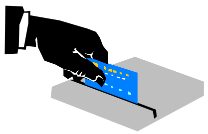
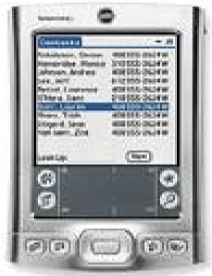
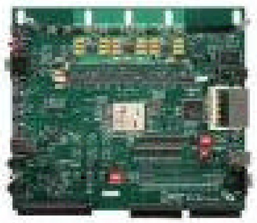
# Improving SHA-2 Hardware Implementations







**Ricardo Chaves**, Georgi Kuzmanov,
Leonel Sousa, and Stamatis Vassiliadis

ricardo.chaves@inesc-id.pt

- Introduction to Cryptography
  - History
- SHA2 implementation
  - Usage
  - Algorithm & optimizations
  - Proposed implementation
  - Results
  - Conclusions

## Cryptography - History and Usage

1900 BC is the 1st kwon usage of cryptography by the Egyptians – simple substitution scheme

Hieroglyphic enciphermemts of proper names and titles, with cipher hieroglyphs at left, plain equivalents at right

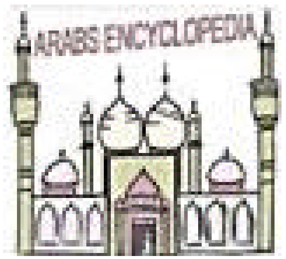
1 AD Julius Cesar used the simple letter shift cipher in the Gallic Wars

Msg: *OMNIA GALLIA EST DIVISA IN PARTES TRES*
Enc: RPQLD JDOOLD HVW GLYLVD LQ SDUWHV WUHV

300 AD a new mathematical cryptographic scheme was used by Sun Tzu

$$CRT\ (2,3,2)_{(357)}=??\ (23)$$

1412 AD an 14vol encyclopedia on Cryptanalysis is compiled by the Arabs




In the 2nd world war the enigma rotor machine is used (substitution scheme using a continuously changing alphabet)

## Cryptography - History and Usage

In the 70's with the development of complex electronic systems much more complex cryptographic systems have been created, such as Lucifer and DES.

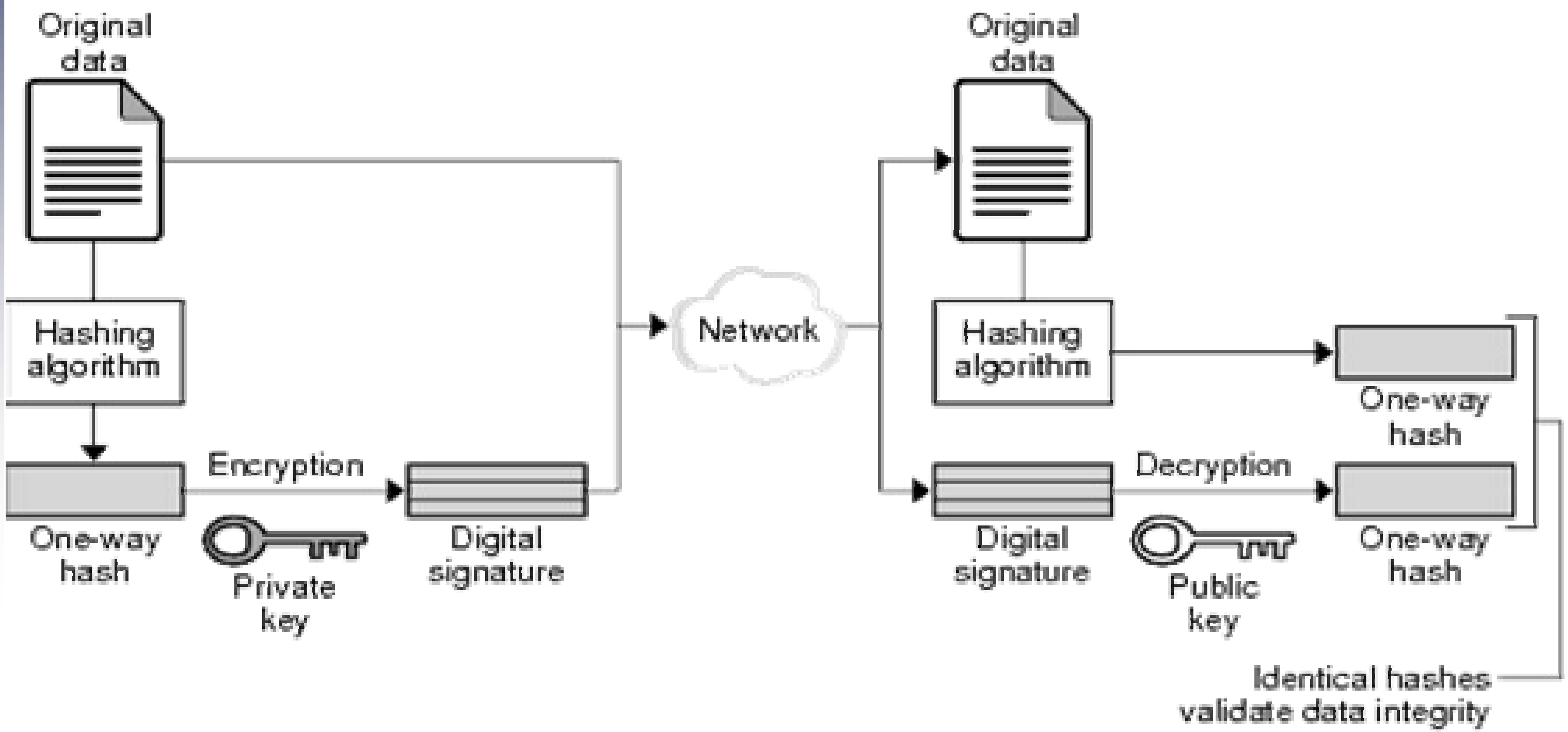
nowadays a variety of cryptographic algorithms exist an they are present in almost every day actions:

- Accessing the internet
- E-shopping
- ATM machines
- Emails
- Buildings access
- Pay TV
- Anti-car theft systems
- Private communications
- …

## Hash functions usage and characteristics

- Digital Signatures
- Data verification

- One-way operation
- *Can not* find a different input message that generates the same hash value



## Tempering with the message

| Message received: | Digest Message: |
| --- | --- |
| Transfer ¥*1* to account *1234* | **975A234B51** |
| Transfer ¥ **1000** to account 1234 | 8E65730FFC |
| Transfer ¥ 1 to account 123**5** | B78D65D8C0 |

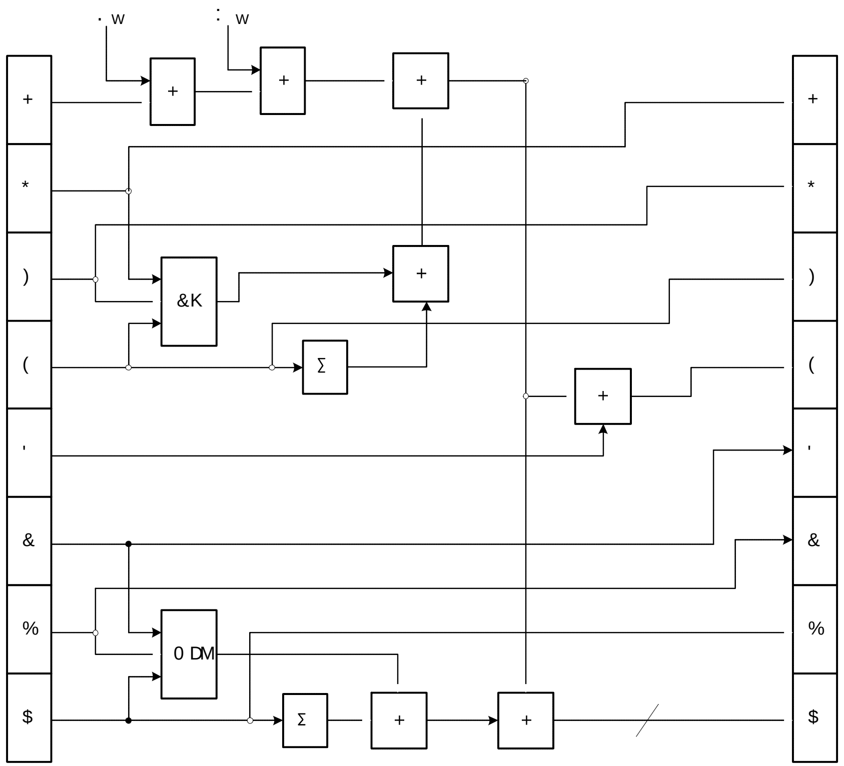
Hash given by the Digital Signature :

***975A234B51***

## SHA2 Hash Function

Characteristics:

- Based on simple logical and arithmetic operations



Additional operations:

- The value $W_t$ is computed from the input data (data block expansion).

