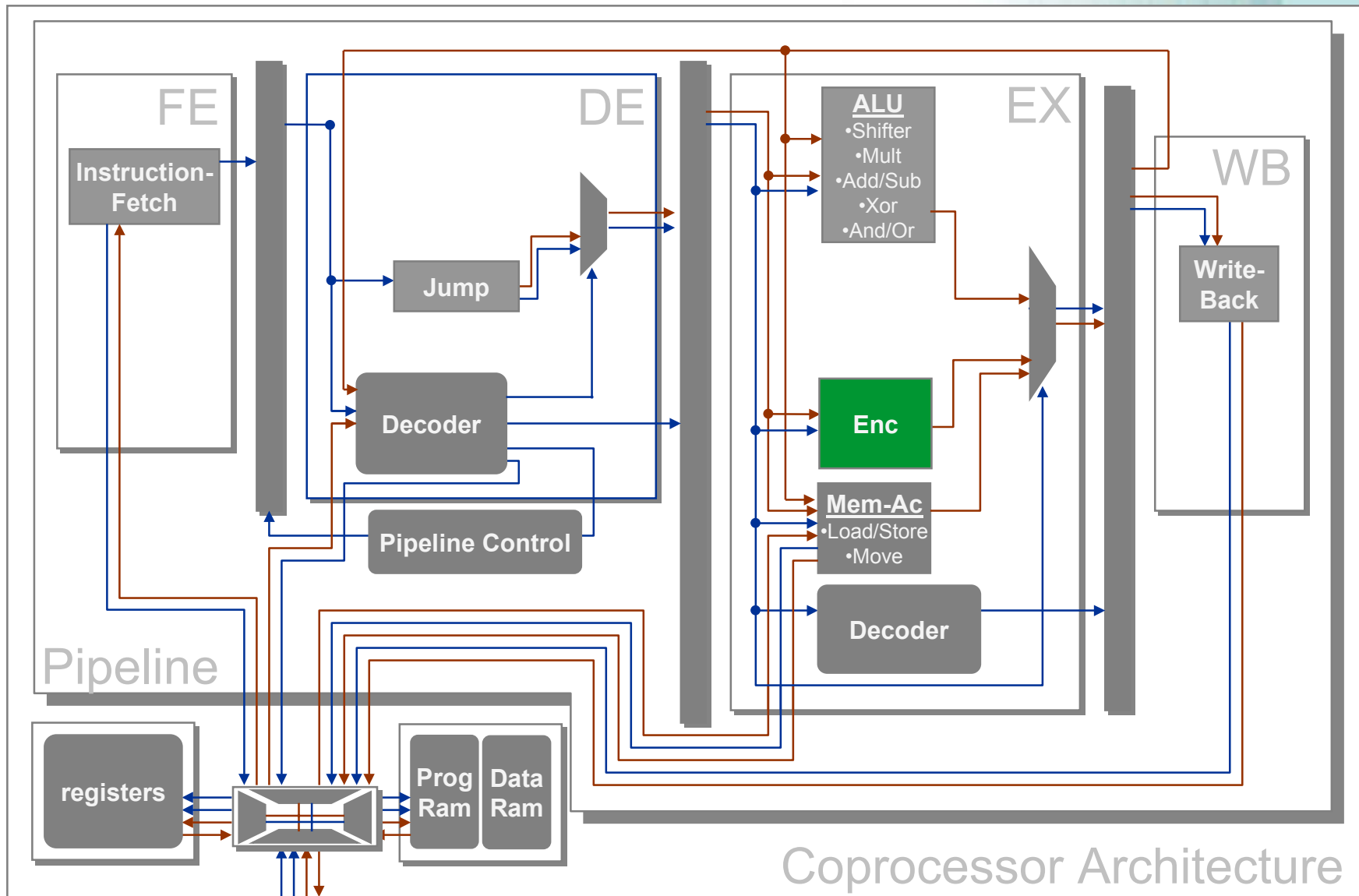
Coprocessor processing states:



Generating the HDL Model



Generating the HDL Model (Simulation Iteration 1)



