# A 1 Gbit/s Partially Unrolled Architecture of Hash Functions SHA-1 and SHA-512

Roar Lien, Tim Grembowski, and Kris Gaj

George Mason University

RSA Conference 2004

# Motivation
# & Problem Statement

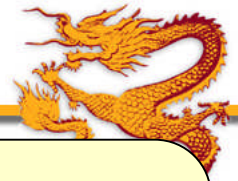
# Hash functions

- standard building block of digital signatures and message authentication codes (MACs)
- used in most common secure protocols, such as SSL and IPSec
- first American standard SHA approved in 1993, and revised in 1995
- in August 2002 four algorithms approved by NIST as American government standard FIPS 180-2

# Hash algorithms supported by the current NIST standard

| Hash algorithm | Complexity of the best attack | Secret-key cipher with equivalent security |
|---|---|---|
| SHA-1 | $2^{80}$ | Skipjack |
| SHA-256 | $2^{128}$ | AES-128 |
| SHA-384 | $2^{192}$ | AES-192 |
| SHA-512 | $2^{256}$ | AES-256 |

# Primary ways of implementing cryptography in hardware



**ASIC**
**A**pplication **S**pecific
**I**ntegrated **C**ircuit

- designs must be sent for expensive and time consuming fabrication in semiconductor foundry
- designed all the way from behavioral description to physical layout

**FPGA**
**F**ield **P**rogrammable
**G**ate **A**rray

- bought off the shelf and reconfigured by designers themselves
- no physical layout design; design ends with a bitstream used to configure a device

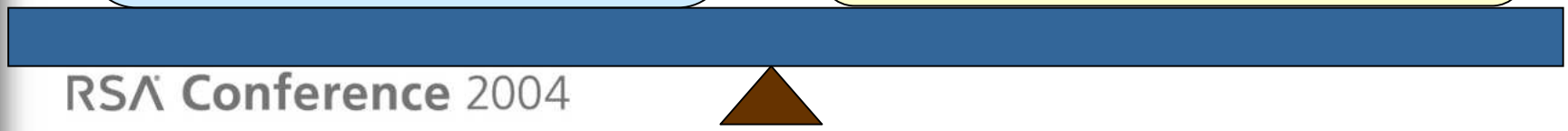
# Which way to go?

**ASICs**

- High performance
- Low power
- Low cost (but only in high volumes)

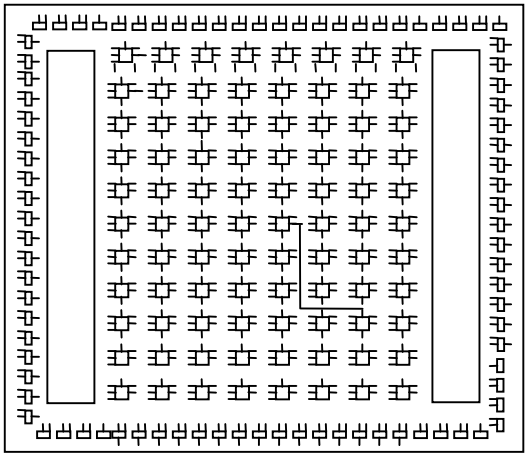
**FPGAs**

- Off-the-shelf
- Low development costs
- Short time to the market
- Reconfigurability



# Hash functions at medium cost FPGA devices

## Target: Xilinx Virtex - XCV 1000



- 1 mln equivalent logic gates
- 12 288 CLB slices
- 32  4-kbit block RAMs

## Best reported performance

SHA-1, Helion Technology Limited - 480 Mbit/s

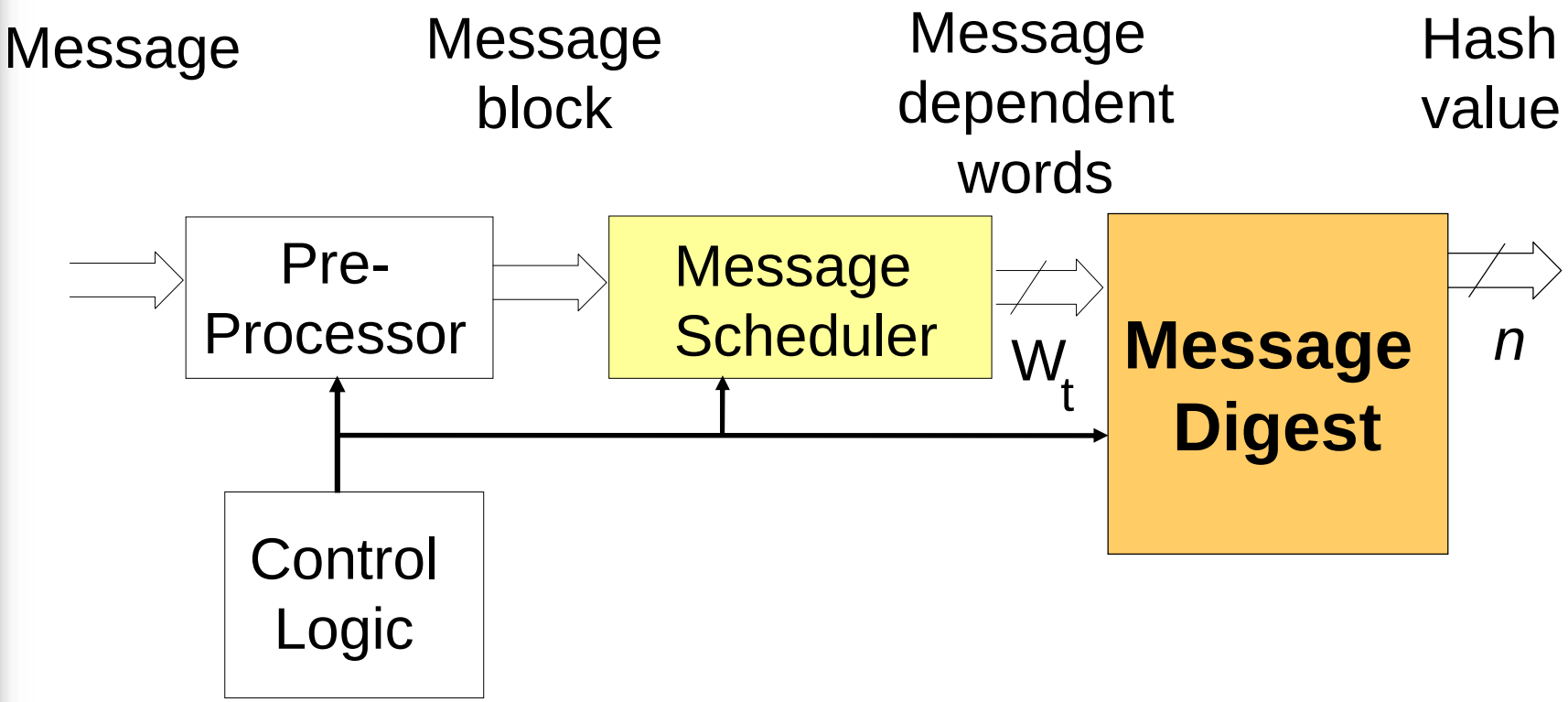
SHA-512, ALMA Technologies - 707 Mbit/s

# Our objectives

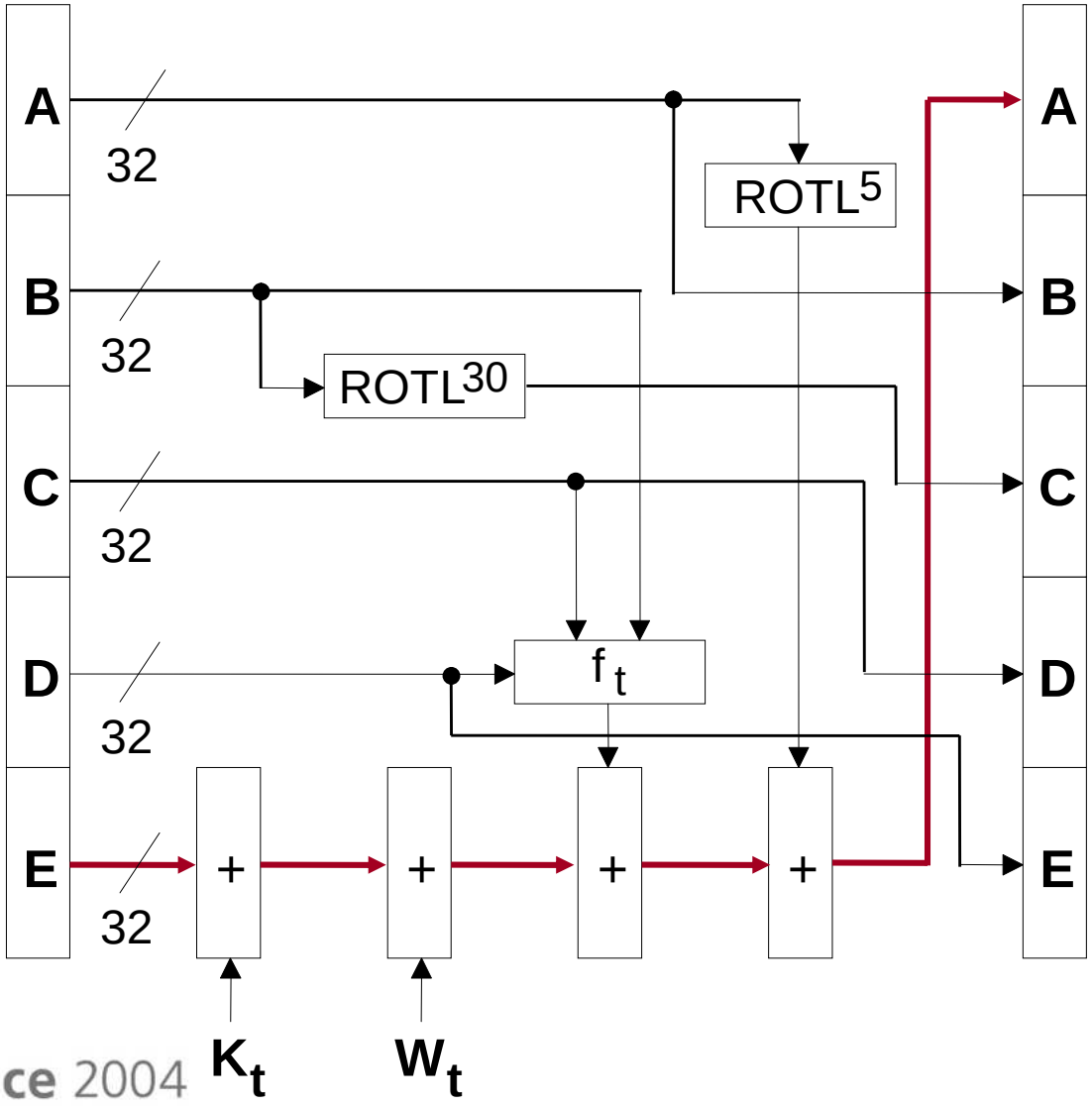
- **New hardware architecture**
- **Throughput ≥ 1Gbps**
- **Platform:**
  **Medium-size FPGAs, such as Xilinx Virtex 1000**
  **(1 mln equivalent logic gates)**
- **Portability**
  **Easy transfer to other FPGAs & other technologies**

