Approximate Communication in Networks-on-Chip based Architectures

Giuseppe Ascia, Vincenzo Catania, Salvatore Monteleone,
Maurizio Palesi, Davide Patti

DIEEI, University of Catania

3rd Italian Workshop on Embedded Systems
Department of Information Engineering and Mathematics
University of Siena
September 13-14, 2018

Outline

- Generalities on Approximate Computing
- Approximate Communication
- Hardware/Software Interface
- Case study
- Conclusions

Trends

- Overall energy consumption of computer systems rapidly increases
- Despite of
  - Advances in semiconductor technologies
  - Advances in energy efficient design techniques

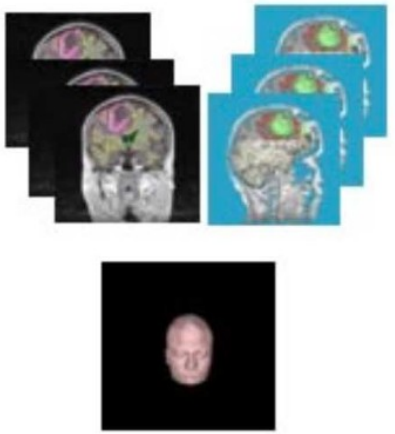
RMS Applications

- Emerging of RMS applications
  - Wide computing spectrum
    - From mobile IoT to large-scale data centers

Recognition

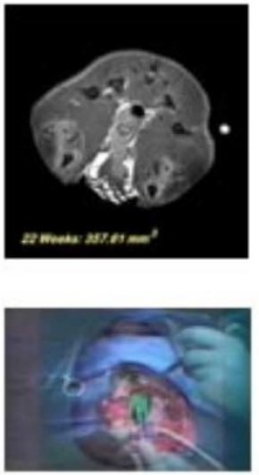
What is a tumor?

Mining



Is there a tumor here?

Synthesis



What if the tumor progresses?

RMS Applications

- Intrinsic error-resilence property
- Noisy and redundant data
- No unique solution