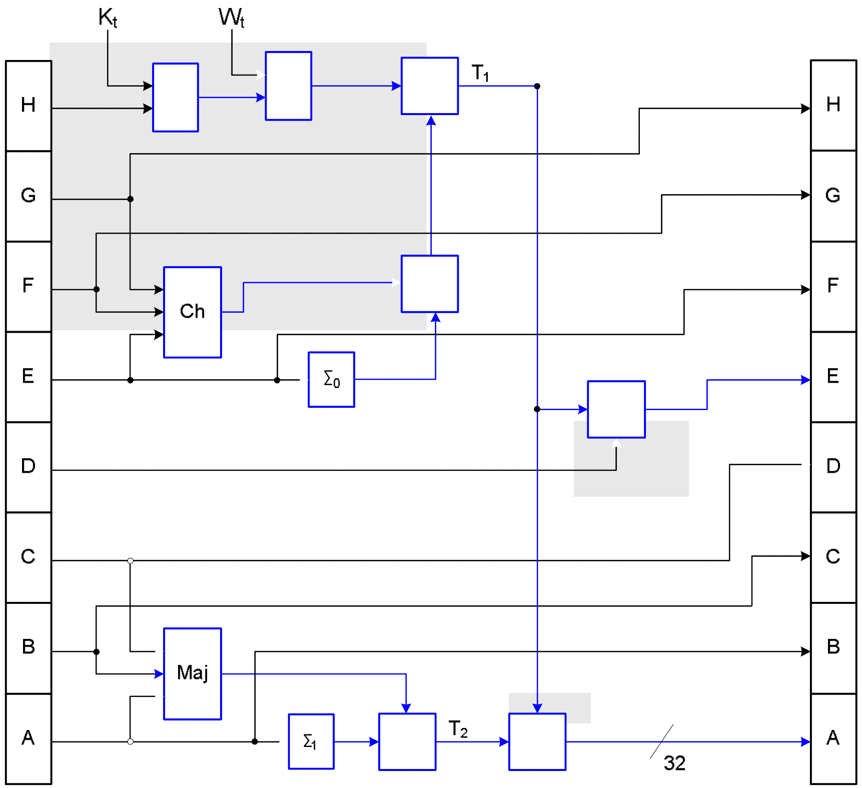
- After the 64 round (for SHA256) the resulting value has to be added to the intermediate Digest Message.

SHA512 requires 80 rounds using 64 bit values

## SHA2 Hash Function

### Characteristics:

- Based on simple logical and arithmetic operations
- Only the values A and E requires computation




### Optimization:

- Only the A and E computation depends on values computed in the previous round

- Values $H_t$,$G_t$,$F_t$ and $D_t$,$C_t$,$B_t$ do not depend on the values of round *t*.