# Standard architectures of dedicated hash functions

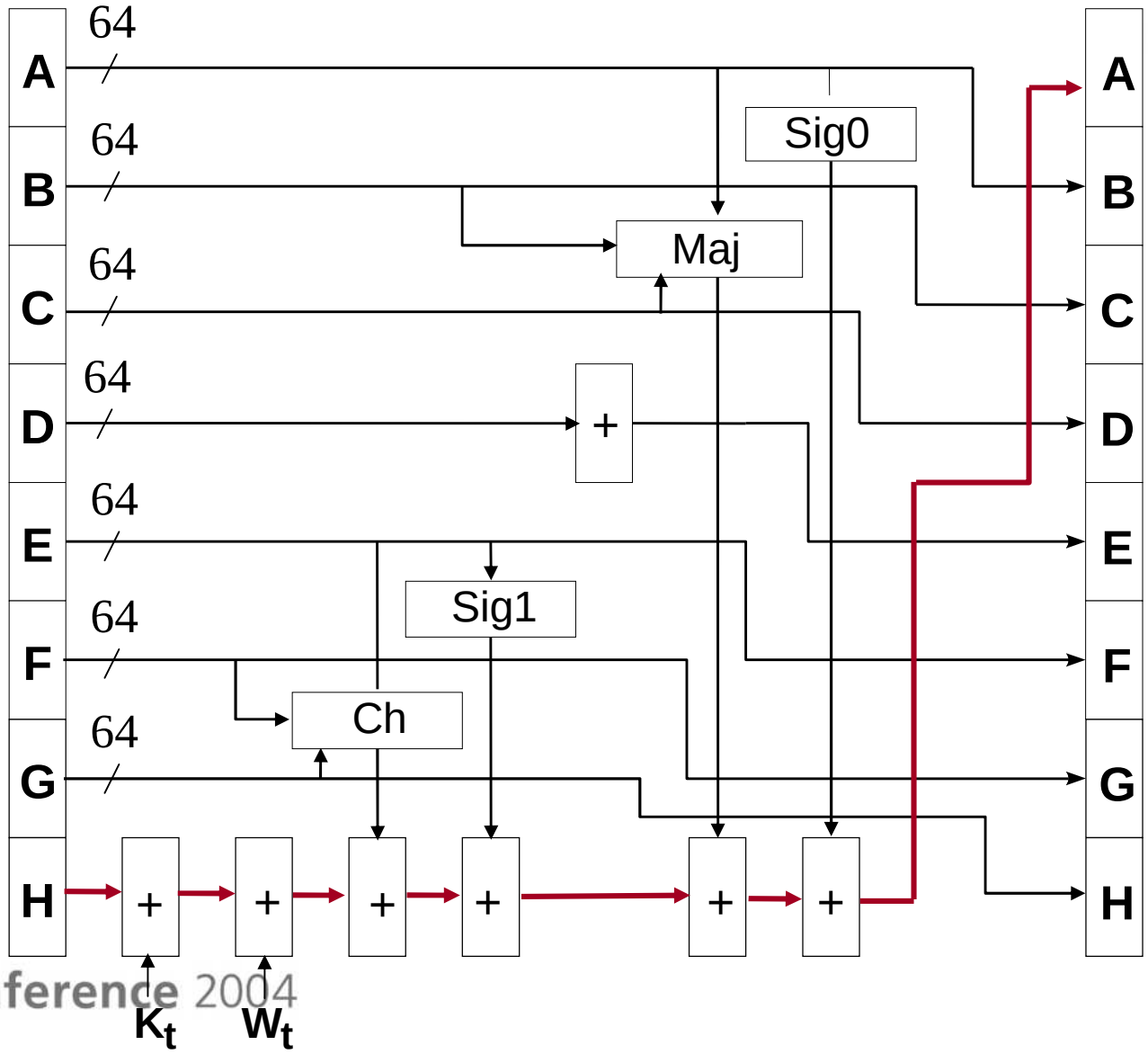
# Hardware implementation of a dedicated hash function



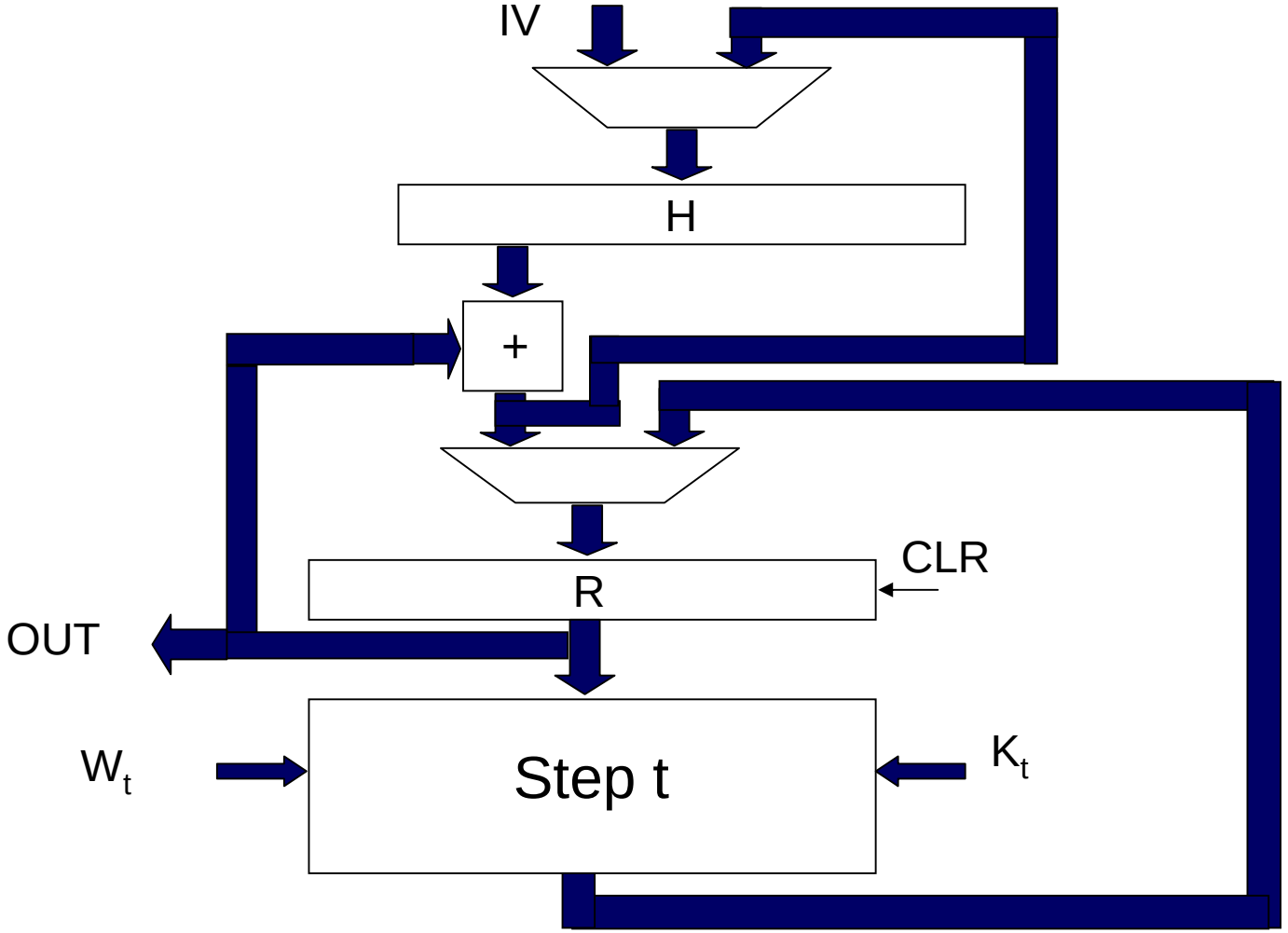
# Message digest step of SHA-1
# Functional block diagram



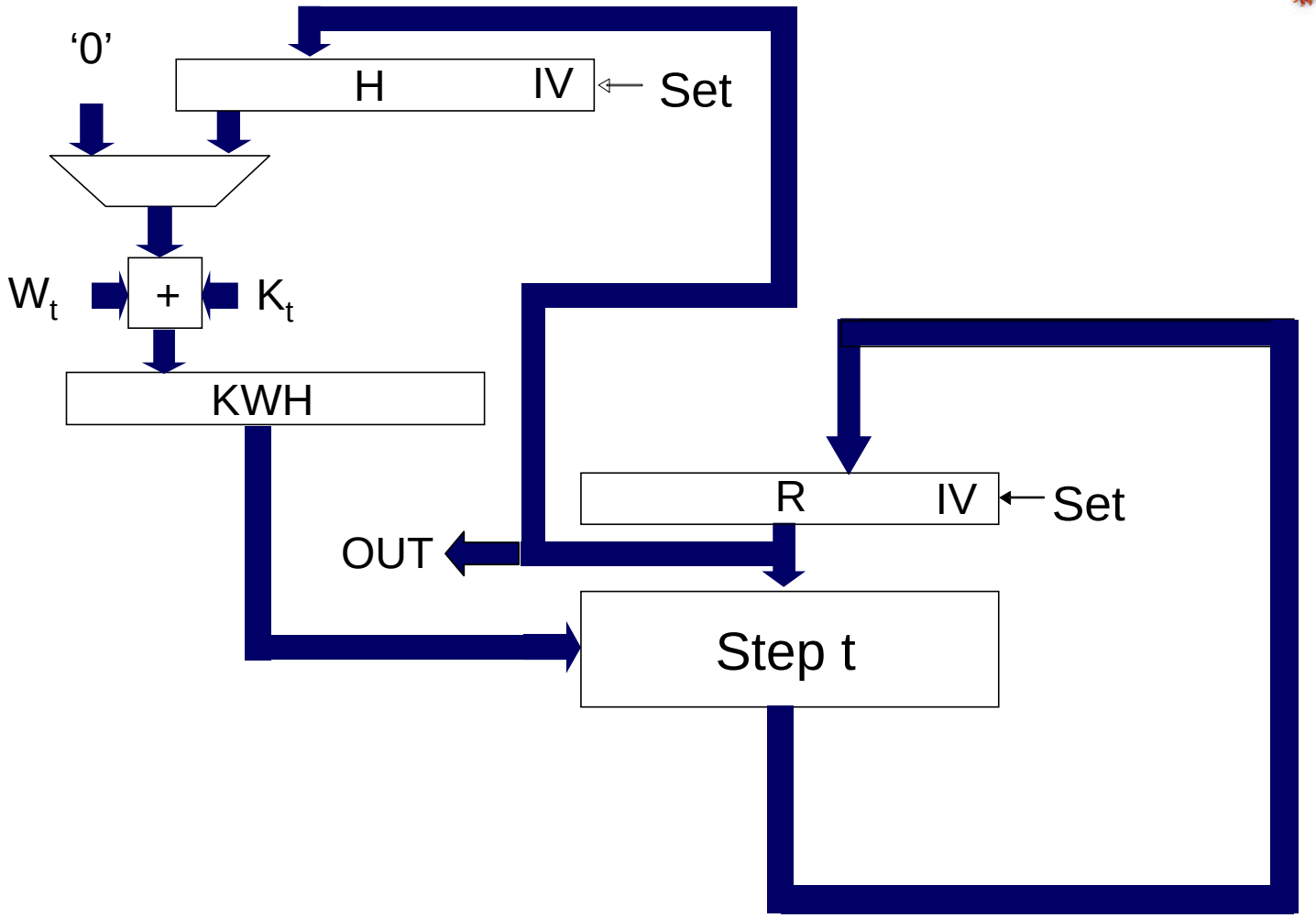
# Message digest unit of SHA-512
# Functional block diagram



