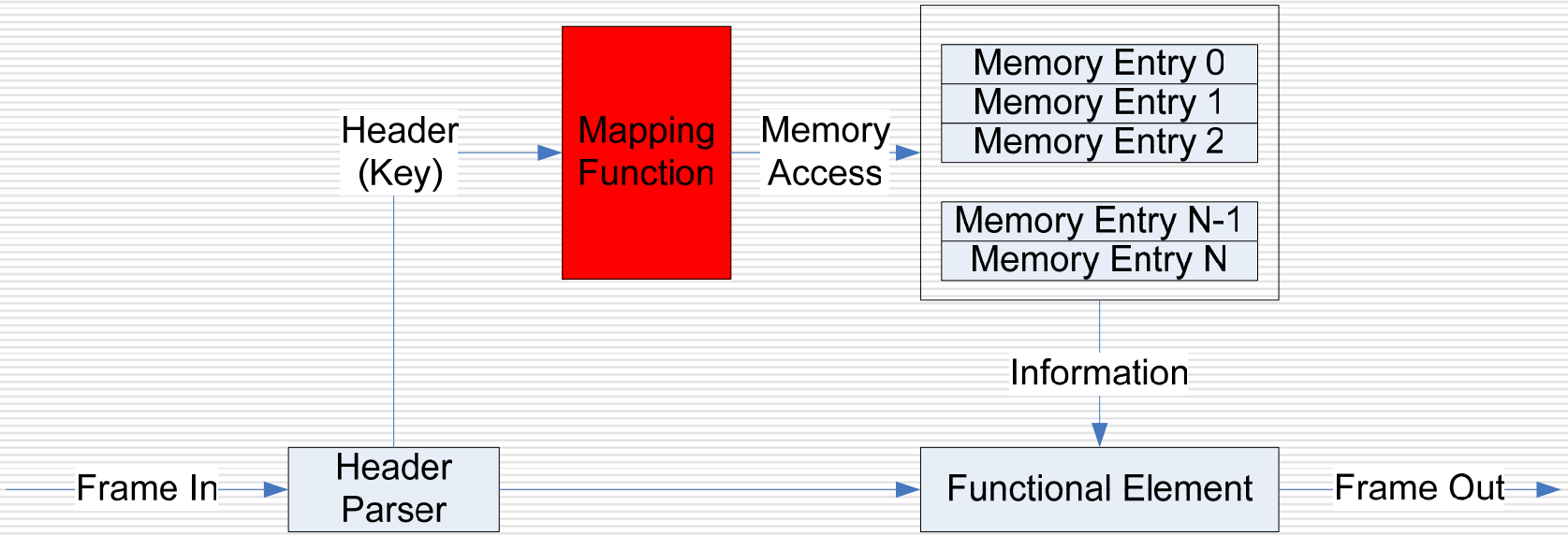
Packet Classification with Evolvable Hardware Hash Functions

Harald Widiger, Ralf Salomon, Dirk Timmermann
University of Rostock
Institute of Applied Microelectronics
and Computer Engineering

Outline

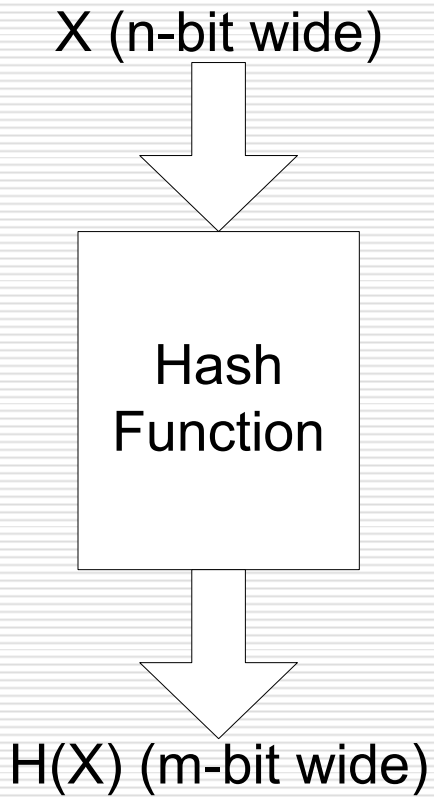
- Classification Problem
- Basic Principles of Hash Functions and evolutionary Algorithms
- Hardware-Architecture of evolvable Hash Functions
- System Architecture of a Packet Classifier
- Conclusion and Outlook