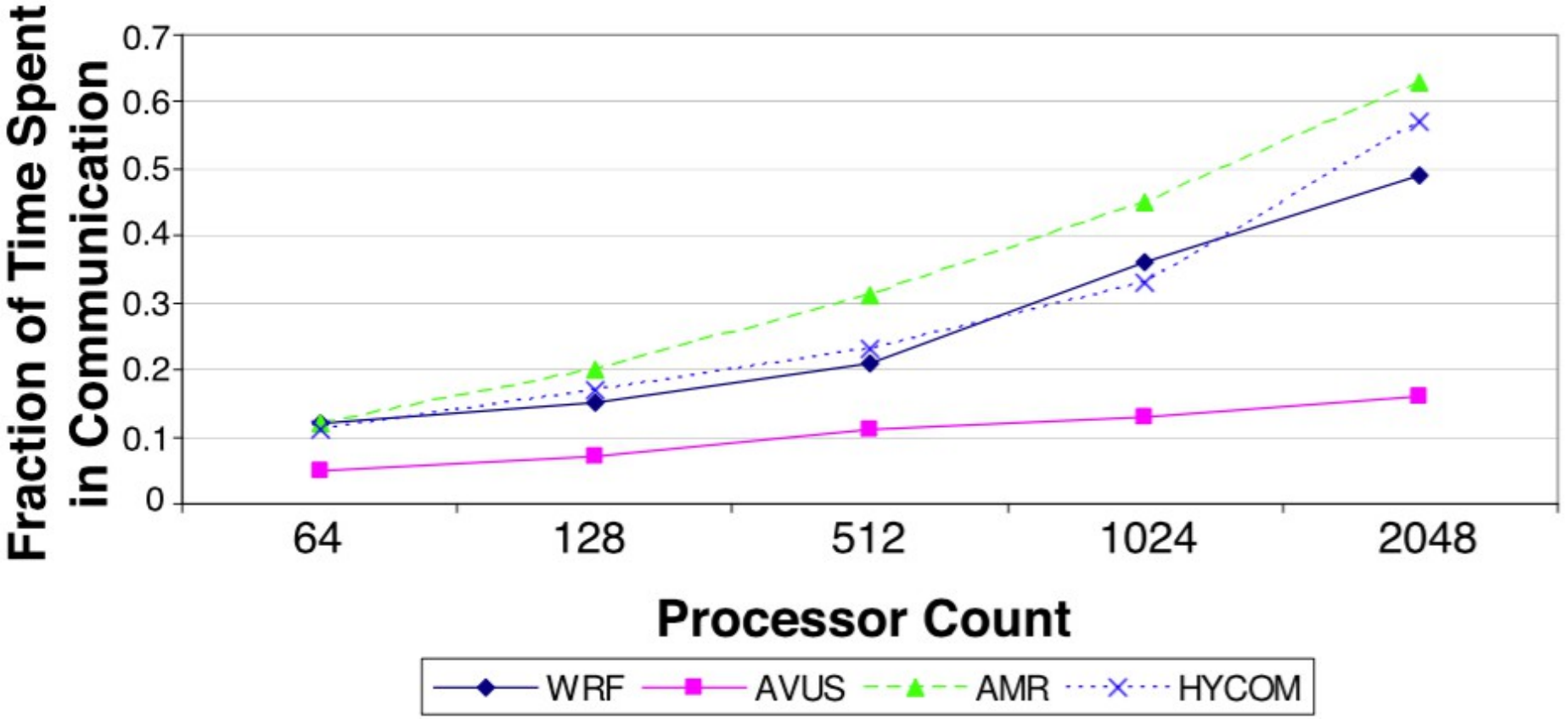
Approximate Computing

- Relax the numerical equivalence
  - Between specification and impementation
  - (of error-tolerant applications)
- Price to pay
  - Application results accuracy
- In return for
  - Higher, scalable performance and energy efficiency

Approximate Kernels

- Hardware level
  - Use less accurate yet energy-efficient circuits
  - Reduce the supply voltage
  - …
- Software level
  - Ignore certain computations
  - Ignore certain memory accesses
  - ...