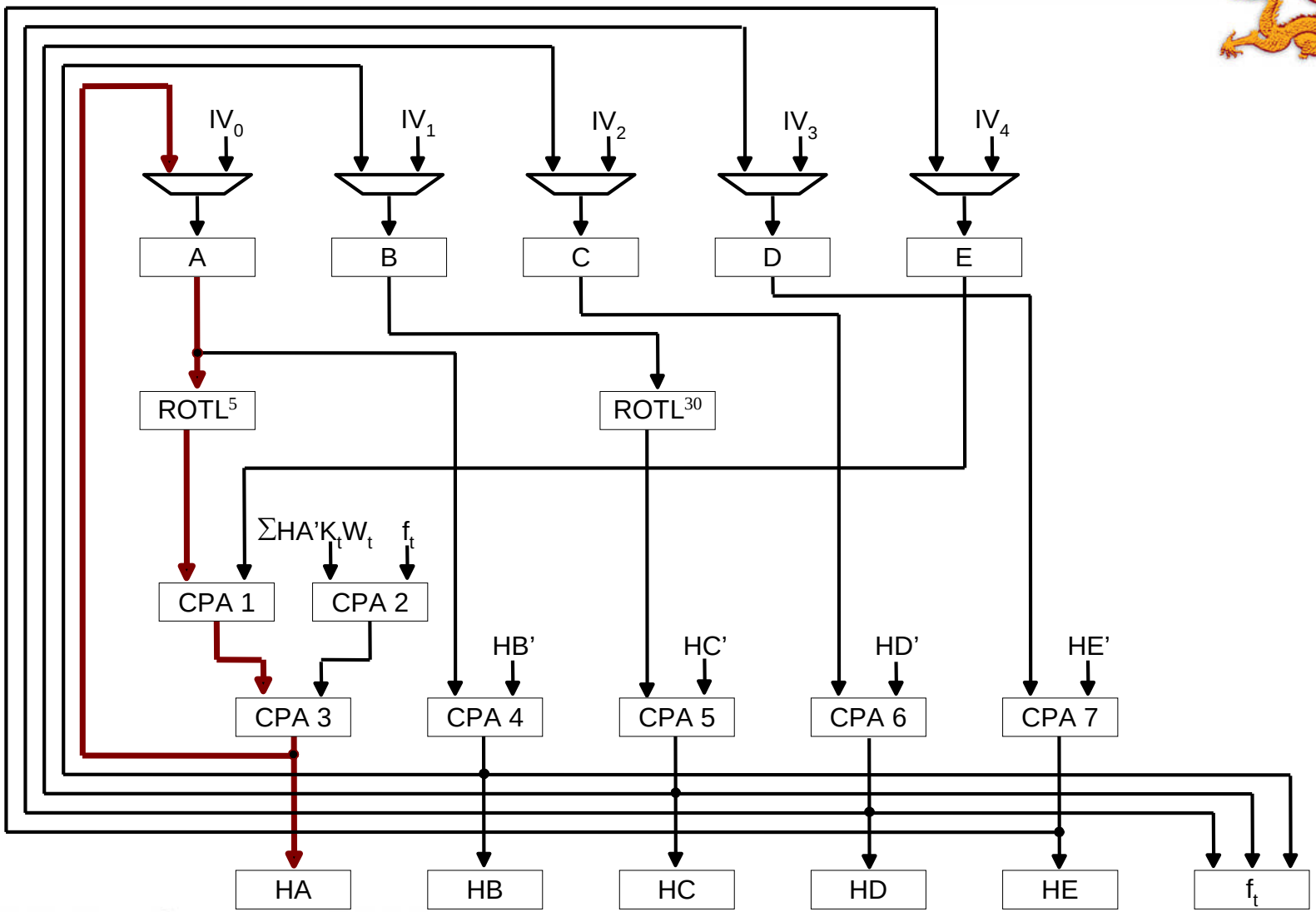
# Basic iterative architecture of a typical message digest



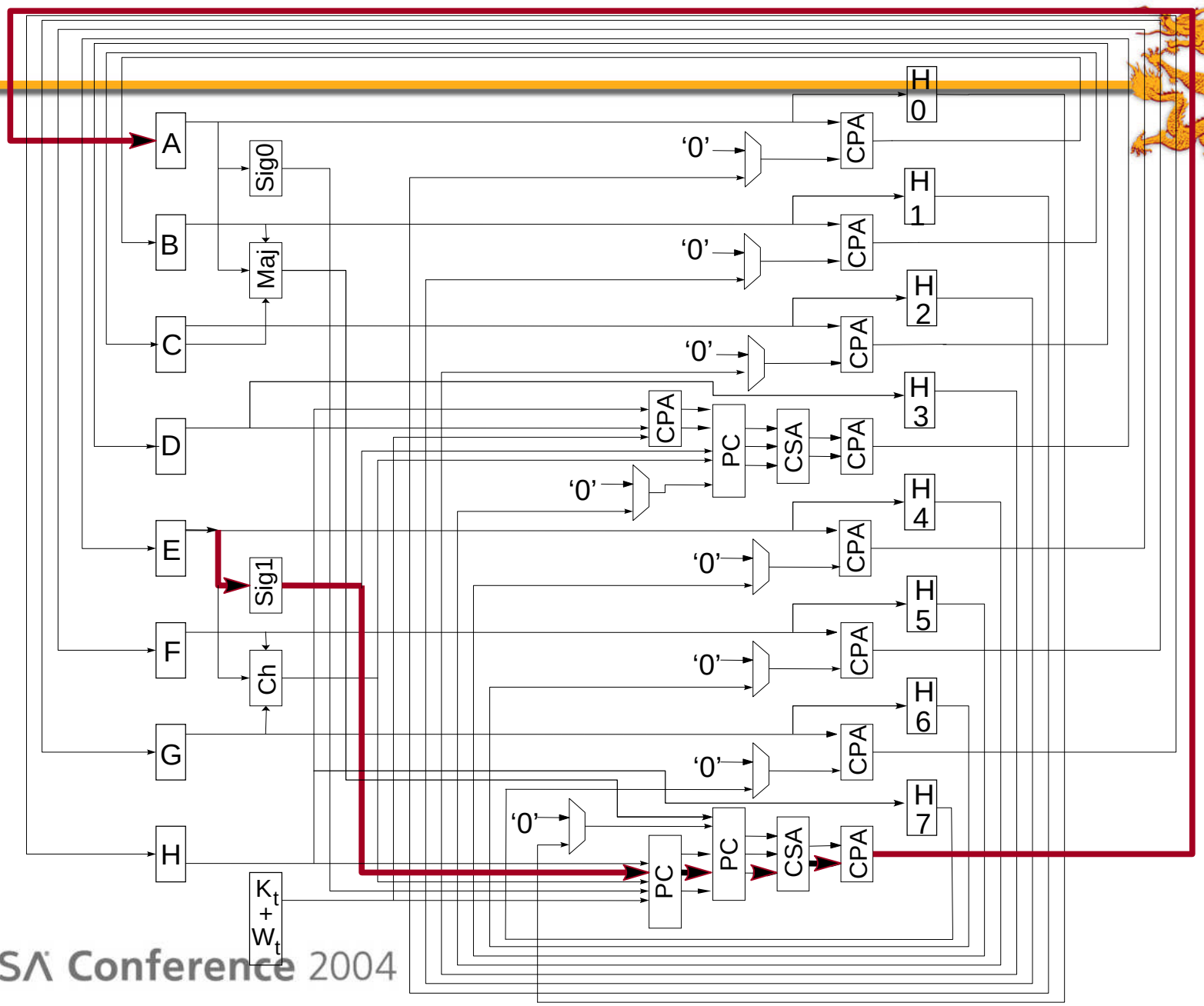
# Optimized version of the basic iterative design