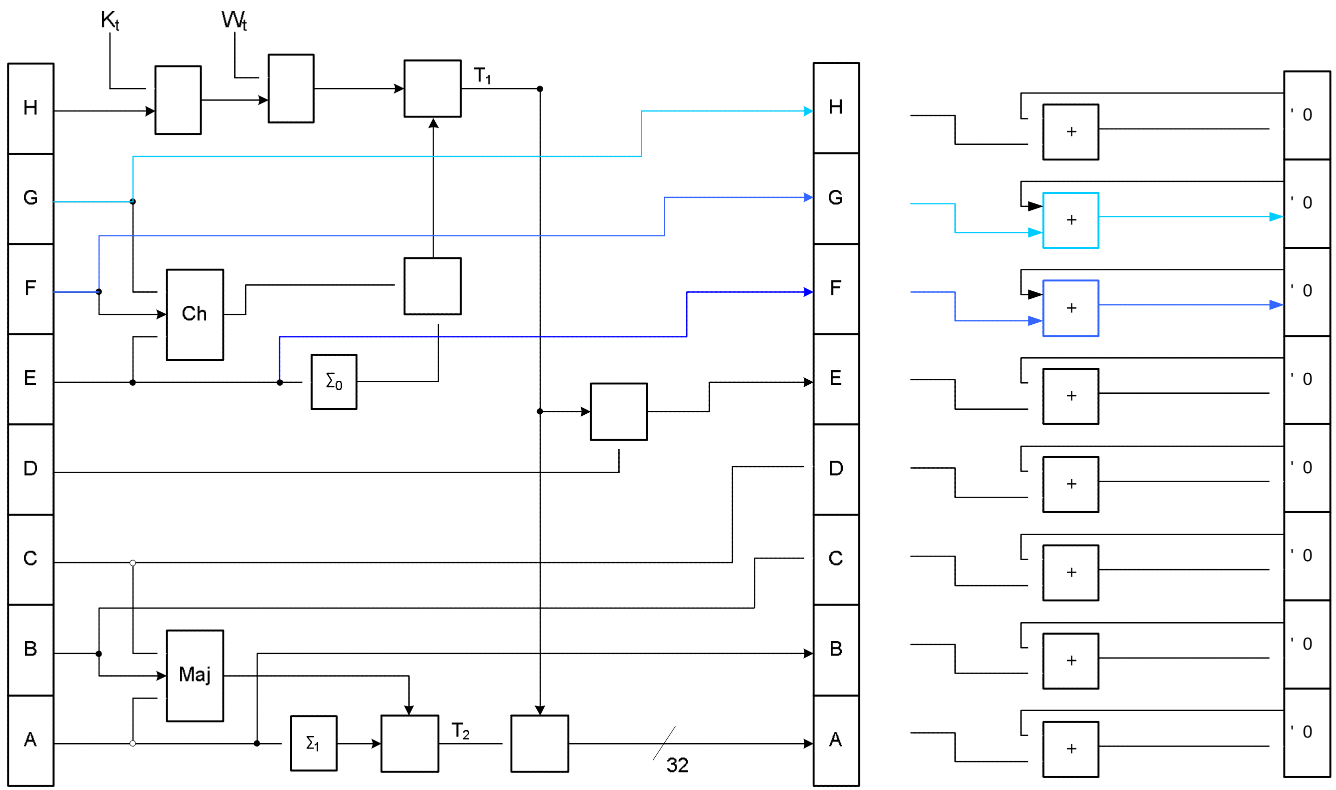
- Part of the round *t* computation can be performed in round *t-1*

# SHA2 Hash Function

## Characteristics:

- Only the values A and E requires computation
- After the rounds the values have to be added to the Digest Message

- $H_t = G_{t-1} = F_{t-2}$ ; $D_t = C_{t-1} = B_{t-2}$
- DM is known in the 1st round