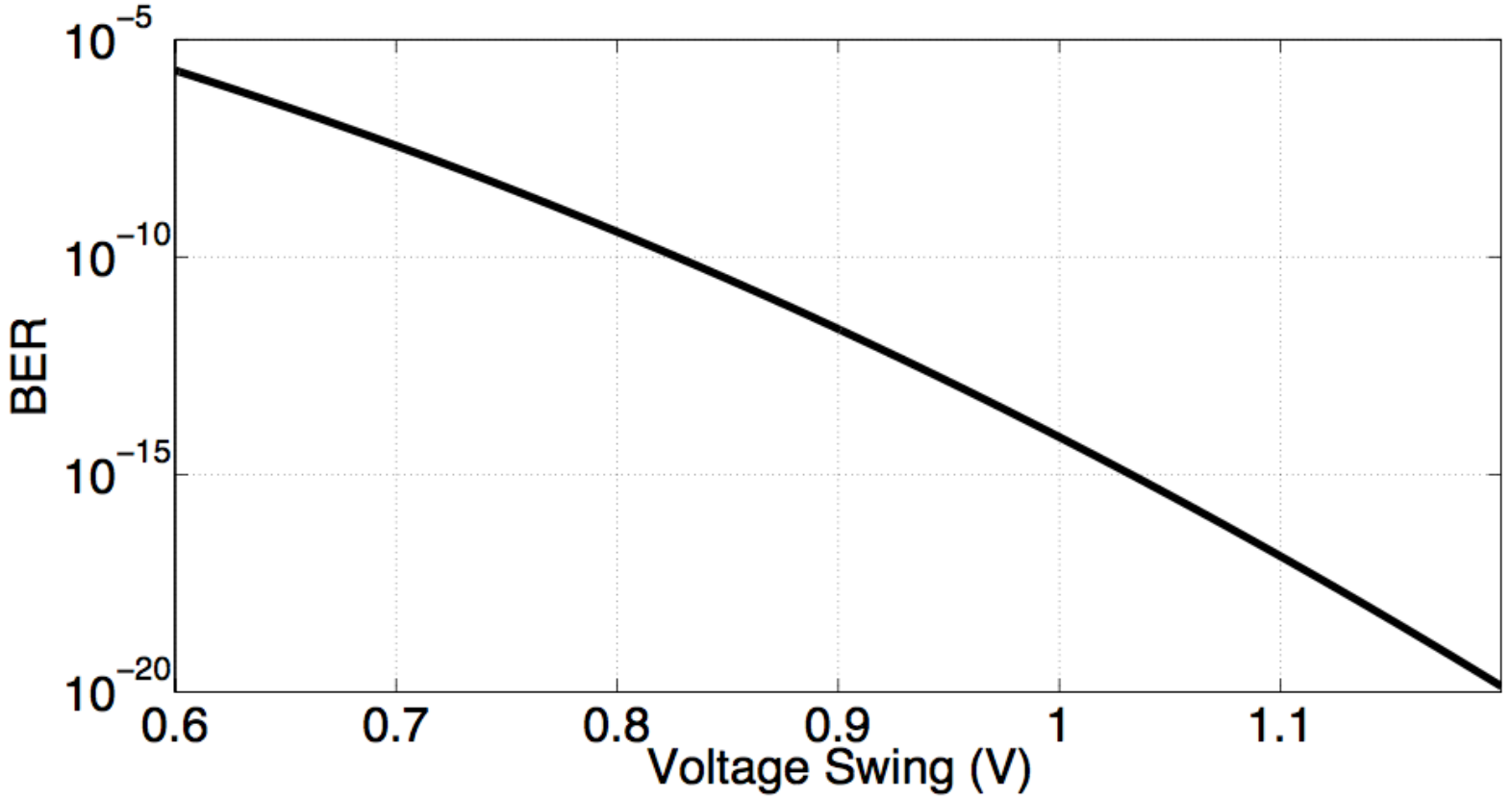
Communication Overhead



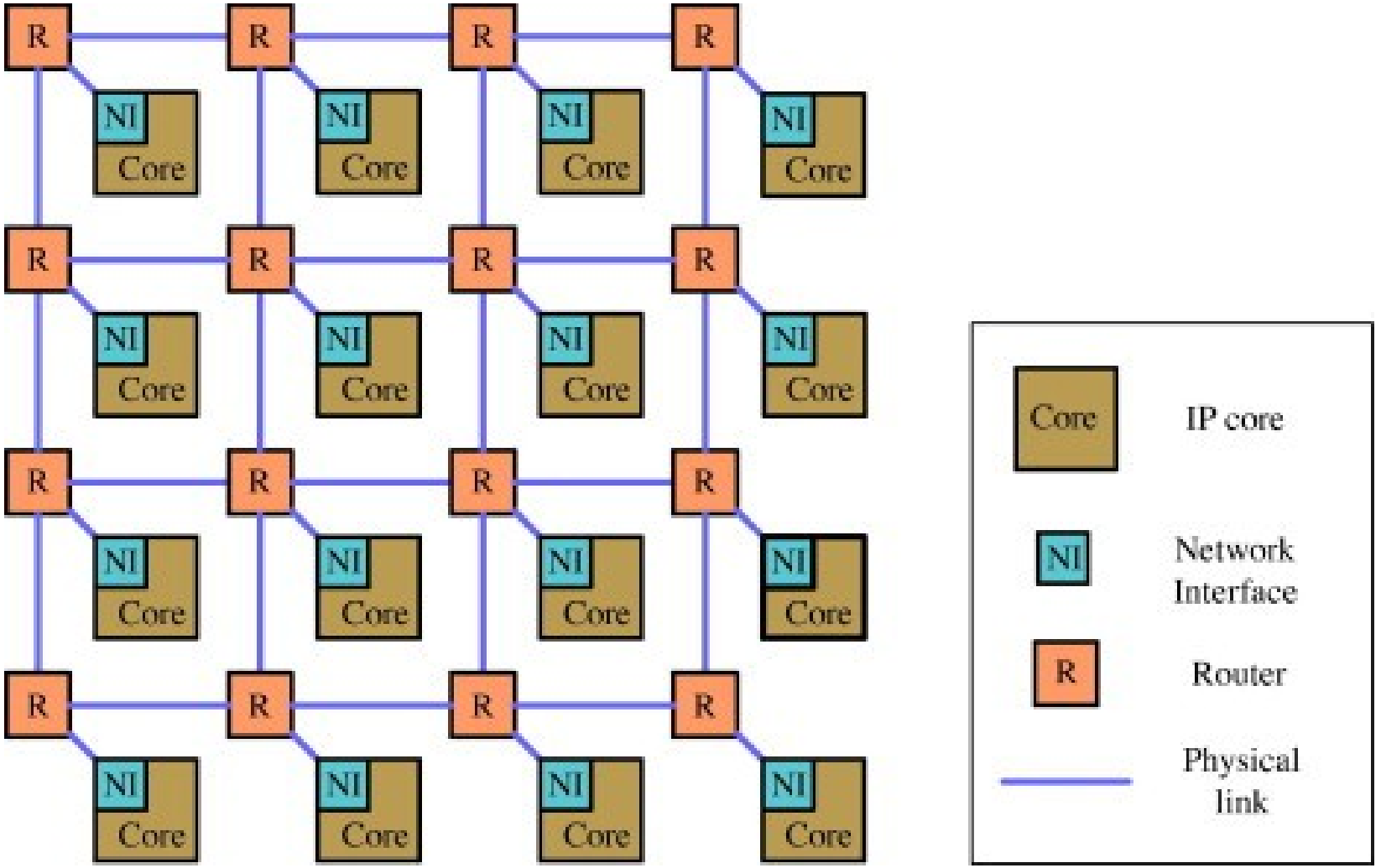
Communication Cost

- Interconnection networks consume 10% to 20% of the power in current HPC systems
  - Majority due to network's links
- NoC based design
  - More than one-third of the chip's power consumption