# SHA-1 Basic Architecture
# Message digest

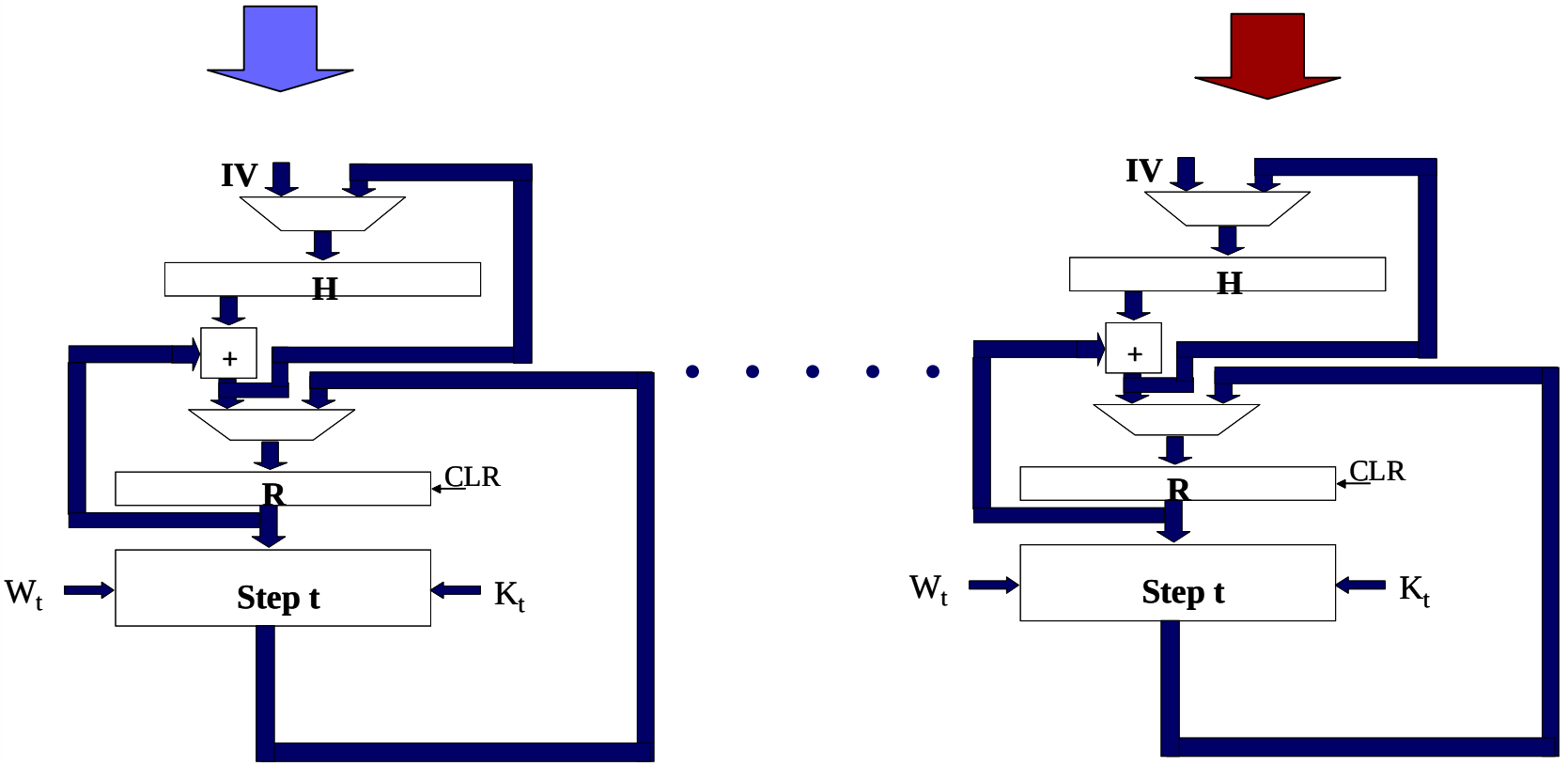


# Message digest unit of SHA-512



# Alternative architectures of dedicated hash functions

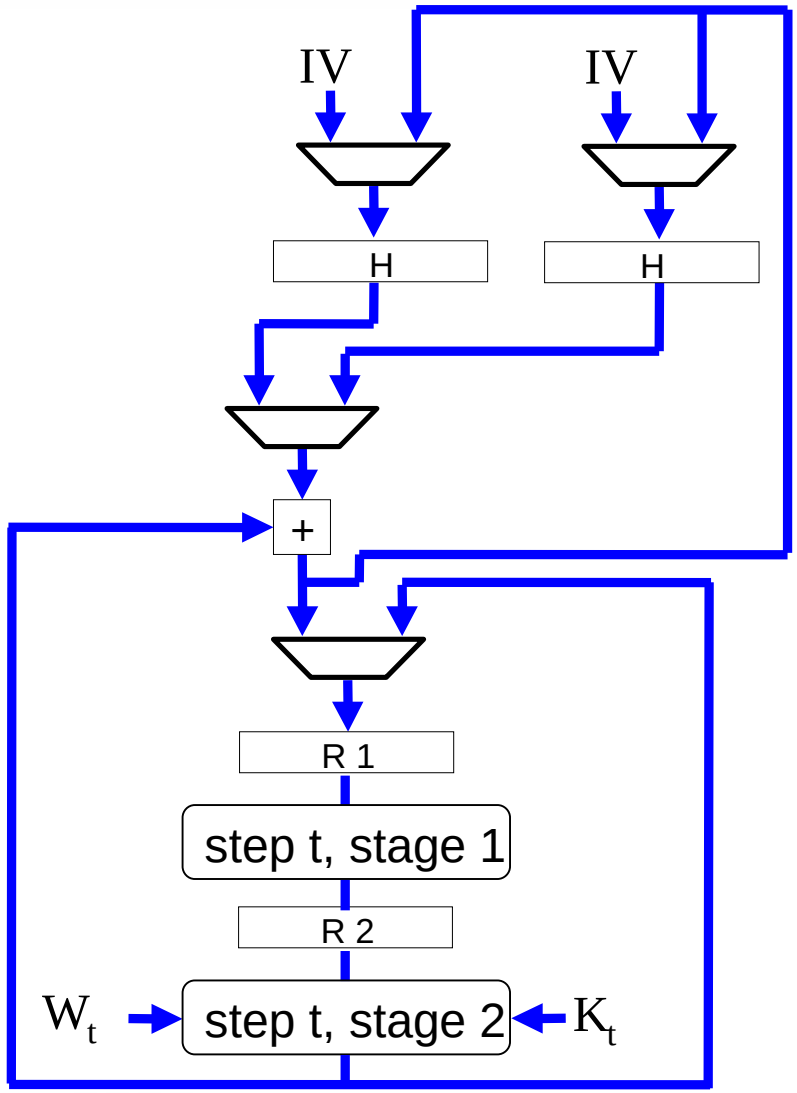
# Architecture with Multiple Processing Units



# Features of architecture with multiple processing units

- Pros
  - Throughput increases by a factor of k
- Cons
  - Latency the same as for basic architecture
  - Area increases by a factor of k
  - Requires k independent data streams (messages)

# Pipelined architecture

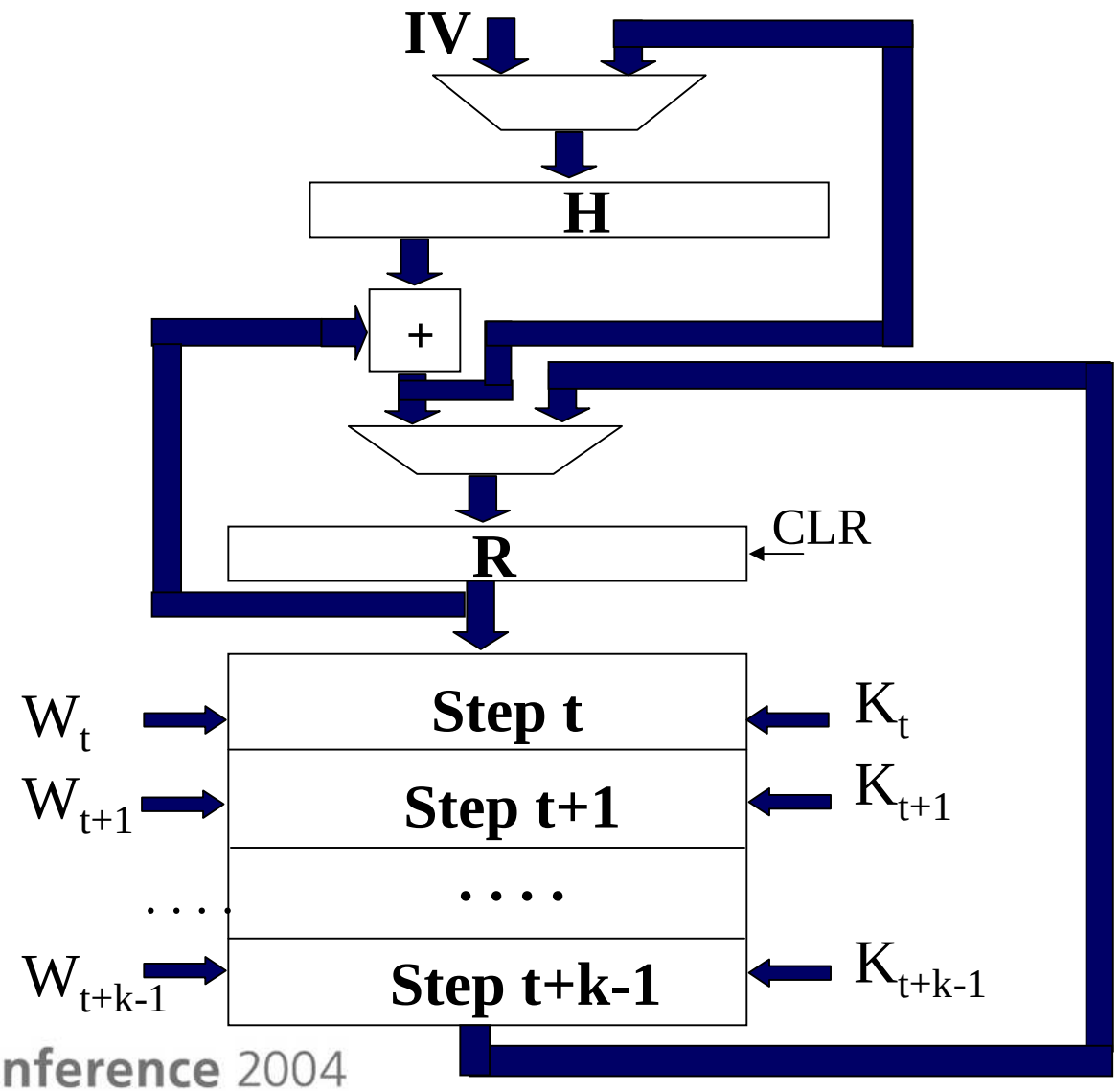


# Features of the pipelined architecture

- Pros
  - Throughput increases by a factor close to k
  - Area increases by a factor smaller than k
- Cons
  - Latency the same as for the basic architecture
  - Requires k independent data streams (messages)