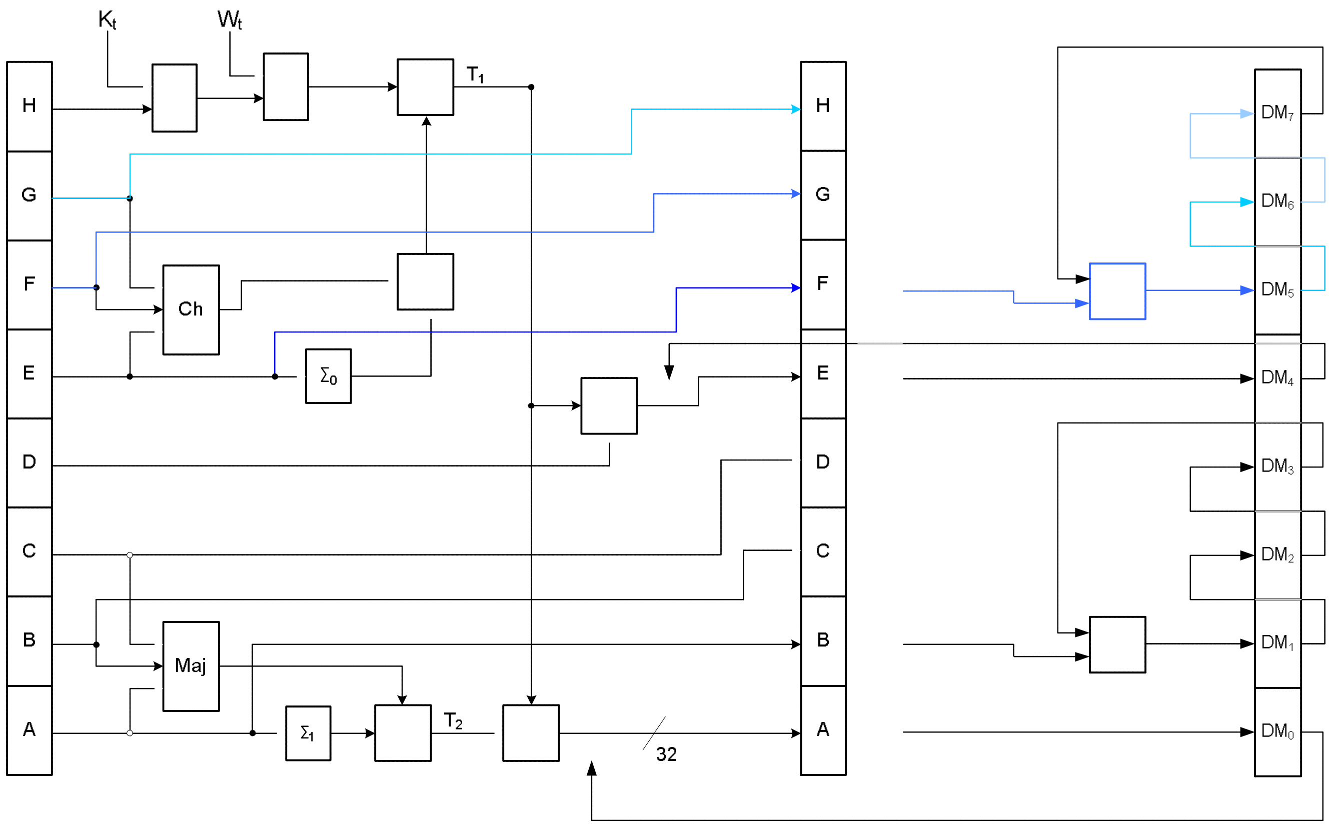
## SHA2 Hash Function

Characteristics:

- Only the values A and E requires computation
- After the rounds the values have to be added to the Digest Message

Optimization:

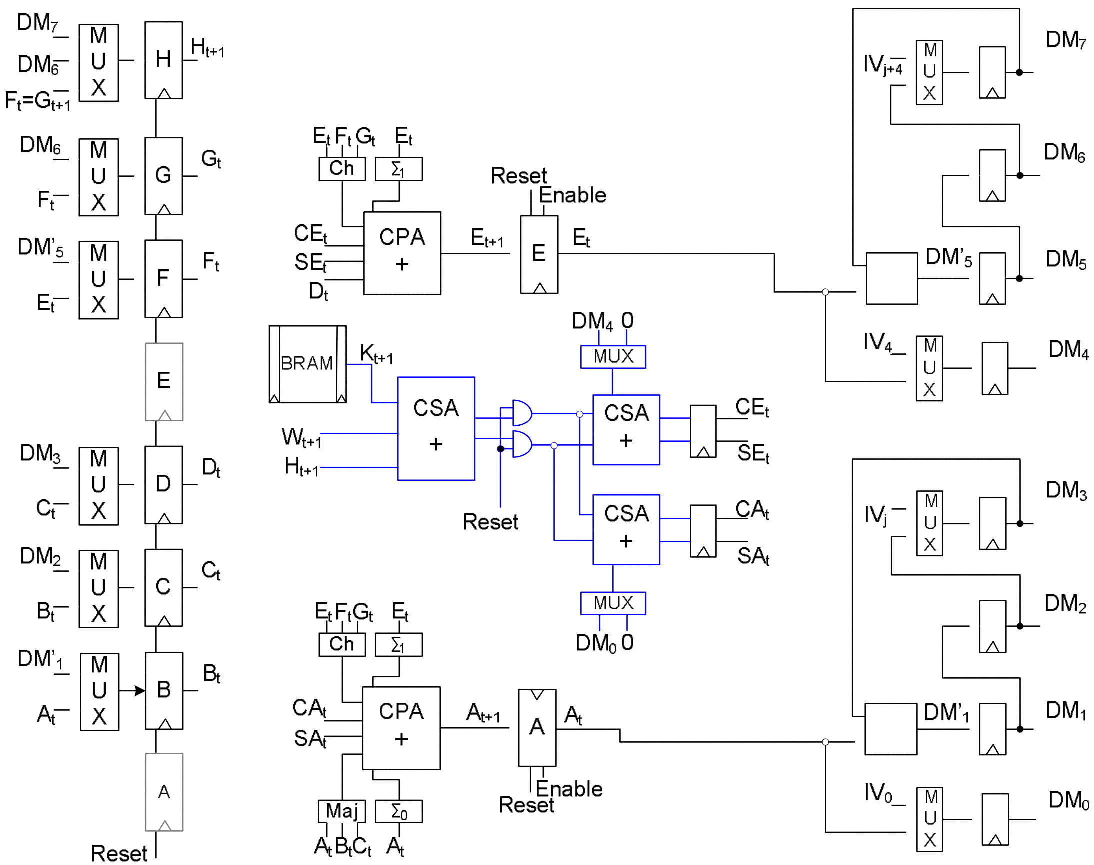
- $H_t = G_{t-1} = F_{t-2}$ ; $D_t = C_{t-1} = B_{t-2}$
- DM is known in the 1st round