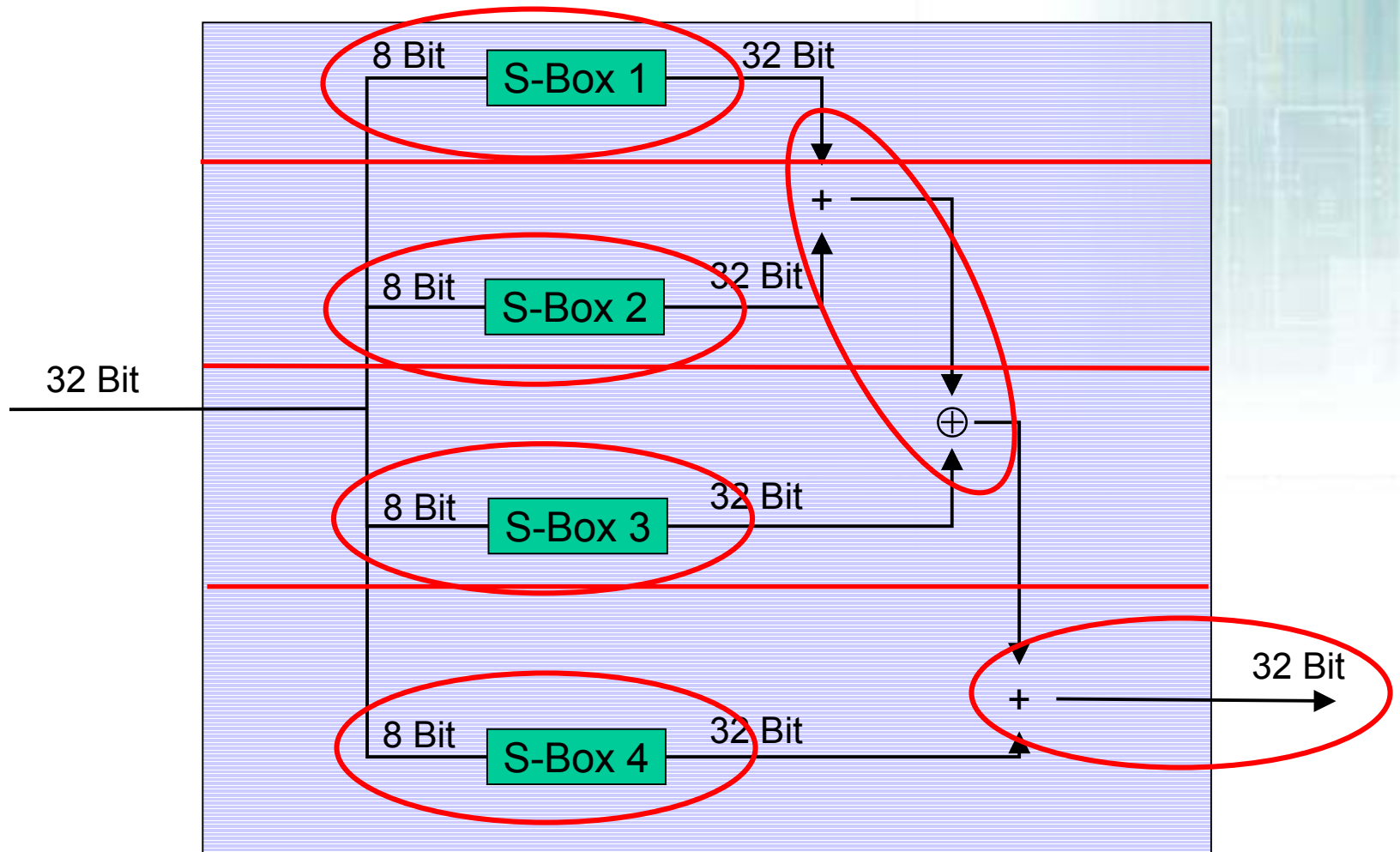
Experimental Results of the Exploration

- Since coprocessor and MIPS run with the same clock, they have to provide the same cycle length.
- Results have been obtained without applying any technology factor
- Critical path of the Coprocessor does not take memory latency into account