# Our architecture

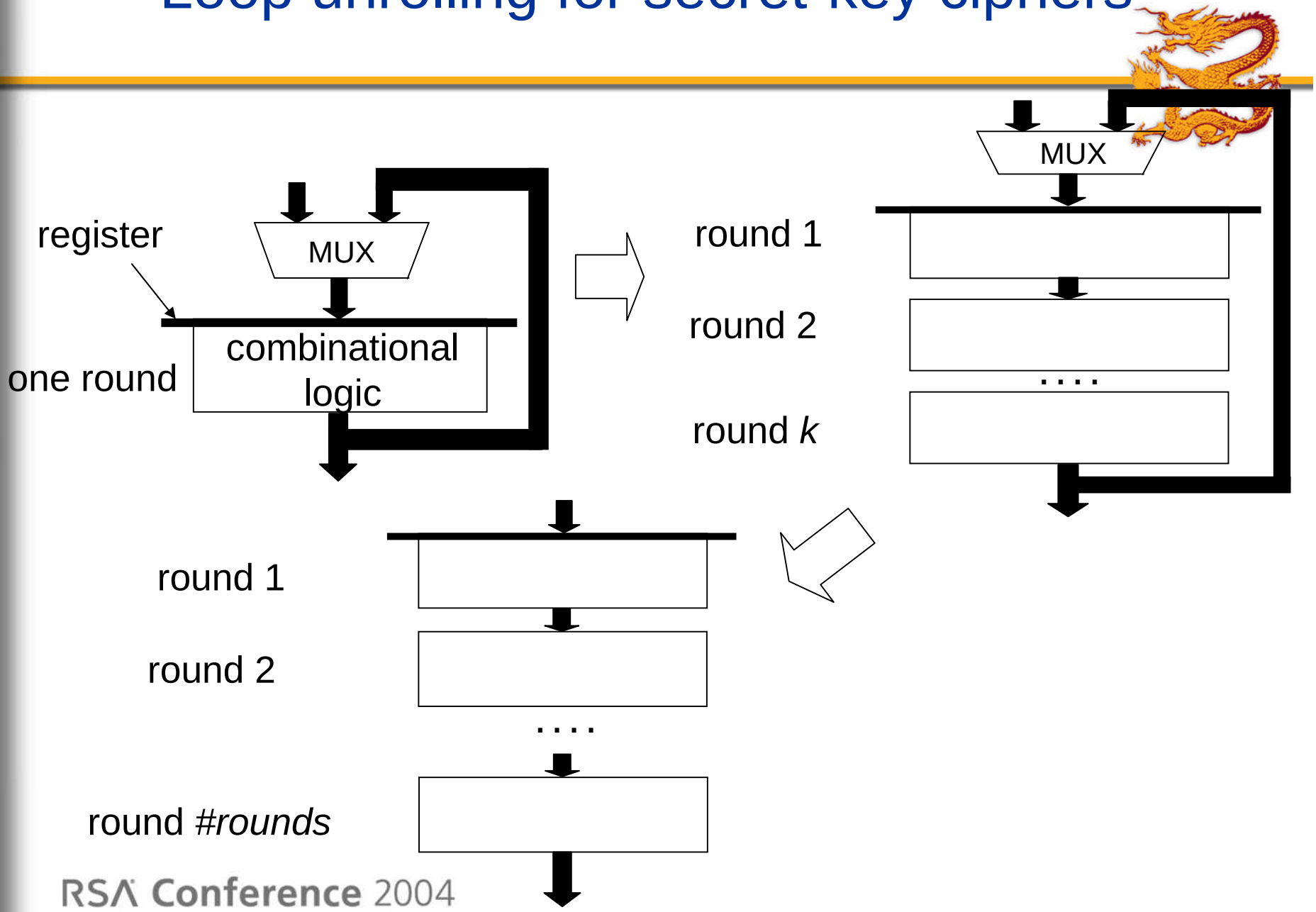
# Unrolled architecture



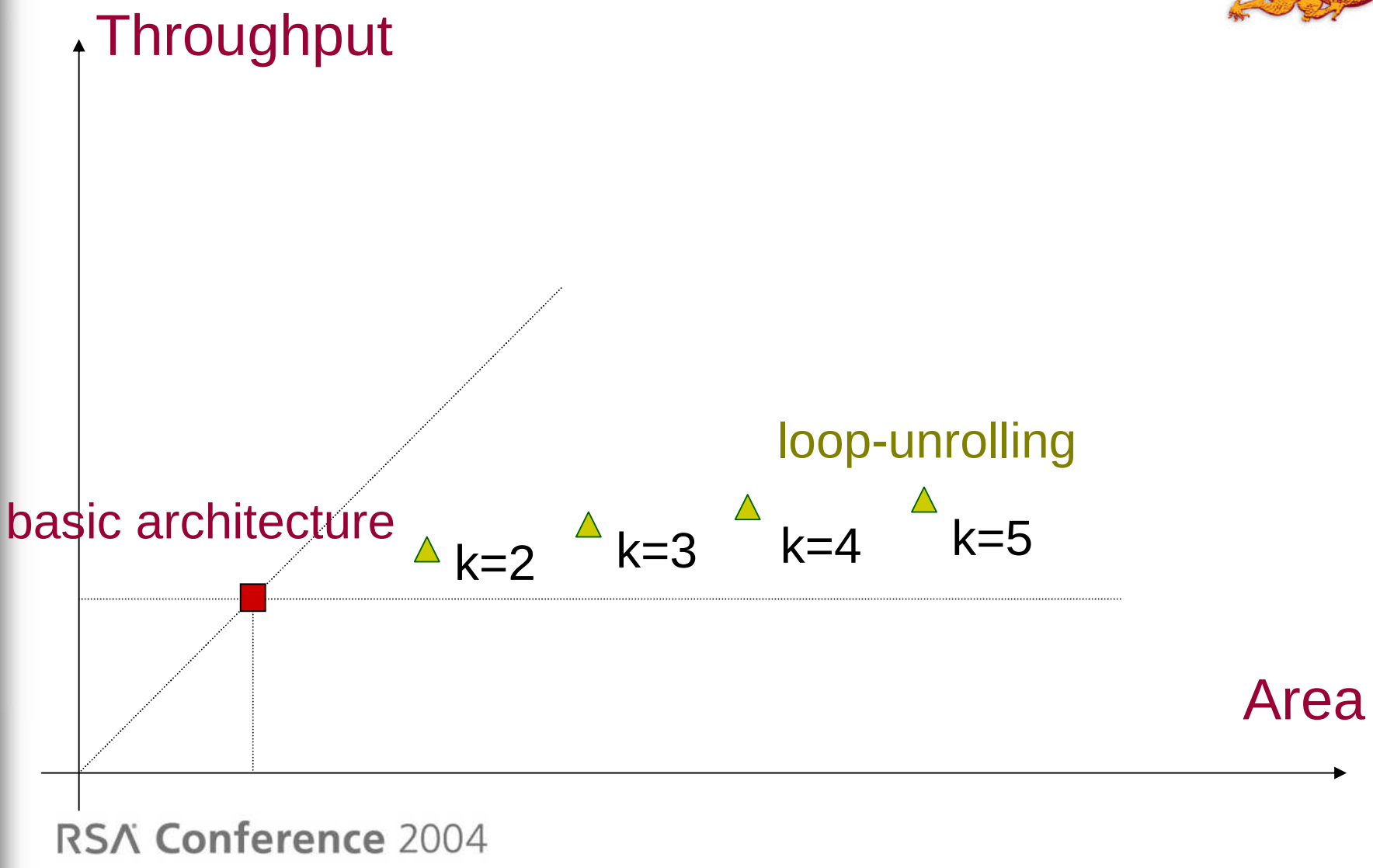
# Features of the unrolled architecture

- Pros
  - Reduces both latency and throughput
  - Requires only one data stream
- Cons
  - Area may increase substantially compared to the basic architecture

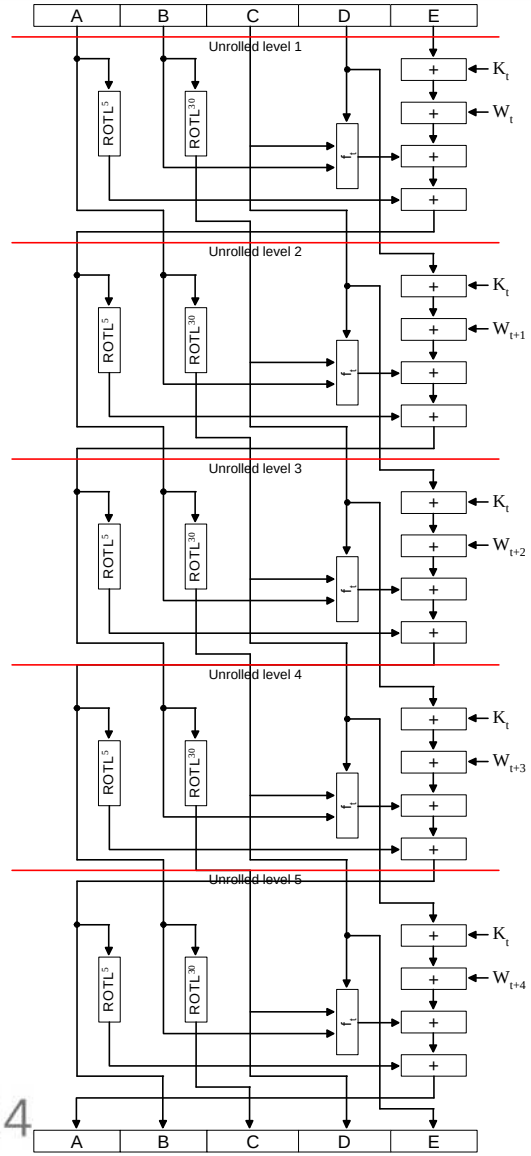
# Loop unrolling for secret-key ciphers



# Typical dependence between throughput and area for secret-key ciphers



# Partially unrolled message digest of SHA-1
# No optimization



# Derivation of critical path and data dependencies in the unrolled architecture of SHA-1

$$A_{t+1} = A_t\text{<<<}5 + f_t(B_t, C_t, D_t) + E_t + K_t + W_t =$$
$$= \mathbf{A_t\text{<<<}5 + f_t(B_t, C_t, D_t) + E_t + \sum K_t\ W_t}$$