Classification Problem



- Find the information corresponding to a key in a memory as quick as possible

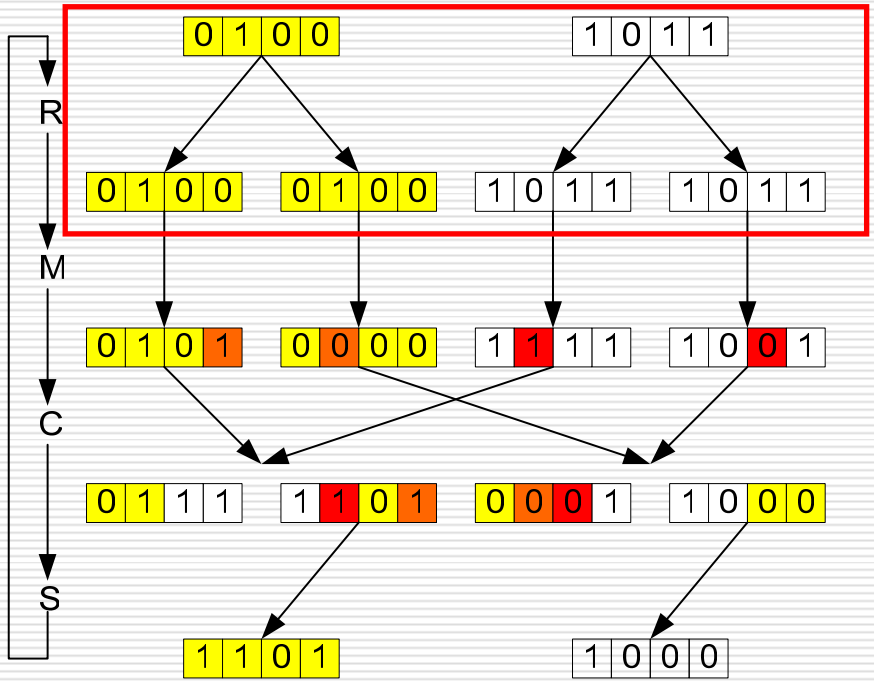
Hash Functions



- Collision: $X \neq Y$; $H(X) = H(Y)$
- Resolution
  - Rehashing H(H(X));
  - Linear H(X)+Prime
- Time Complexity: O(1)
- Memory Space : O(N)

Evolutionary Algorithms

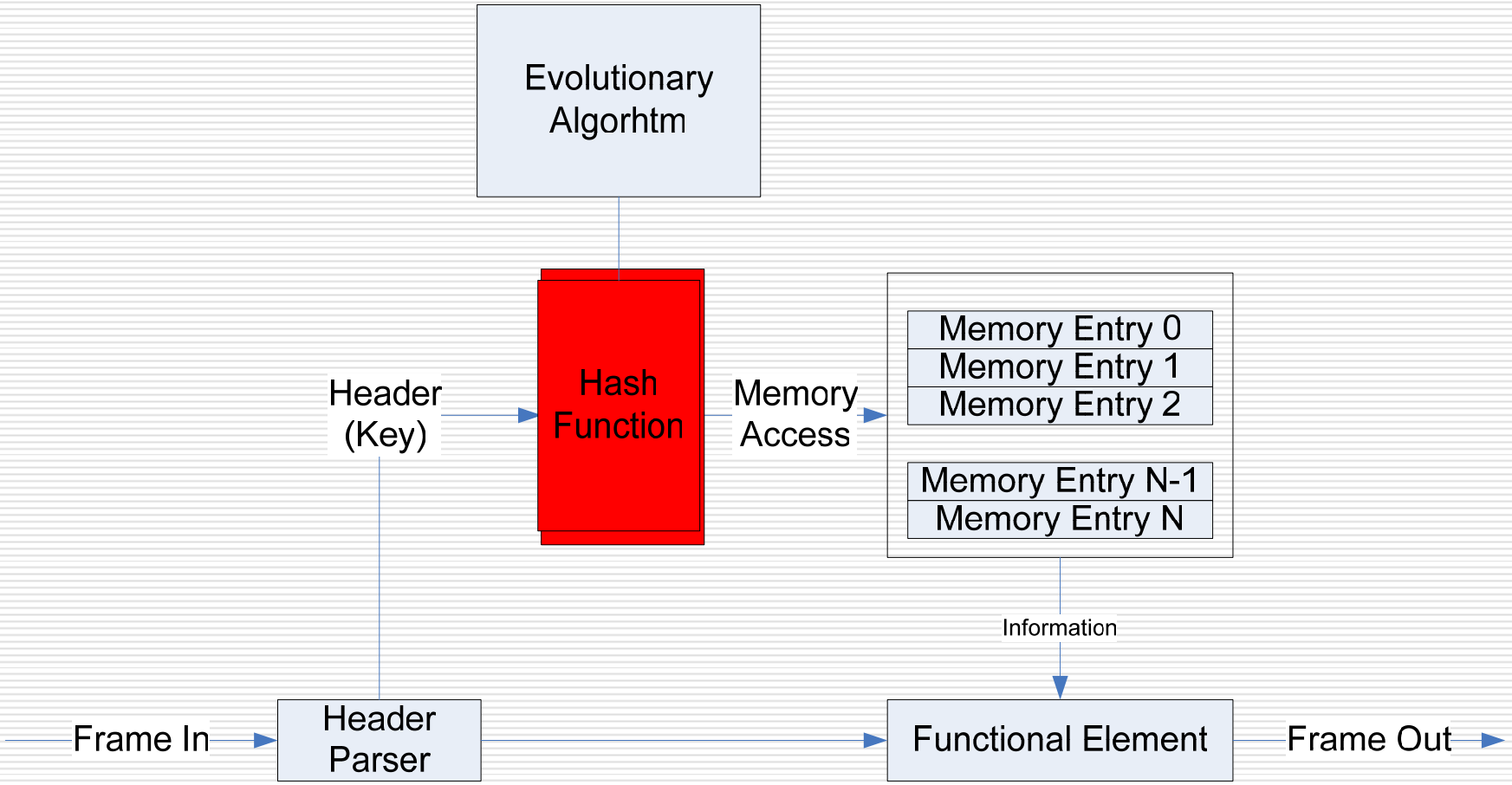
- Search algorithms based on mechanics of natural selection and natural genetics
  - **R**eproduction
  - **M**utation
  - **C**rossover
  - **S**urvival of the fittest