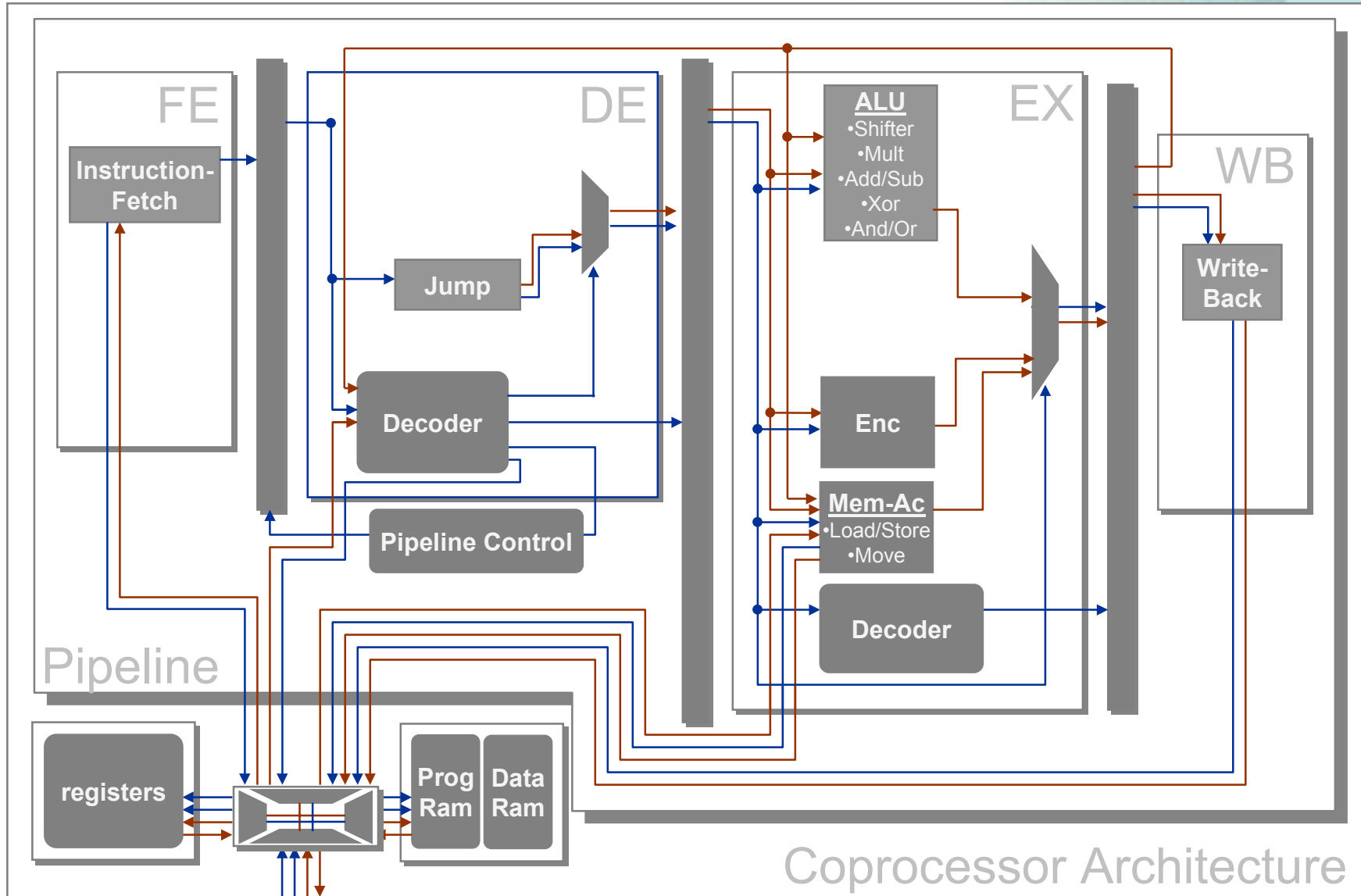
	Area Consumption (kG)	Timing (ns)	Executed Instructions
MIPS	70-300	5	917844
Coprocessor	52	4.5	117546

Coprocessor's speed exceeds the MIPS clock cycle length!