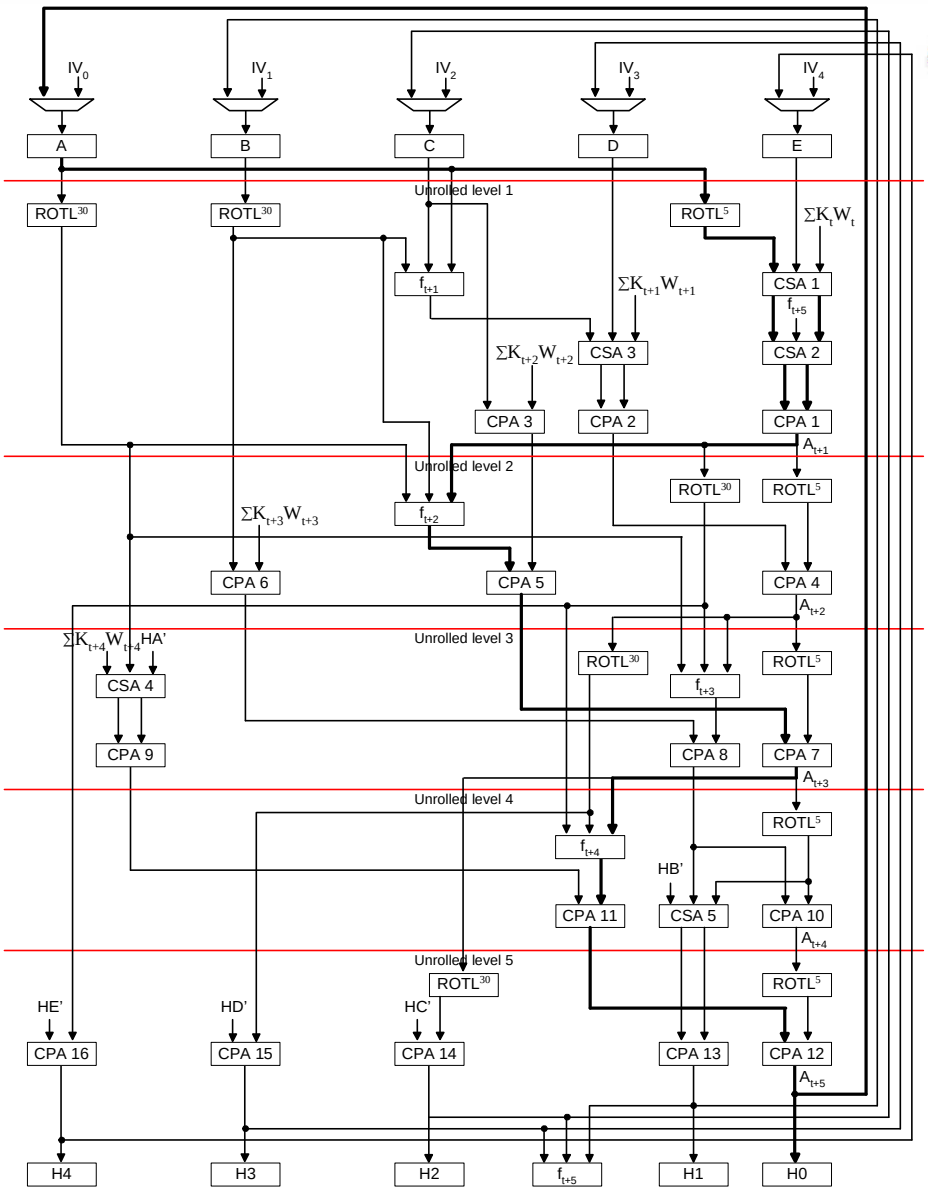
$$A_{t+2} = A_{t+1}\text{<<<}5 + f_{t+1}(B_{t+1}, C_{t+1}, D_{t+1}) + E_{t+1} + K_{t+1} + W_{t+1} =$$
$$= A_{t+1}\text{<<<}5 + [\ f_{t+1}(A_t, B_t\text{<<<}30, C_t) + D_t + \sum K_{t+1}\ W_{t+1}\ ]$$

$$A_{t+3} = \mathbf{A_{t+2}\text{<<<}5 + [\ f_{t+2}(A_{t+1}, A_t\text{<<<}30, B_t\text{<<<}30) + [C_t + \sum K_{t+2}\ W_{t+2}\ ]]}$$

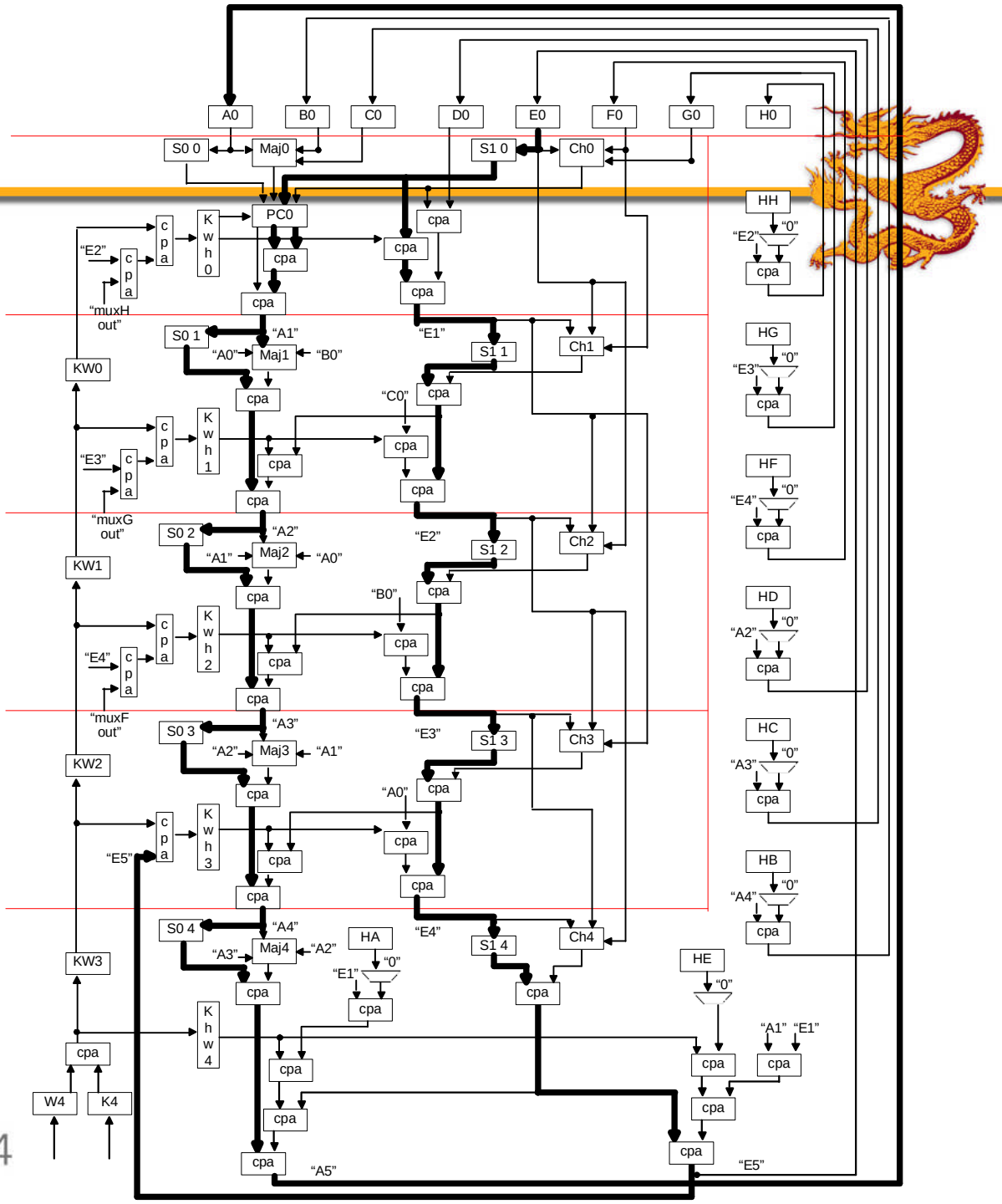
$$A_{t+4} = A_{t+3}\text{<<<}5 + [\ f_{t+3}(A_{t+2}, A_{t+1}\text{<<<}30, A_t\text{<<<}30) + [B_t\text{<<<}30 + \sum K_{t+3}\ W_{t+3}\ ]]$$

$$A_{t+5} = \mathbf{A_{t+4}\text{<<<}5 +}$$
$$\mathbf{[\ f_{t+4}(A_{t+3}, A_{t+2}\text{<<<}30, A_{t+1}\text{<<<}30) + [A_t\text{<<<}30 + \sum K_{t+4}\ W_{t+4} + HA'_{t+4}]]}$$