Method of Solution

- Goal: High performance for different and changing key set with low hardware costs
  - Time complexity: O(1)
  - Memory space complexity: O(N)
- Solution: a hash function which can adapt constantly and autonomously to an actual key set: Adaptivity by an evolutionary algorithm in hardware

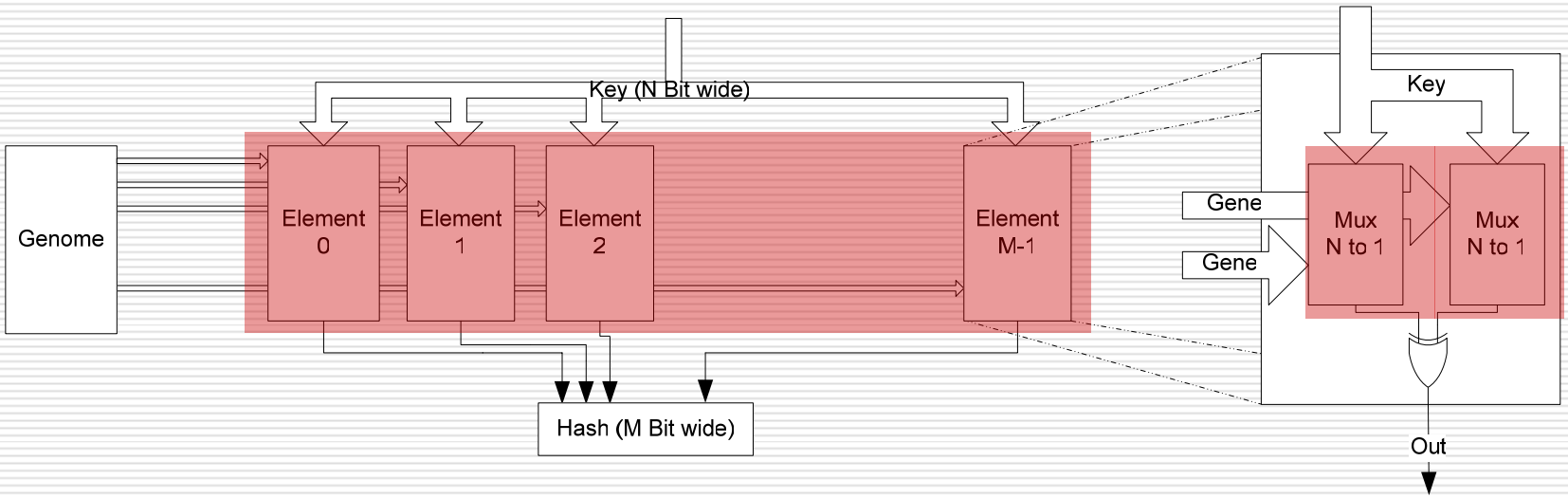
Packet Classification



Hardware Hash Functions

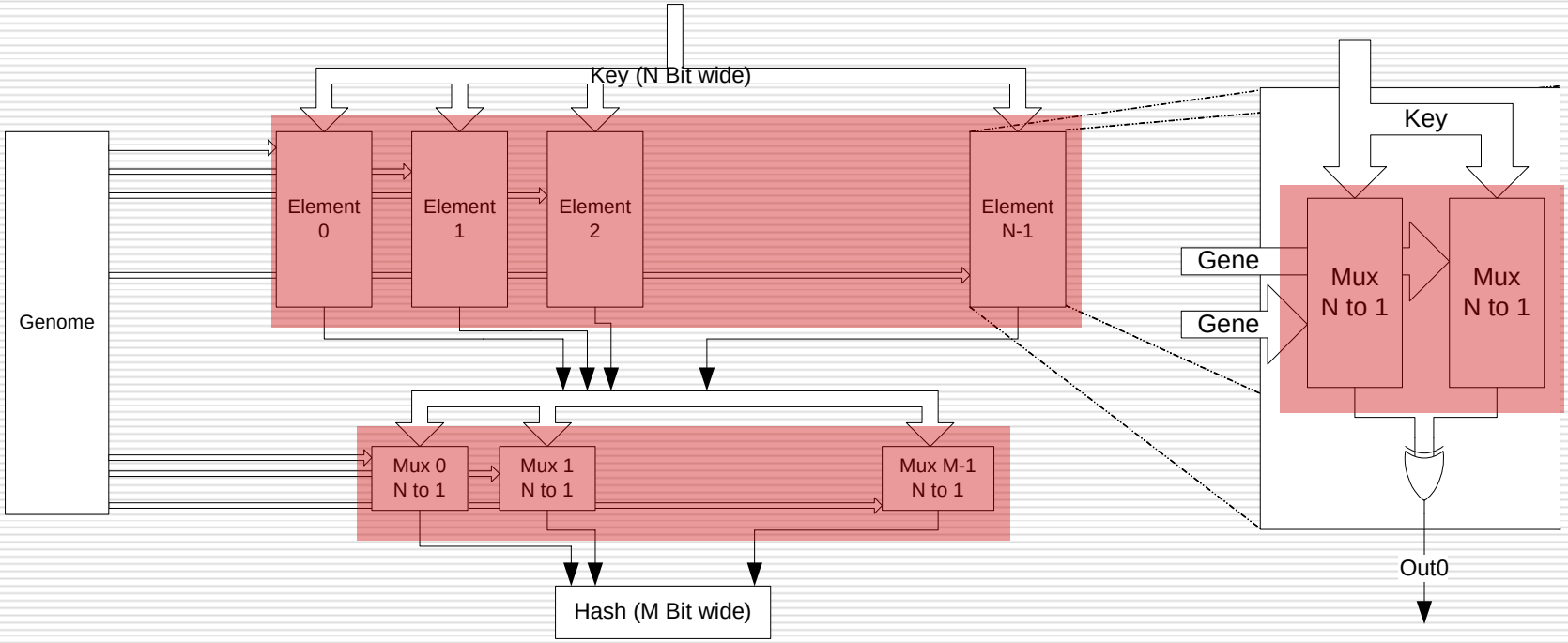
- Three different achitectures of evolvable hash functions developed
- All controlled by registers → genome
  - Genome → register value → functionality

Hardware Hash Function (Hash 1)