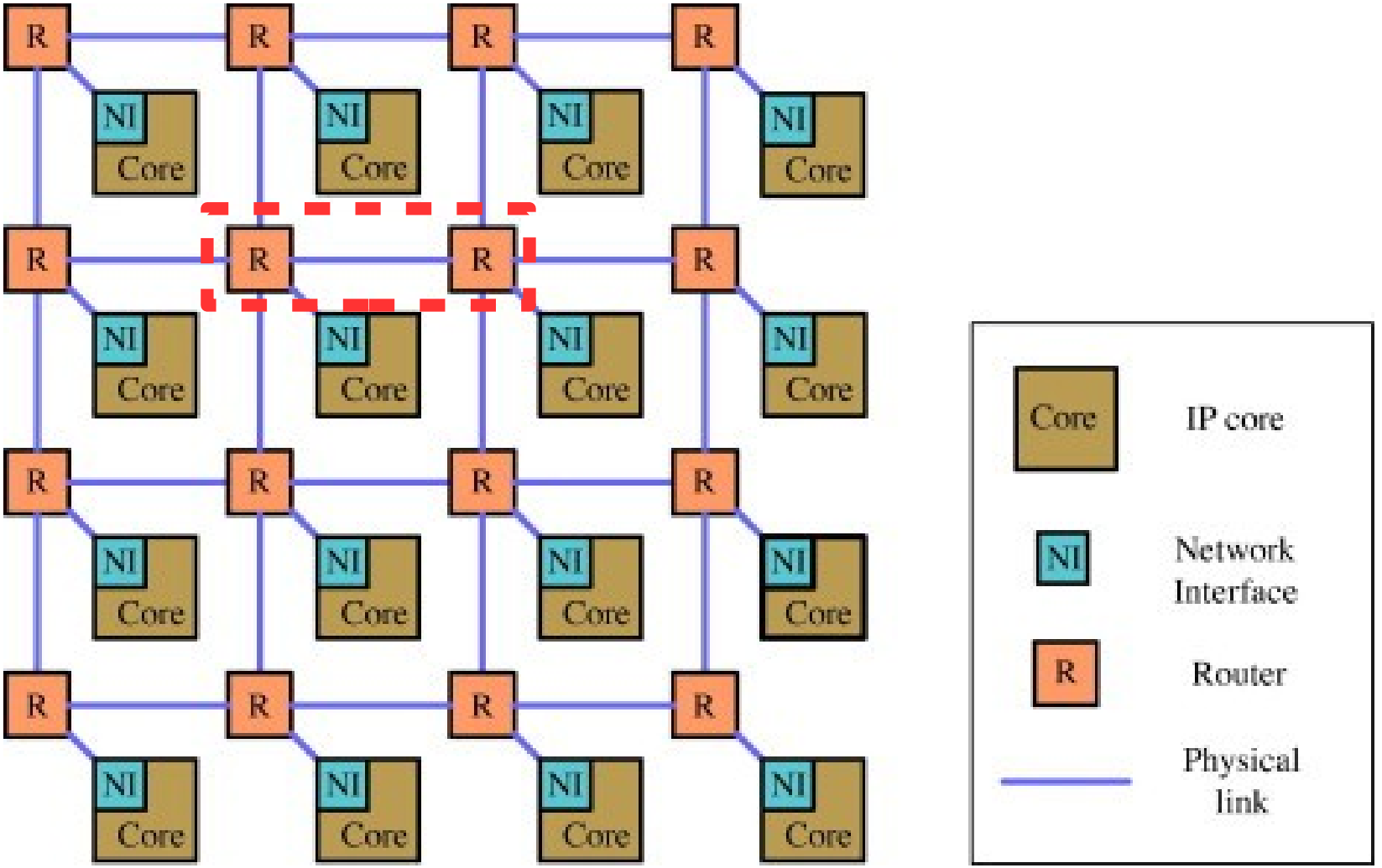
BER *vs.* Voltage Swing



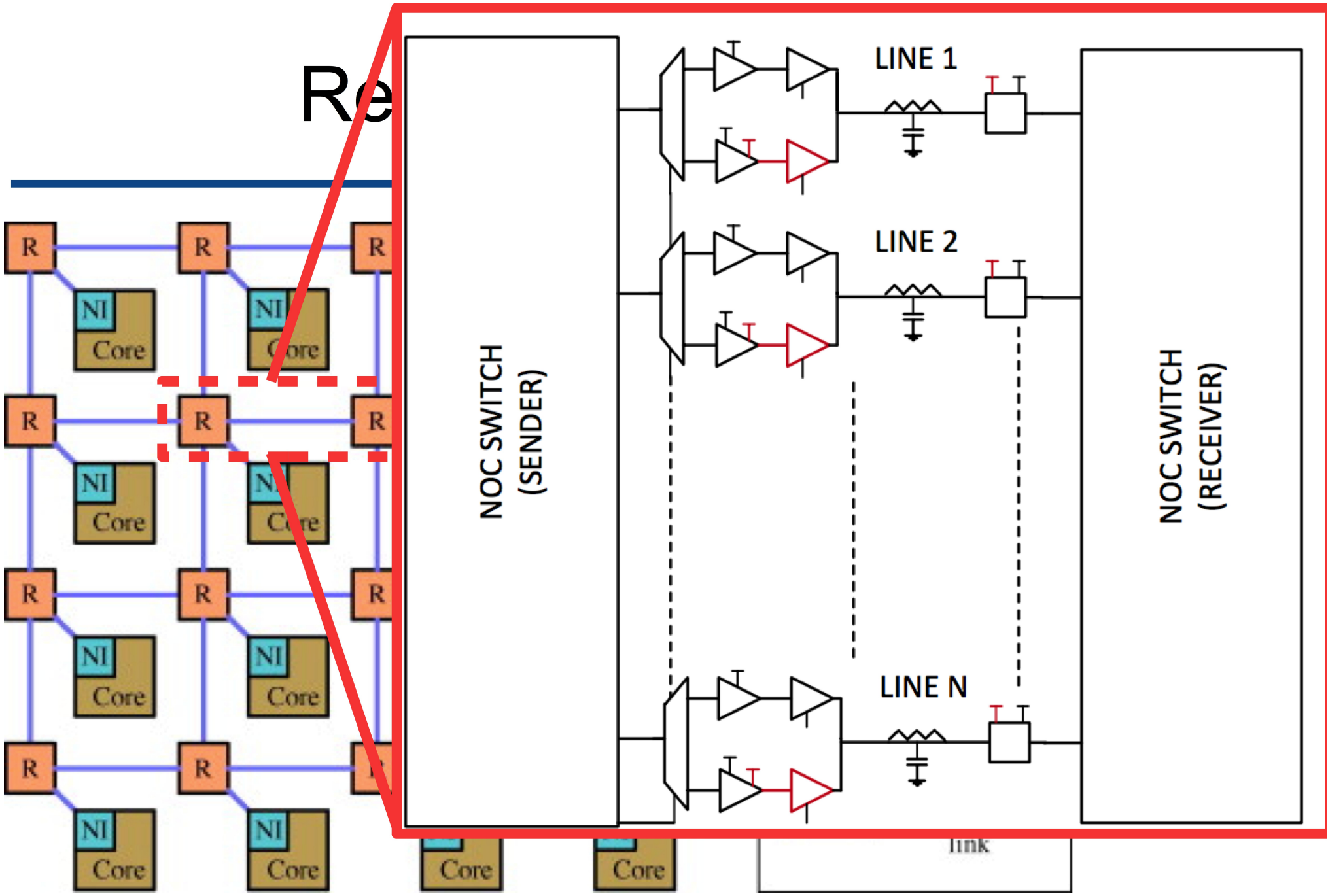
Reconfigurable Link



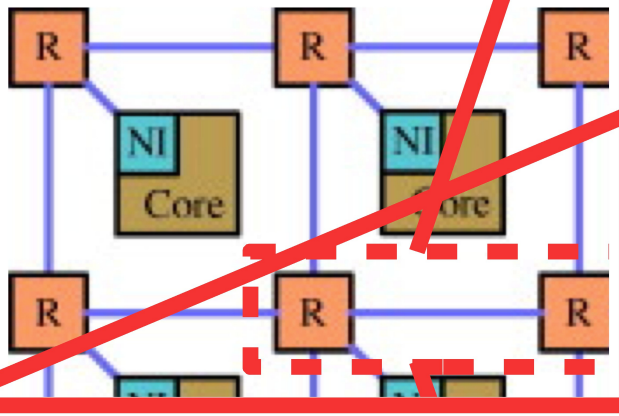
Reconfigurable Link

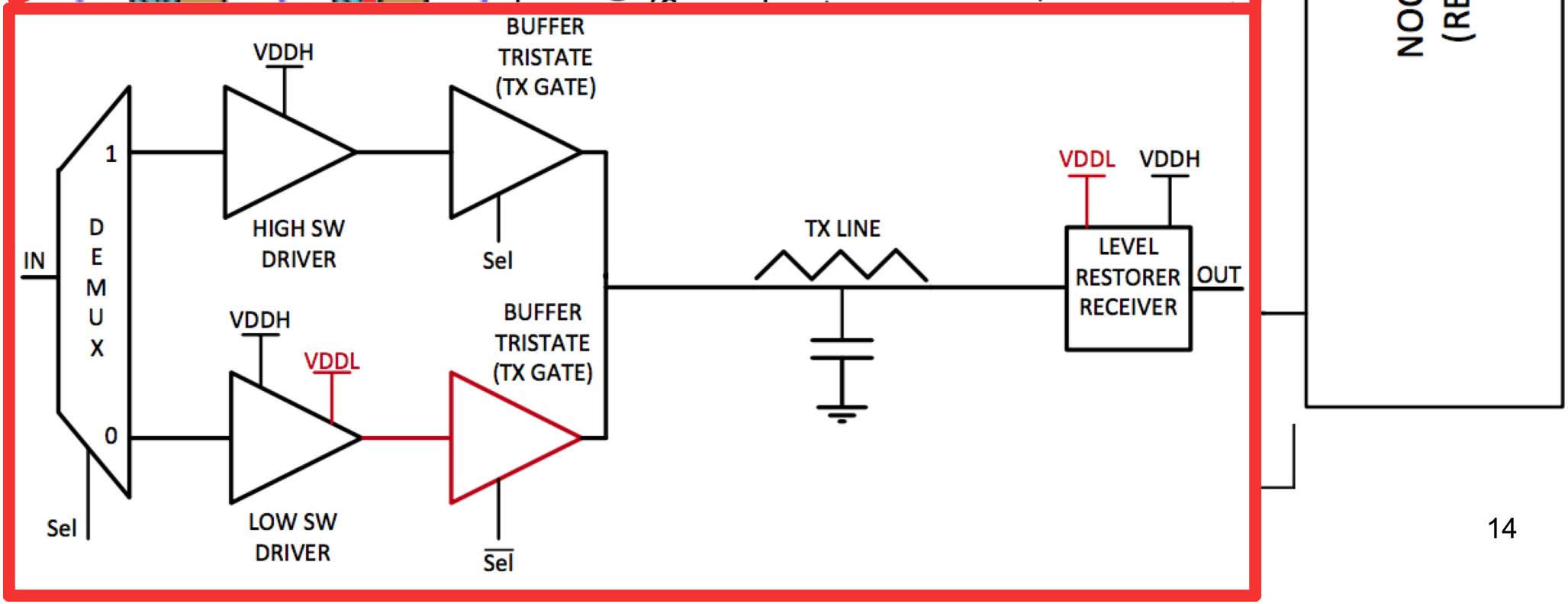


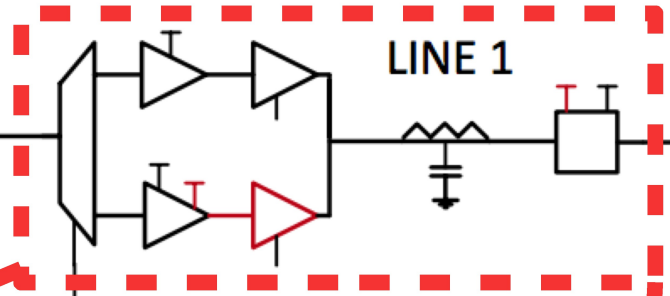
Re

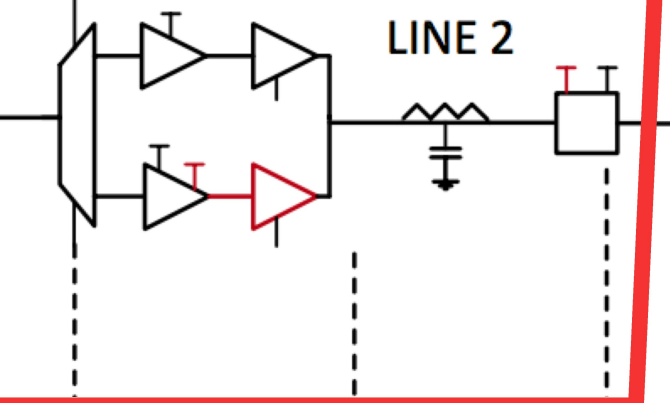