Function F (Exploration Iteration 2)



Generating the HDL Model



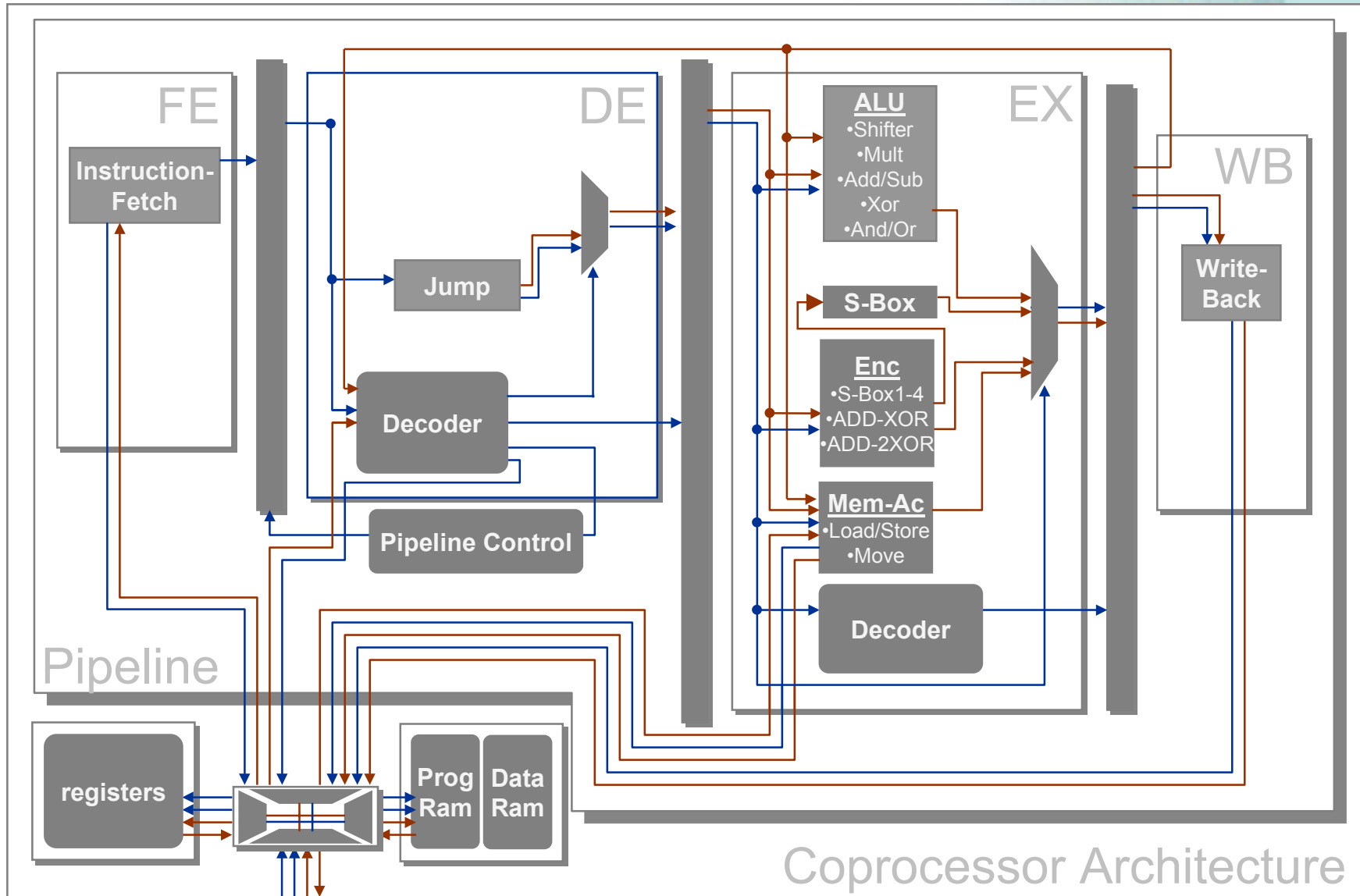
Simulation Results

- Test setup consisted of calls to
 - key generation procedure
 - 40 bytes data encryption/decryption procedure
- Coprocessor runs with 190 MHz in worst case!