## SHA2 hardware implementation

# SHA2 core:

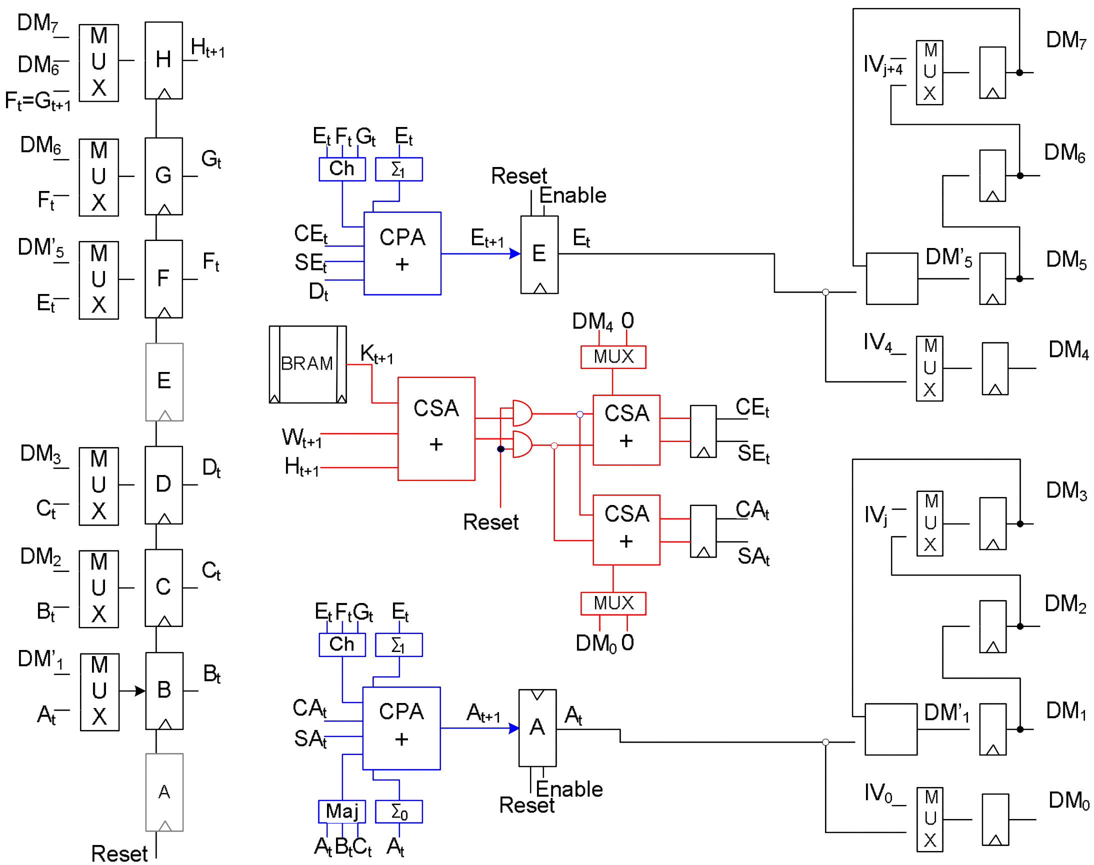
- Variable IV initializations
- The pipeline has to be filled