R
NI
Core
NOC SWITCH
(SENDER)
LINE 1
LINE 2
LINE N
NOC SWITCH
(RECEIVER)

Re









C SWITCH
SENDER)

NOC SWITCH
(RECEIVER)

HSPICE Link Simulation

- HSPICE, 45 nm CMOS technology from Nangate
  - 10 metal layers
  - 3 mm link line using the seventh metal layer

	Conventional	Configurable	
	VDDH	VDDH	VDDL
Technology	1.1 V, 10 metal, 45 nm CMOS LVT		
Interconnect (Metal 7)	Width 0.4 μm, Space 0.32 μm, Length 2.8 mm Rwire 225 Ω, Cwire 946 fF		
Supply	1.1 V	1.1 V	0.6 V
Worst case total delay	214 ps		410 ps
Avg. Energy/Transition	512 fJ	527 fJ	152 fJ
BER	1.3E-17	1.3E-17	3.8E-6

HSPICE Link Simulation

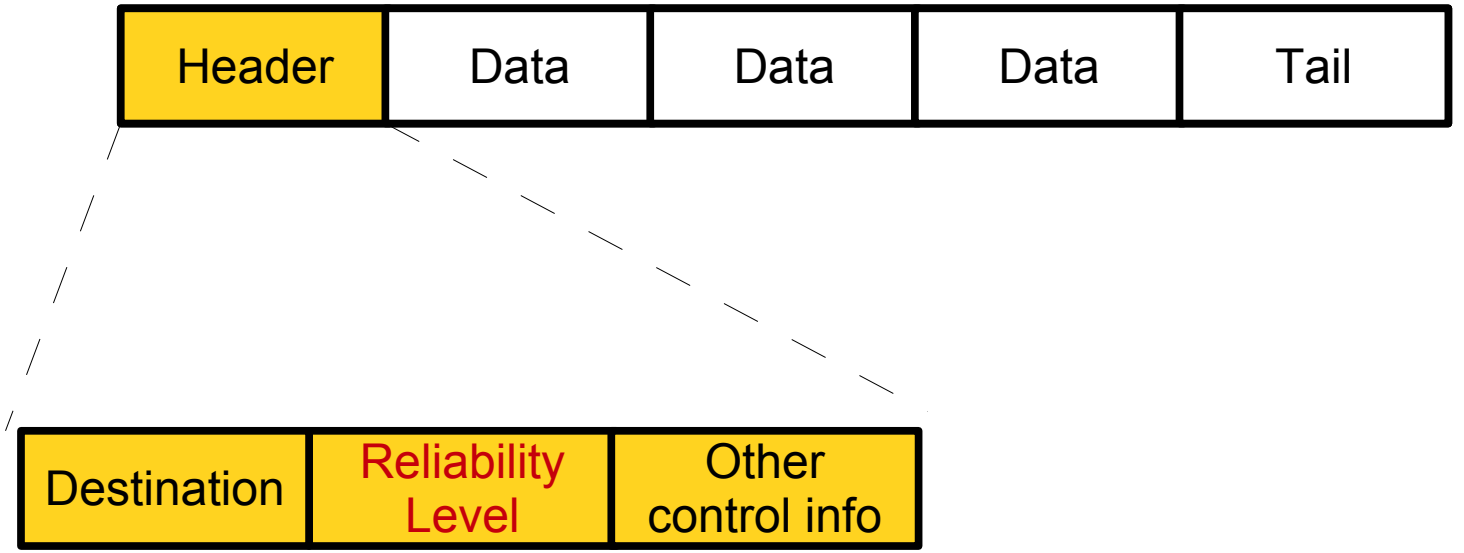
	Conventional	Configurable	
	VDDH	VDDH	VDDL
Technology	1.1 V, 10		CMOS LVT
Interconnect	Width 0.4 μm,		m, Length 2.8 mm
(Metal 7)		Rwire 225 Ω, Cwire 946 fF	
Supply	1.1 V		0.6 V
Worst case total delay	214 ps		410 ps
Avg. Energy/Transition	512 fJ	527 fJ	152 fJ
BER	1.3E-17	1.3E-17	3.8E-6