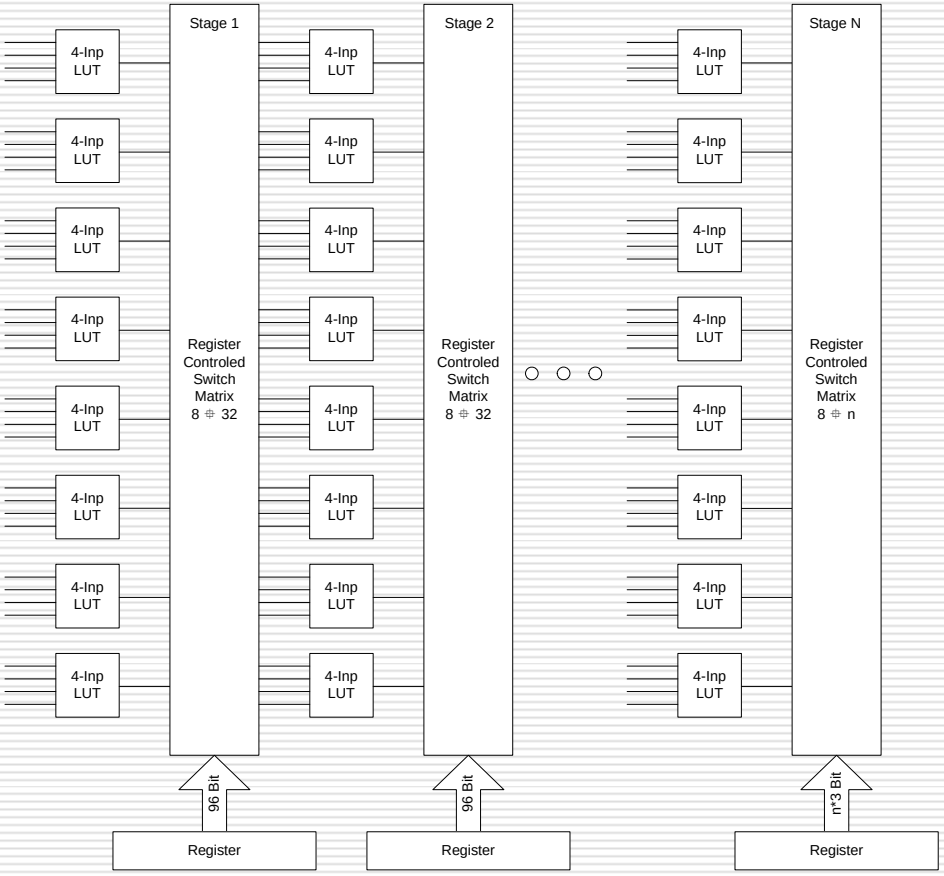
- Size of genome: $M \cdot 2 \cdot \log_2(N)$

Hardware Hash Function (Hash 2)



- Size of genome: $(2 \cdot N + M) \cdot \log_2(N)$