## SHA2 hardware implementation

### SHA2 core:

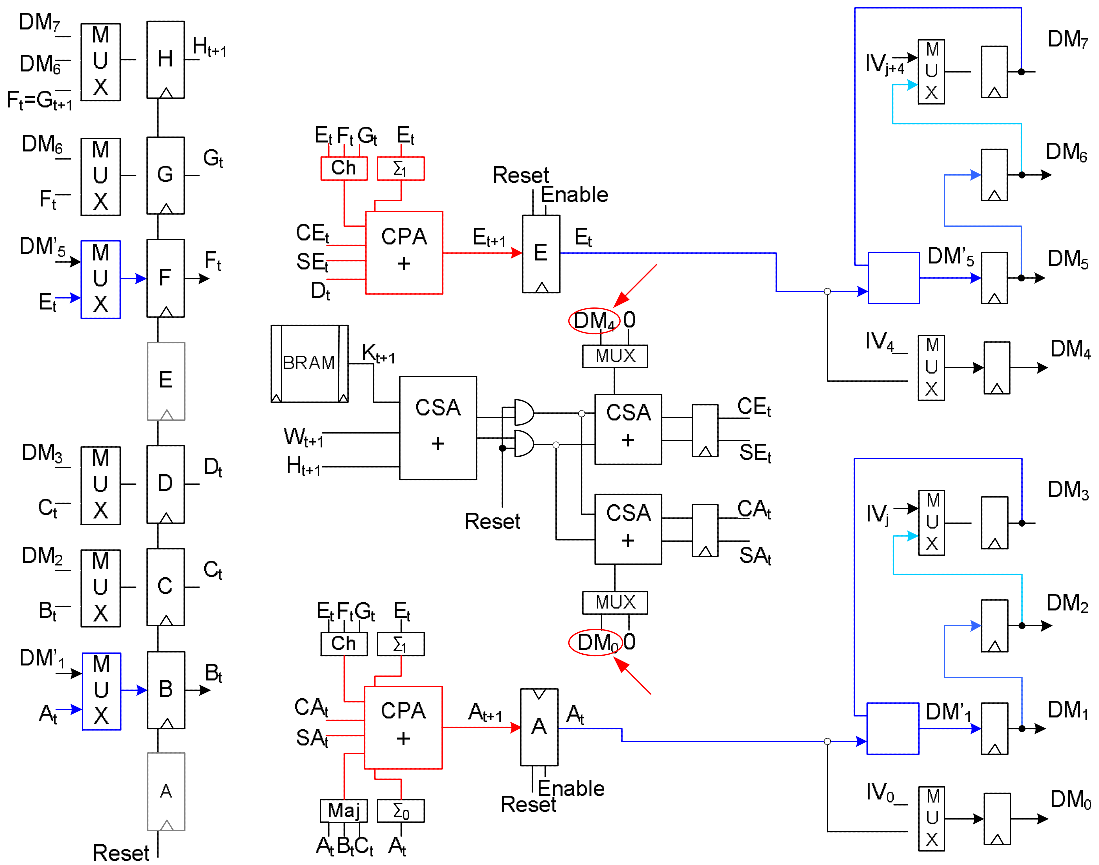
- Round *t* is being calculated in blue while round *t*-1 is being calculated in red
- Critical path ~ 6 input adder



## SHA2 hardware implementation

# SHA2 core:

- The DM is added to the calculated values – e.g. $DM_5 = DM_5 + F_{t+1} = DM_5 + E_t$
- $DM_0$ and $DM_4$ are calculated in the round hardware



# Improving SHA-2 Hardware Implementations

## Results for the standalone SHA2 core

SHA256 core:

- Variable **IV** initializations allowed
- ***1.4 Gbits*** throughput
- Only ***5%*** utilization of the available logic (Virtex II Pro 30)

Efficiency improved:

- ***50%*** improvement of the Throughput/Slice metric regarding commercial cores
- **100%** improvement of the Throughput/Slice metric regarding academia cores

| **SHA256** | Sklav | Our | McEv. | Our | Helion | Our |
|---|---|---|---|---|---|---|
| Device | XCV | XCV | XC2V | XC2V | XC2PV-7 | XC2PV-7 |
| Slices | 1060 | 764 | 1373 | 797 | 815 | **755** |
| BRAMs | = 1 | 1 | =1 | 1 | 1 | 1 |
| Frequency | 83 | 82 | 133 | 150 | 126 | 174 |
| Throughput | 326 | 646 | 1009 | 1184 | 977 | **1370** |
| Throughput/Slice | 0.31 | 0.84 | 0.74 | 1.49 | 1.2 | 1.83 |
| Improved | **171%** | | **101%** | | **53%** | |

# Improving SHA-2 Hardware Implementations

## Results for the standalone SHA2 core

SHA512 core:

- Variable **IV** initializations allowed
- ***1.8 Gbits*** throughput
- Only ***13%*** utilization of the available logic (Virtex II Pro 30)