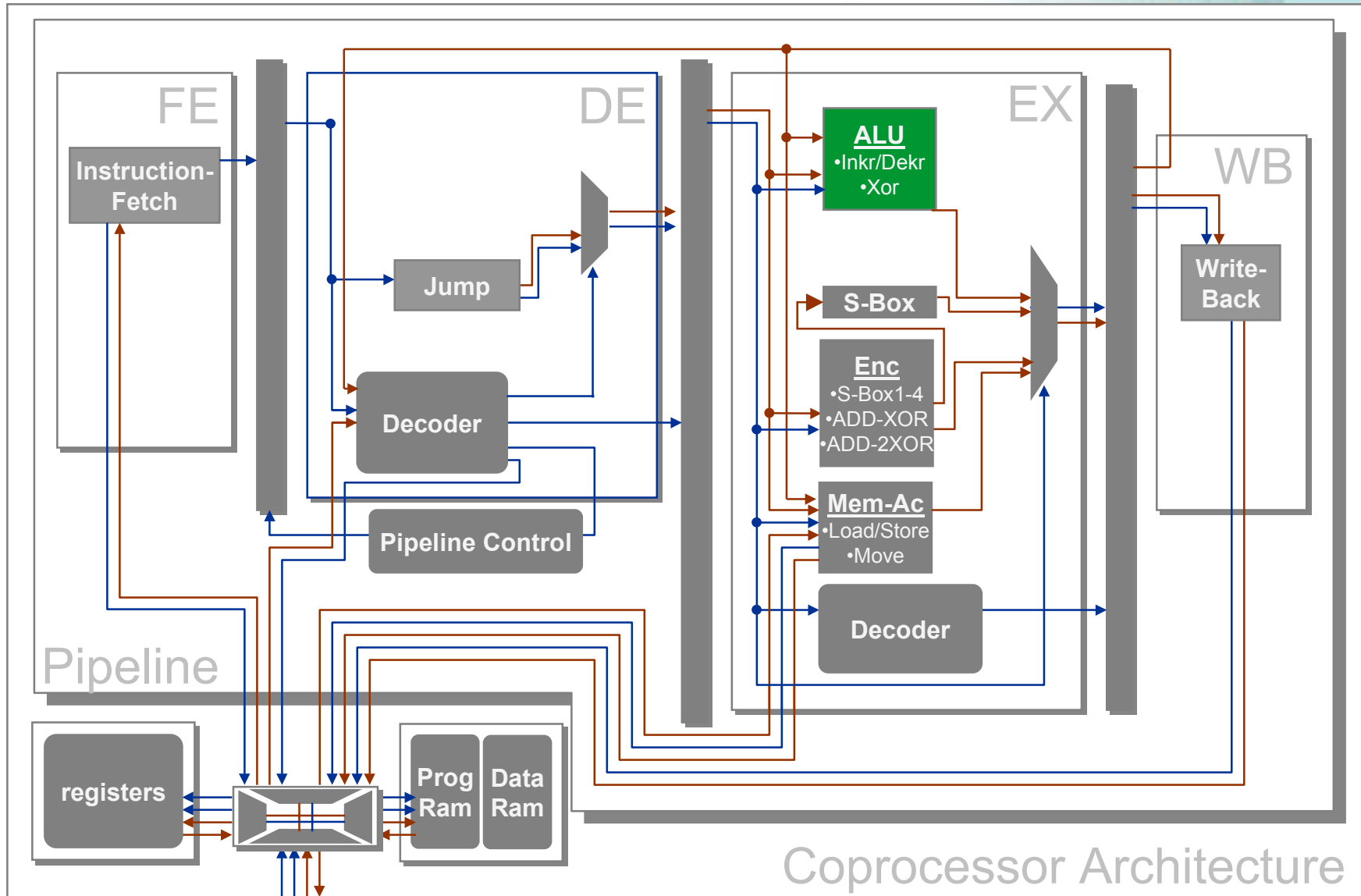
	Simulation 0: (Standalone Simulation on MIPS)	Simulation 1: (Simulation with 4 instructions)	Simulation 2: (Simulation with 6 instructions)
Code size	531	235	267
Number of cycles	917844	117546	176319