Hardware Hash Function (Hash 3)



- Size of genome:

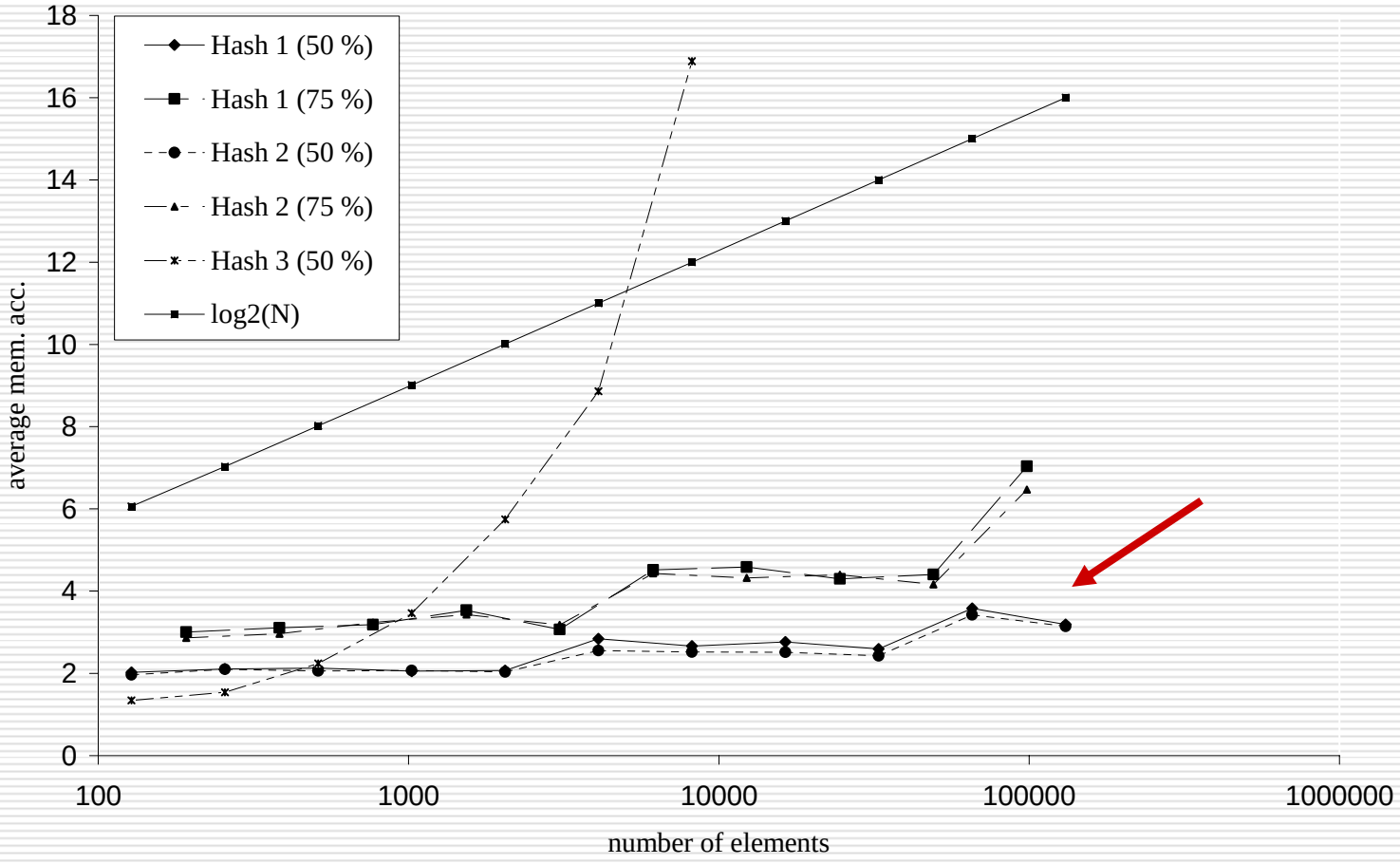
$$\left[\left(N \cdot \log 2(\frac{N}{4})\right) + \frac{N}{4} \cdot 16\right] \cdot (S-1) +$$