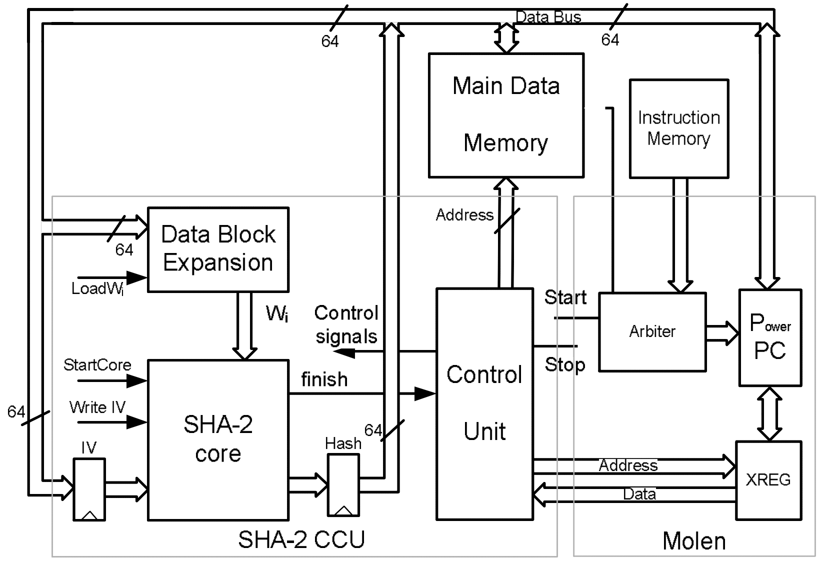
Efficiency improved:

- ***77%*** improvement of the Throughput/Slice metric.

| **SHA512** | Sklav | Lien | Our | McEv | Our | Our |
|---|---|---|---|---|---|---|
| Device | XCV | XCV | XCV | XC2V | XC2V | XC2VP-7 |
| Slices | 2237 | 2384 | 1680 | 2726 | 1666 | **1667** |
| BRAMs | n.a. | n.a. | 2 | = 1 | 1 | 1 |
| Frequency | 75 | 56 | 70 | 109 | 121 | 141 |
| Throughput | 480 | 717 | 889 | 1329 | 1534 | **1780** |
| Throughput/Slice | 0.21 | 0.31 | 0.53 | 0.49 | 0.92 | 1.01 |
| Improved | **165**% | **77**% | | **88**% | | |

## MOLEN with an SHA2 CCU

Main characteristics:




Low FPGA utilization.

- *7 %* of a XCV2P30 for the SHA256
- *13 %* of a XCV2P30 for the SHA512

Throughput of **785 *Mbits/s*** @100MHz for SHA256

Throughput of **1200 *Mbits/s*** @100MHz for SHA512

Minimal software integration costs

- A large range of encryption applications can be speedup just be being recompiled for the MOLEN processor with the SHA2 core.

Prototype @:

TU Delft

# Improving SHA-2 Hardware Implementations

## SHA2 core in the MOLEN processor

Minimal Software integration costs:

*Original Software*:

Declaration: void SHA256(){ instructions…}

Usage : **SHA256(Data,size, IV);**

*Modified for MOLEN*:

Declaration: **#pragma call_fpga encrypt**

void SHA256(){ /*implemented in Hardware*/}

Usage : **SHA256(Data,size, IV);**

Significant Speedup for a minimal area cost:

- **7 % occupation** of a Virtex II Pro 30 FPGA (994 Slices)
- Throughput of 785Mbits (in MOLEN @100MHz) instead of 5 Mbits (in Software @300MHz), **153**x Speedup.

## Conclusions – SHA2

Efficiency gains to existing state-of-the-art

Higher Throughput/Slice ratio

- **50 %** when compared with known SHA256 commercial cores
- **100%** when compared with academia SHA256 related art
- **77 %** when compared with academia SHA512 related art

Throughput of 1.37 Gbits for SHA-256 (@ 174 MHz) **5%** Occupation (533 Slices)
Throughput of 1.78 Gbits for SHA-512 (@ 141 MHz) **13%** Occupation (1667 Slices)

MOLEN implementation

Low FPGA utilization: **7%-13%** (Virtex II Pro 30)

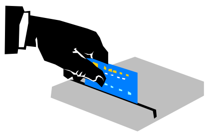
High throughput: **785** (SHA256) – **1200 Mbits/s** (SHA512) (@ 100 MHz)

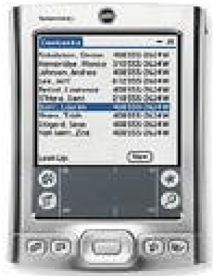
SHA256 throughput speedup of **153x**

Minimal software integration costs

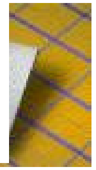
* Prototype running @100MHz

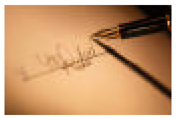
## Improving SHA-2 Hardware Implementations

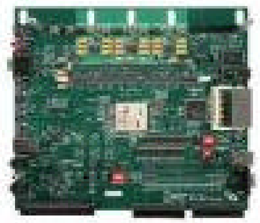




# The End







Ricardo Chaves, et al.

ricardo.chaves@inesc-id.pt