# Optimized unrolled architecture of SHA-1



# Optimized Unrolled Architecture of SHA-512 (k=5)



# Design & Testing Methodology

# Methodology and Tools

**Code in VHDL**

***Implementation***

***Verification***

**2. Synthesis, mapping, placing & routing**

*Xilinx, ISE*

**1. Functional simulation**

*Aldec, Active-HDL*

**Netlist with timing**

**3. Timing simulation**

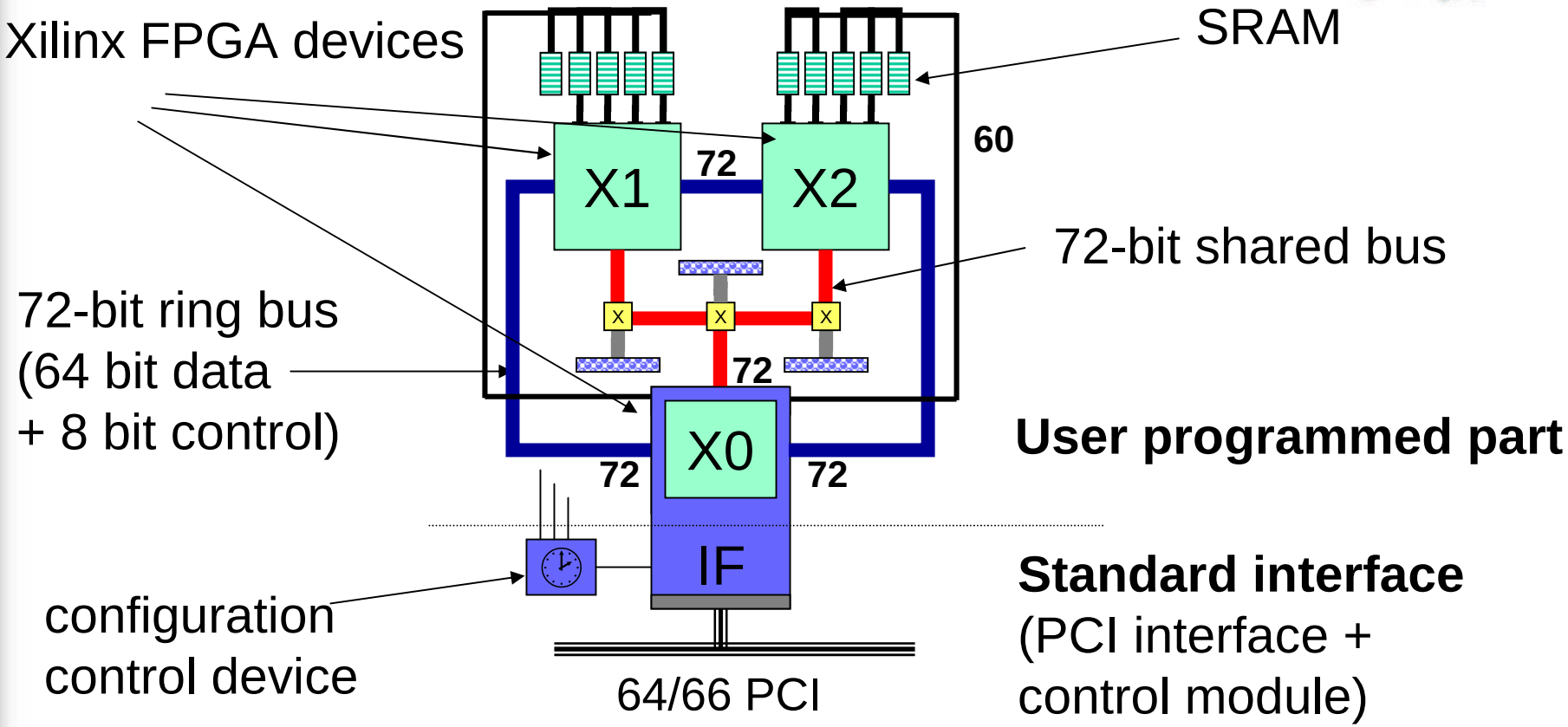
*Aldec, Active-HDL*

**Bitstream**

**4. Experimental Testing**

*USC-ISI, SLAAC-1V FPGA board*

# SLAAC-1V FPGA Board



# Testing Procedure

## 1. Functional testing

Digital Signature Standard Validation System (DSSVS) User's Guide

- Known Answer Tests
- Monte Carlo Test

## 2. Maximum clock frequency test

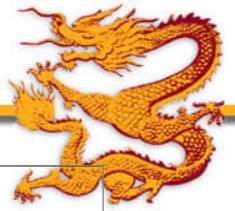
- clock frequency varied using binary search
- 30 x 3 MB of pseudorandom data hashed
- results compared with results from software implementation

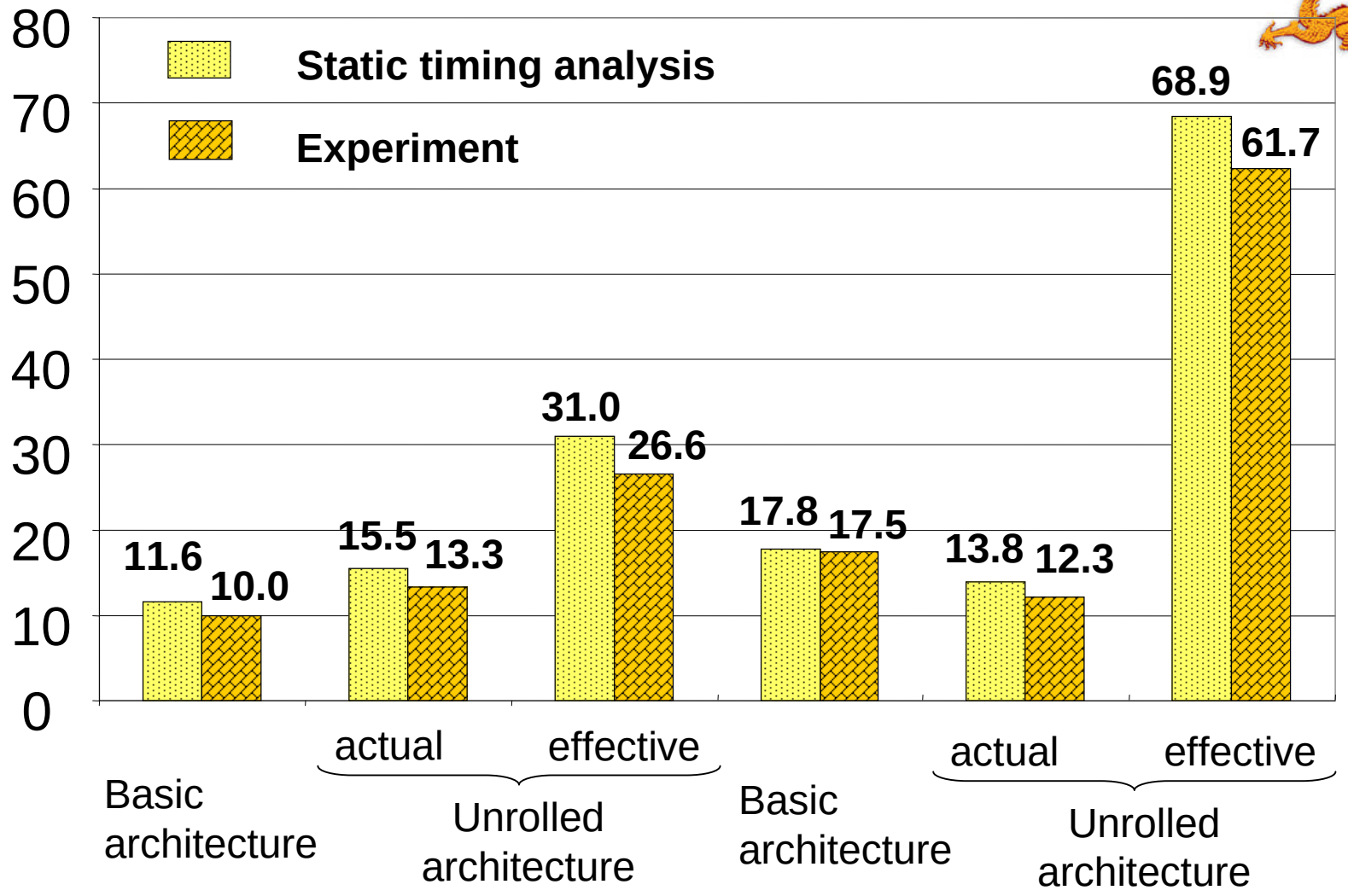
## 3. Maximum data throughput test

- maximum clock frequency
- 3 MB of pseudorandom data hashed
- time necessary to complete all operations determined