$$M \cdot \log 2(\frac{N}{4}) + \frac{N}{4} \cdot 16$$

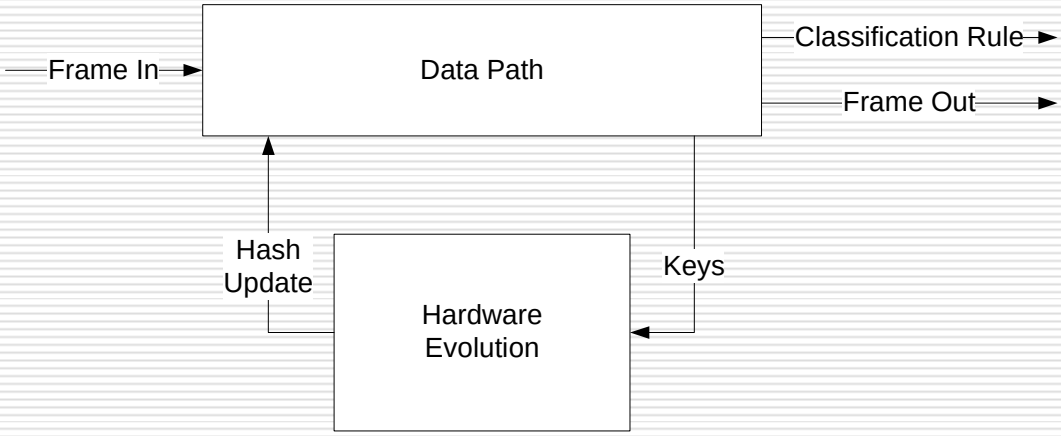
Genetic Algorithm

- 4 Individuals
- 12 Offspring
- Mutation Rate: 1/N
- Survivor Selection:
  - 4 new parents out of 12 offspring and fittest old parent
- Fitness Evaluation – Number of Collisions
  - Fittness Reversed → 0 = perfekt
  - Hash all existing keys
  - H(Key) in the memory used?
    - H(Key) = H(Key)+ prime
    - F++