**Overall encryption speedup by a factor of 5!
Code size has been reduced by 50%!**

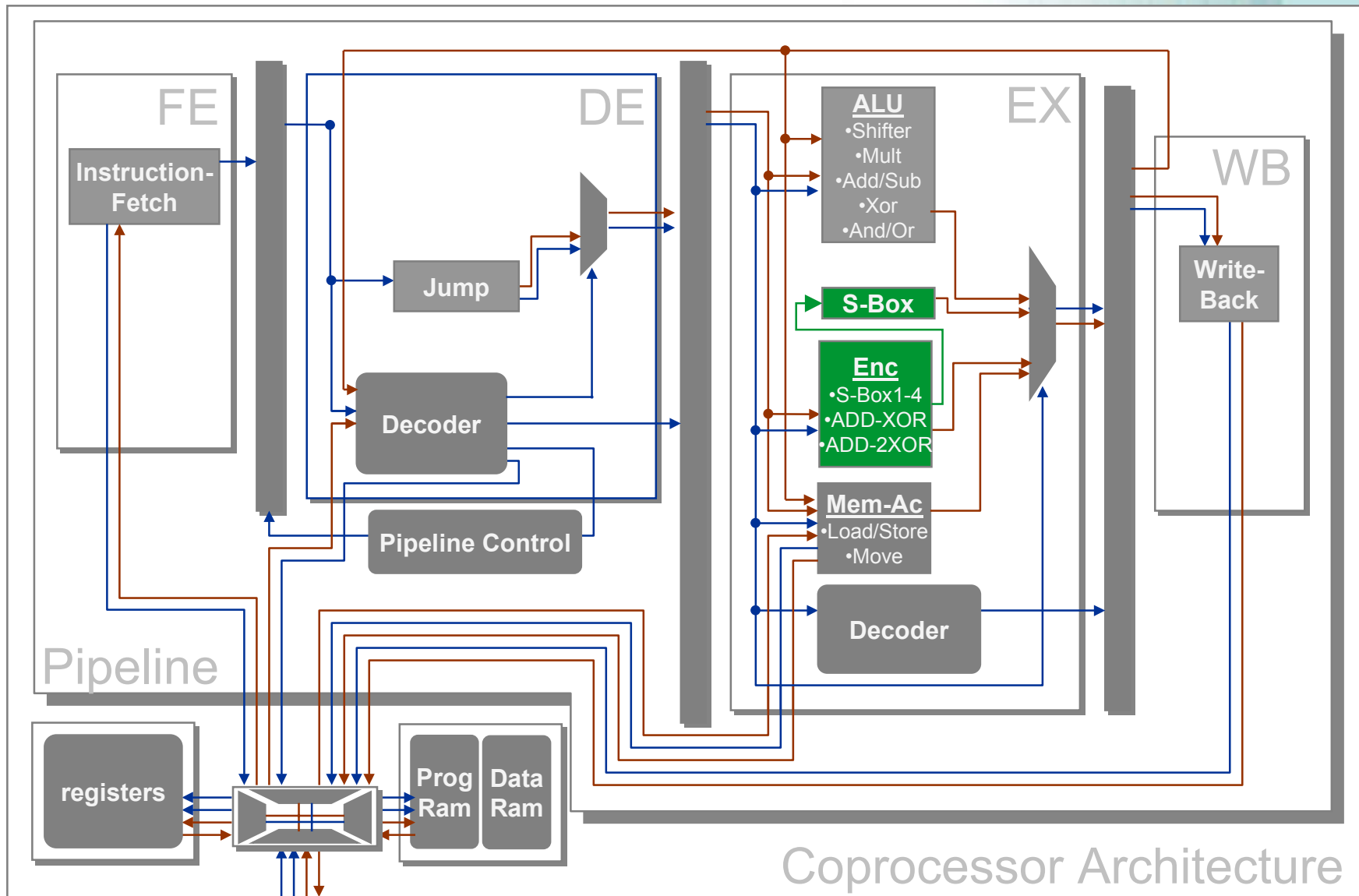
Generating the HDL Model (Synthesis Iteration 1)