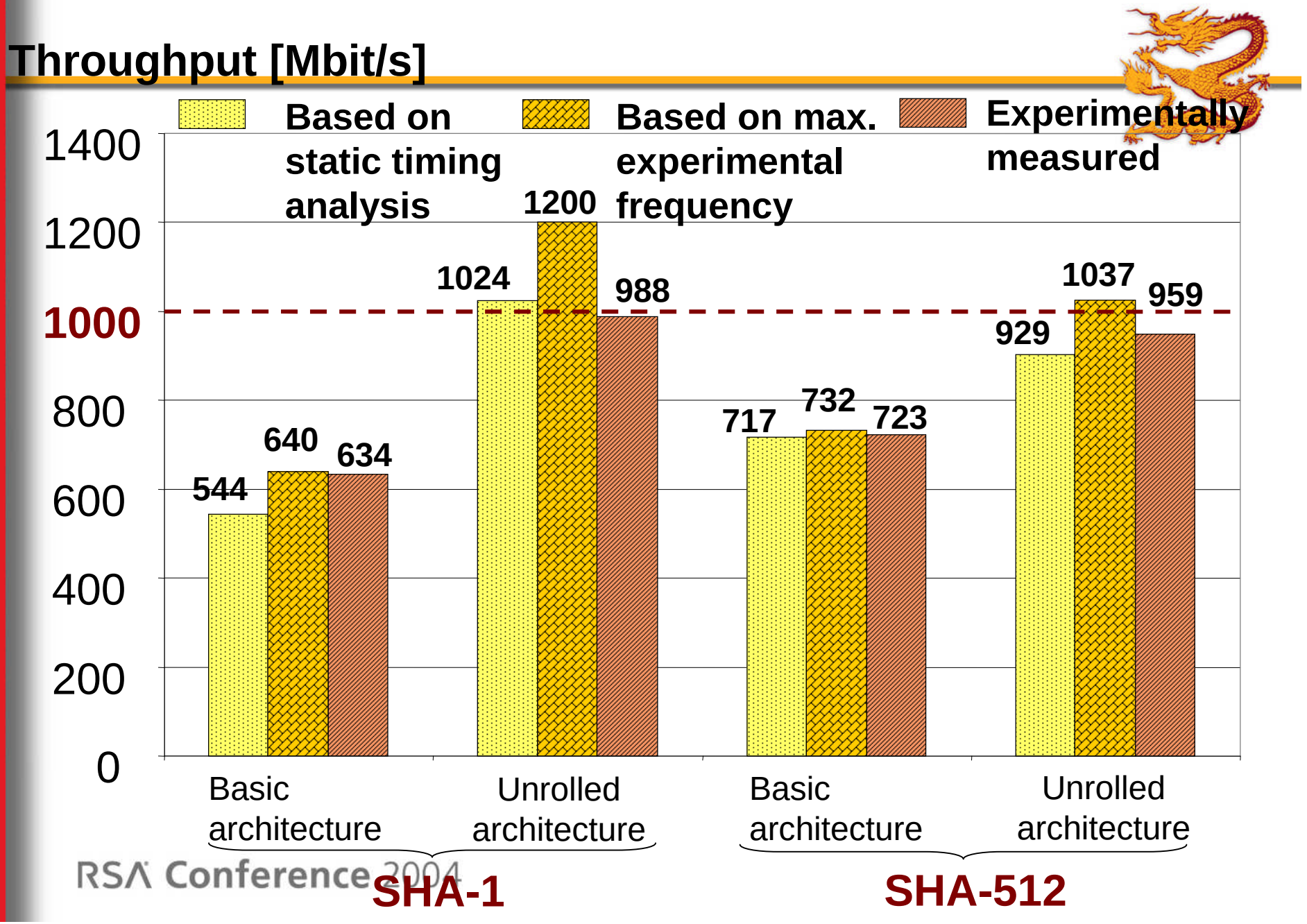
# Results

# Clock periods



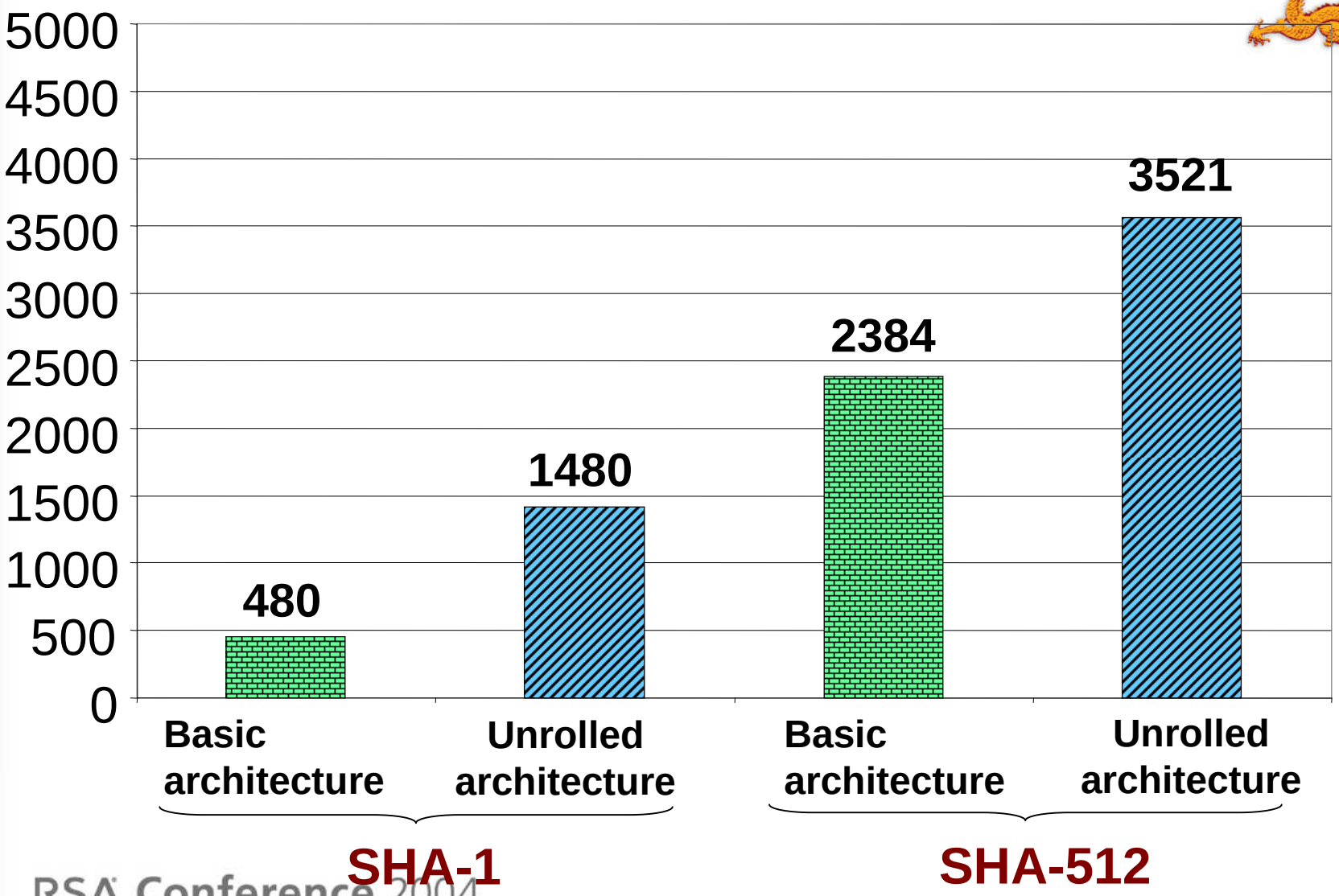


# Throughputs



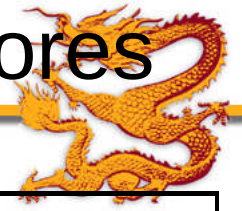
# Resource requirements

**Area [# CLB Slices]**



# Comparison with commercial FPGA cores

# Comparison of our designs for SHA-1 with the representative commercial IP cores

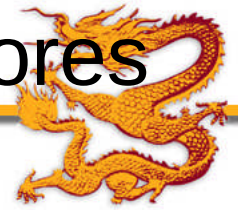


| Source | Clock frequency [MHz] | Throughput [Mbit/s] | Area [CLB Slices] |
|---|---|---|---|
| **Xilinx Virtex** | | | |
| **Our, basic** | **85** | **544** | **480** |
| **Our, unrolled (k=5)** | **64**[1] | **1024** | **1480** |
| ALMA Technologies | 70 | 442 | 686 |
| Helion Technology Ltd. | 76 | 480 | 689 |
| Ocean Logic Pty Ltd | 56 | 352 | 612 |
| **Xilinx Virtex-E** | | | |
| **Our, basic** | **103** | **659** | **484** |
| **Our, unrolled (k=5)** | **72.5** | **1160** | **1484** |
| ALMA Technologies | 87 | 549 | 686 |
| Bisquare Systems Private Limited | 66 | 422 | 579 |
| Helion Technology Ltd. | 95 | 600 | 689 |
| Intron, Ltd. | 71 | 449 | 716 |
| Ocean Logic Pty Ltd | 71.5 | 452 | 612 |



[1] – multicycle path used

# Comparison of our designs for SHA-512 with the representative commercial IP cores



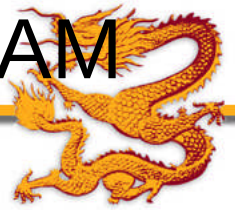
| Source | Clock frequency [MHz] | Throughput [Mbit/s] | Area [3] [CLB Slices] |
|---|---|---|---|
| **Xilinx Virtex** | | | |
| **Our, basic** | **56** | **717** | **2384 Slices** |
| **Our, unrolled (k=5)** | **67**[2] | **929** | **3521 Slices** |
| ALMA Technologies | 56 | 707 | 2690 Slices |
| **Xilinx Virtex-E** | | | |
| **Our, unrolled (k=5)** | **72**[2] | **1034** | **3517 Slices** |
| ALMA Technologies | 68 | 859 | 2690 Slices |

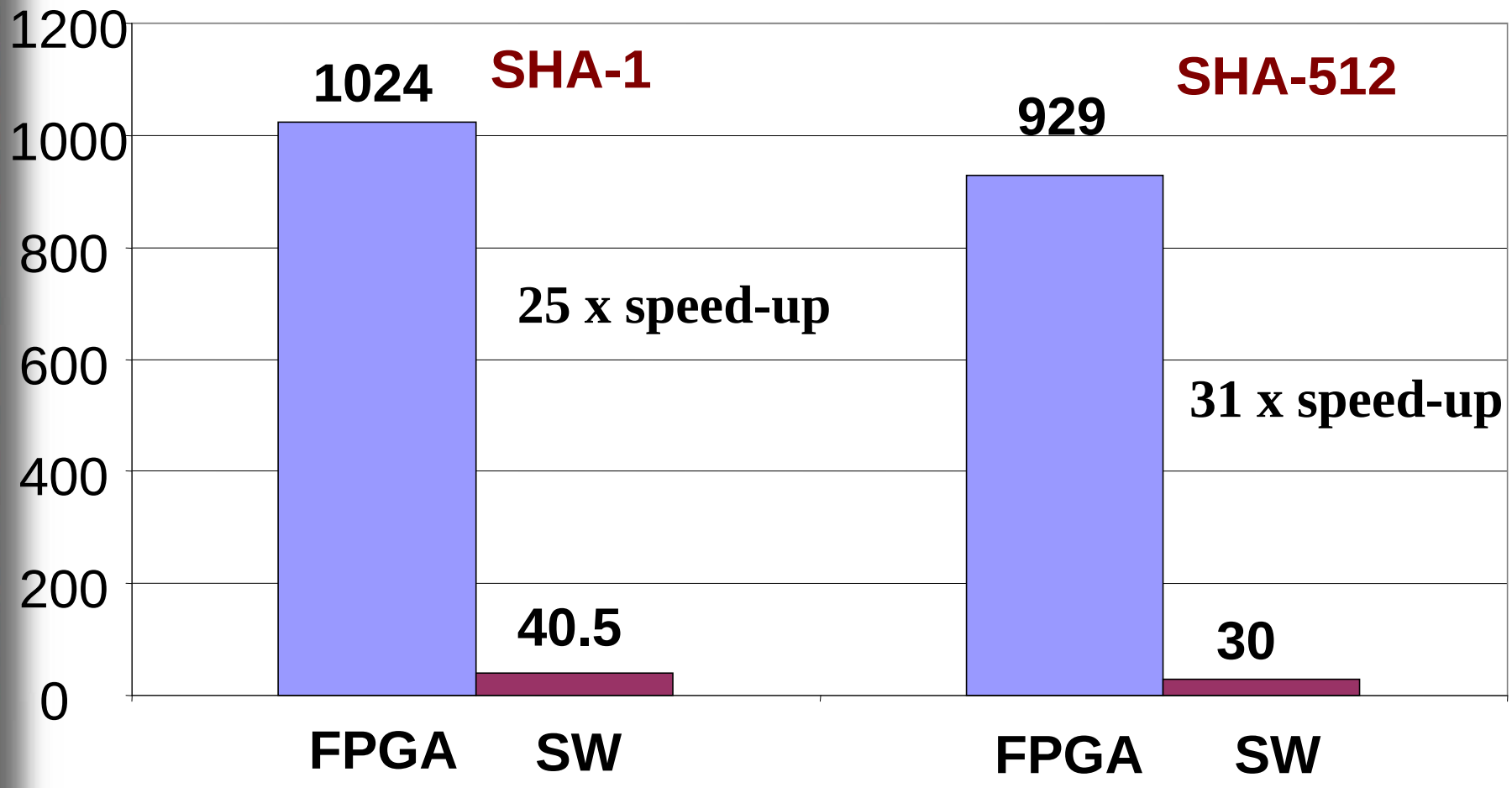
[2] – multicycle path used

[3] – each circuit contains additionally 4 Block RAMs

# Comparison with software

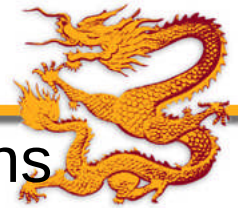
# Speed-up vs. software
# Crypto++, 2.2 GHz Pentium 4, 1 GB RAM





# Summary

# Summary (1)



A **new partially unrolled architecture** of hash functions proposed, and implemented for SHA-1 and SHA-512

The only architecture known to substantially increase the throughput for a **single stream of data**

The only architecture allowing to sustain the throughput in excess of **1 Gbit/s for medium-size** families of **FPGA** devices

Our implementation of **SHA-1** outperformed all reported commercial IP cores by **at least a factor of 2**

Our implementation of **SHA-512** outperformed all reported commercial IP cores by **at least 31%**

# Summary (2)

Loop unrolling **more suitable for hash algorithms** than for symmetric-key ciphers

**Speed up** compared to the basic iterative architecture:

| | |
|---|---|
| **SHA-1:** | **1.9** |
| **SHA-512:** | **1.3** |

Speed up is a strong **function of data dependencies** present in the algorithm