Simulation Results

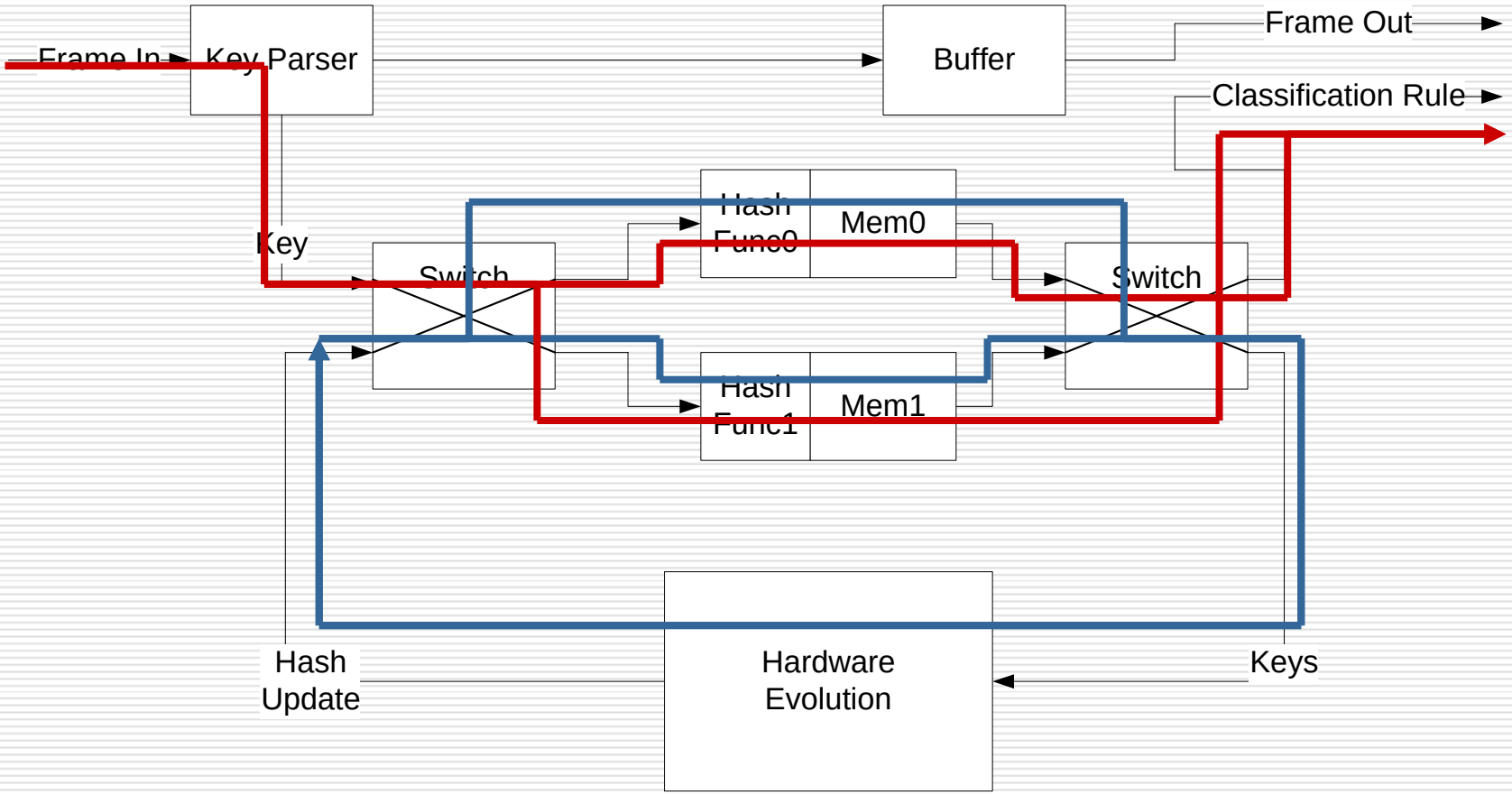


System Architecture

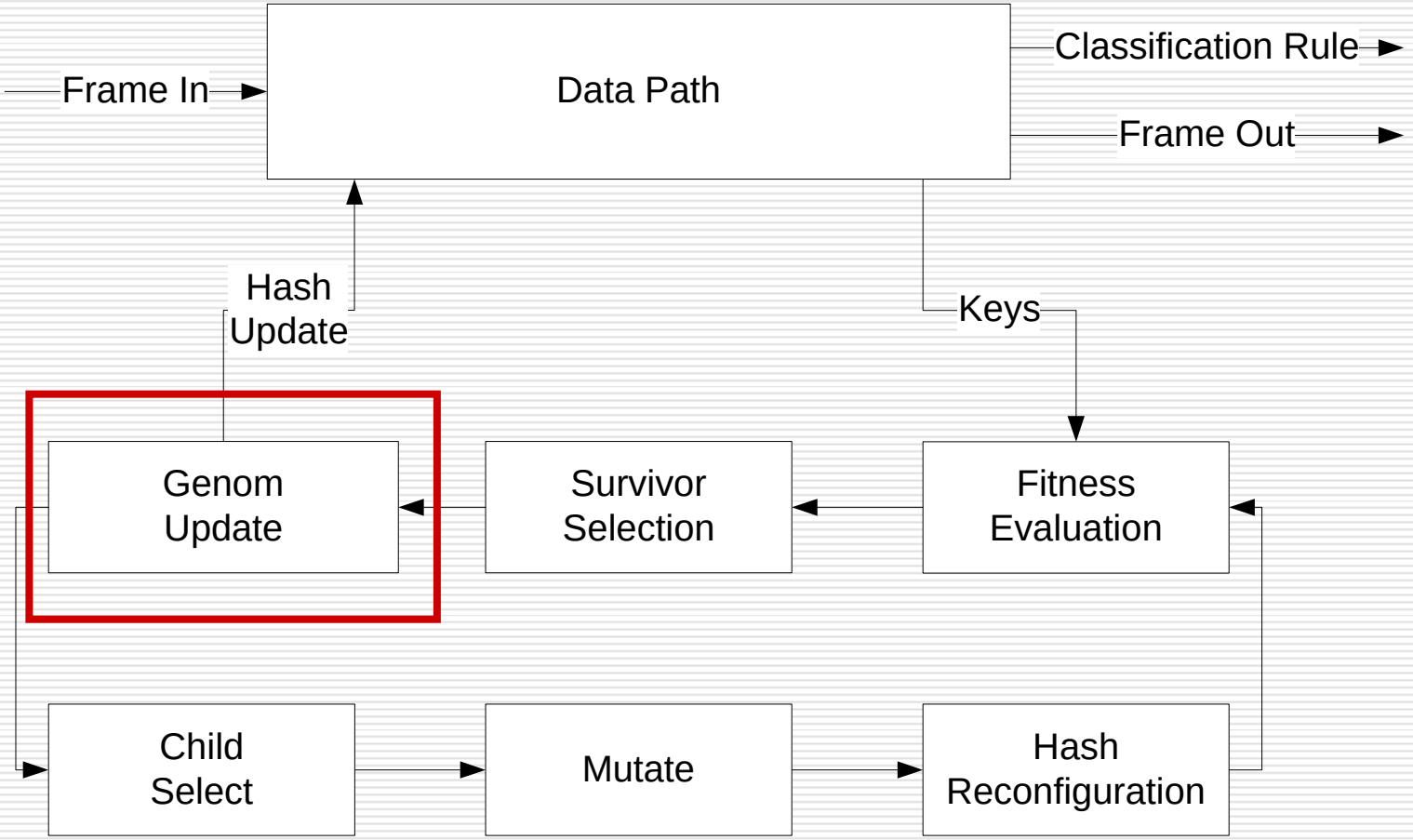


- Hardware consists of a datapath and an evolution module
  - In the datapath the packet classification is done by finding classification rules for incoming frames with the help of the hash functions
  - The evolution module changes the hash function depending on the existing key to increase lookup performance

Data Path


Frame In
Key Parser
Buffer
Frame Out
Classification Rule
Key
Switch
Hash Func0
Mem0
Switch
Hash Func1
Mem1
Hash Update
Hardware Evolution
Keys

Genetic Algorithm



Implementation Results (Xilinx Virtex 2)

Hardware Module			Speed in MHz	Area	
				Logic	BRAMs
Packet Classifier			127	2800	34
	Data Path (hash1)		137	1508	32
		Key Parser	143	159	1
		Switches	155	129	0
		2x Hash Function 1	248	624	0
		2x Hash Function 2	248	1760	0
		2x Memory	163	378	26
		Buffer	208	119	5
	Evolution Module		129	1324	2
		Parent Selection	248	97	0
		Mutation	394	148	0
		Fitness Evaluation	131	685	1
		Survivor Selection	130	439	1

Conlusion and Outlook

- Packet Classifiers consisting of an evolvable hash function can be very effective
  - Time coplexity roughly O(1)
  - Memory demand O(N)
  - Classifier improves over time and adapts to changing key sets
- Limited range of application
- Outlook:
  - Ranges, Wildcards → multiple hash functions with different widths
  - Simulation of the lookup performance with real rules sets and different change rates of the key set
  - Improvement of the performance of the fitness evaluation
  - Better utilization of the internal DRAMs in FPGAs
    - Access via Interleaving with, i.e., 4 banks : max 23 → 6 average 3 → 1.x