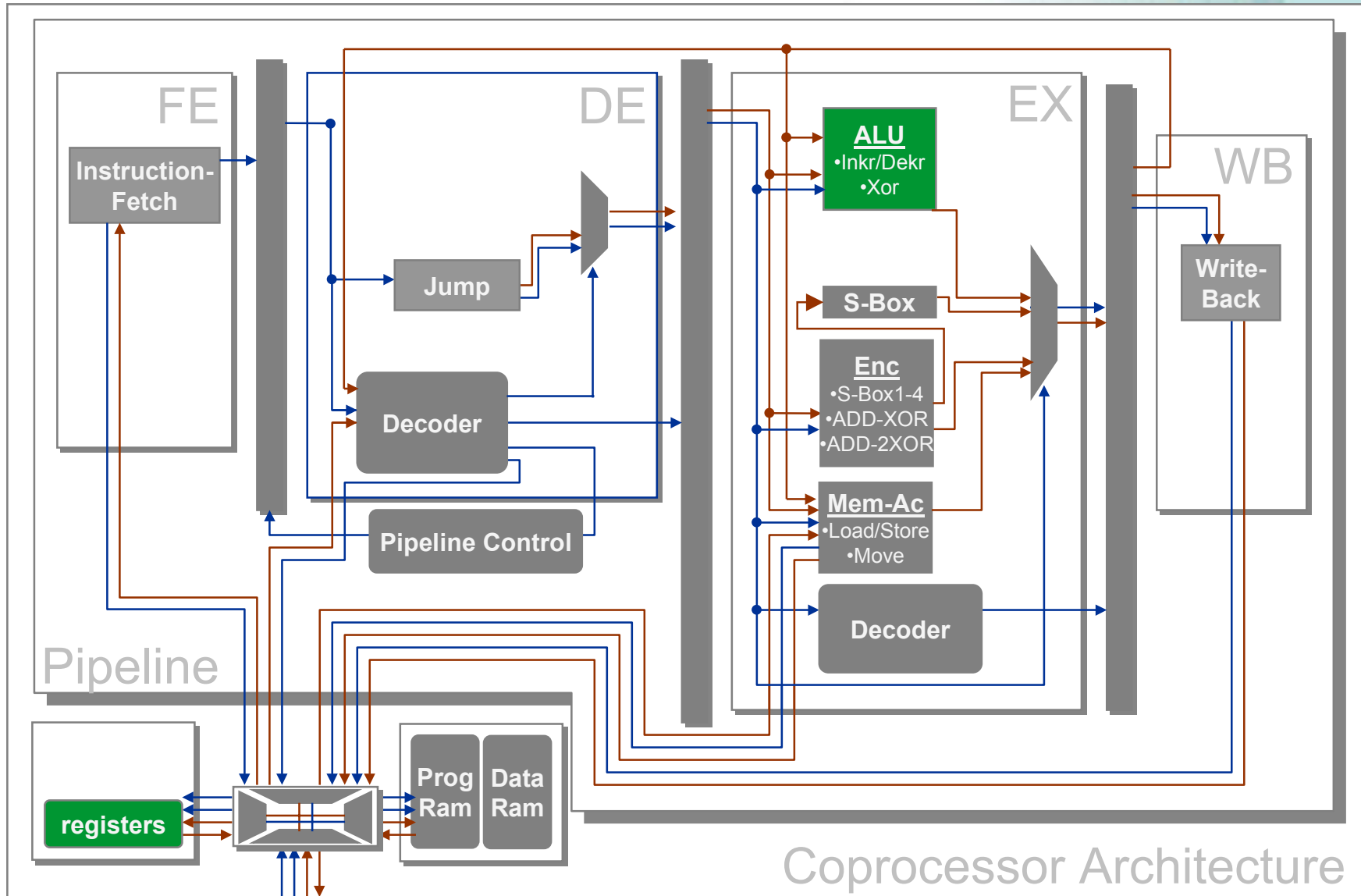
Generating the HDL Model (Synthesis Iteration 2)



Generating the HDL Model (Simulation Iteration 2)



Generating the HDL Model (Synthesis Iteration 3)



Synthesis Results (in kGates)

- Reference area consumption of the MIPS relies between 70 and 300 kilo Gates (kGates)

	Synthesis 1: (extended by Encryption Instructions)	Synthesis 2: (eliminated redundant instructions)	Synthesis 3: (with reduced register file)
total	31.4	25.8	22.2
Pipeline	21.1	15.0	14.9
Register file	10.1	10.5	7.1

**Coprocessor's area consumption is at most
30% of the MIPS architecture!**

Summary

- Encryption is one of the primary bottlenecks in IPsec and IPv6 processing, respectively.
- We developed a coprocessor that supports implementation of symmetric block cipher algorithms by customized instructions.
- Doing this we obtained:
 - Speed-up by a factor of 5 for the Blowfish algorithm.
 - Reusability by ASIP design approach for other block cipher algorithms.
 - Loose coupling by establishing communication via shared memory.

Thank you!