70% saving

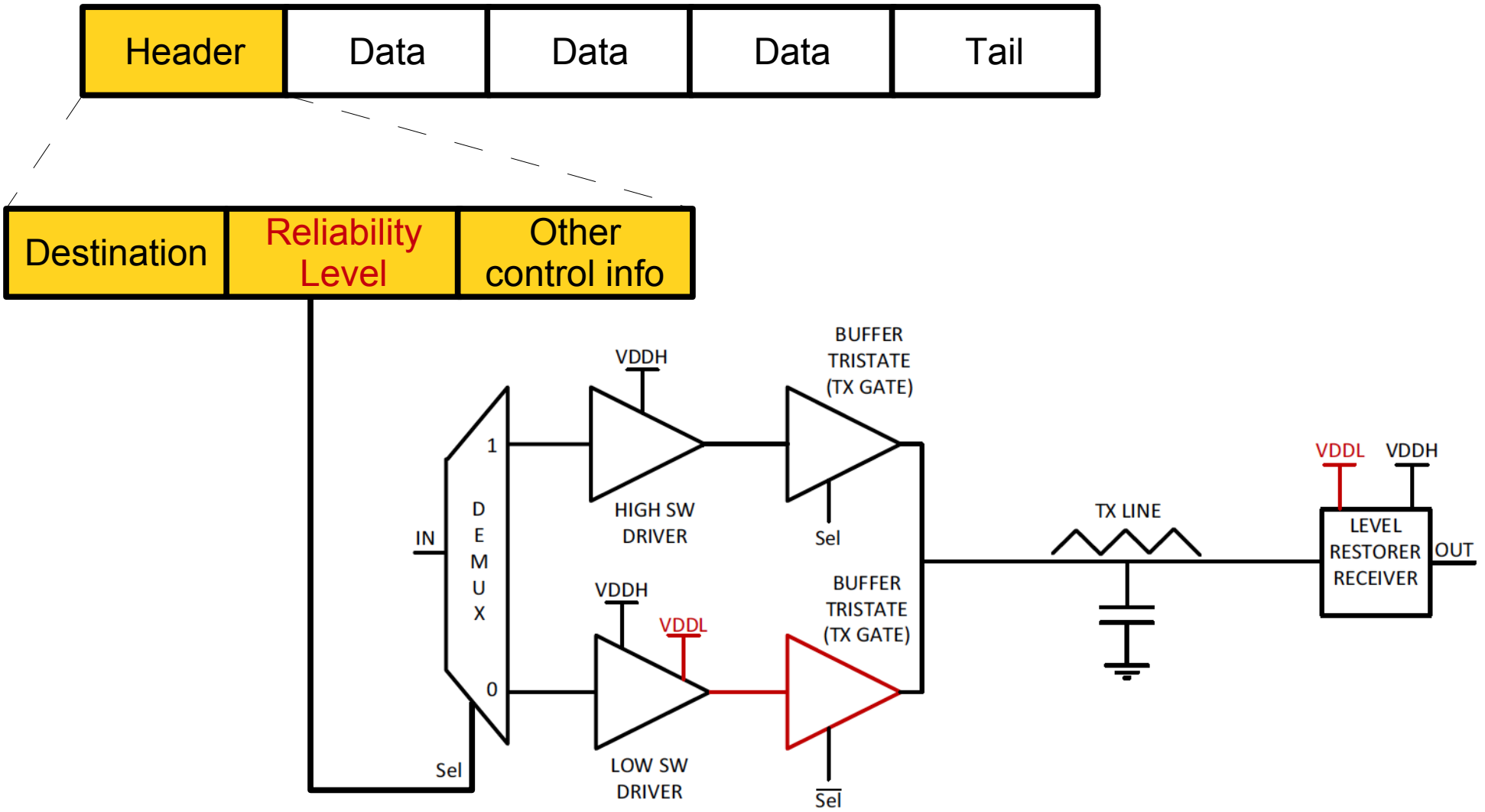
3% overhead

Implementation

- `Send(data, destination, reliability_level)`

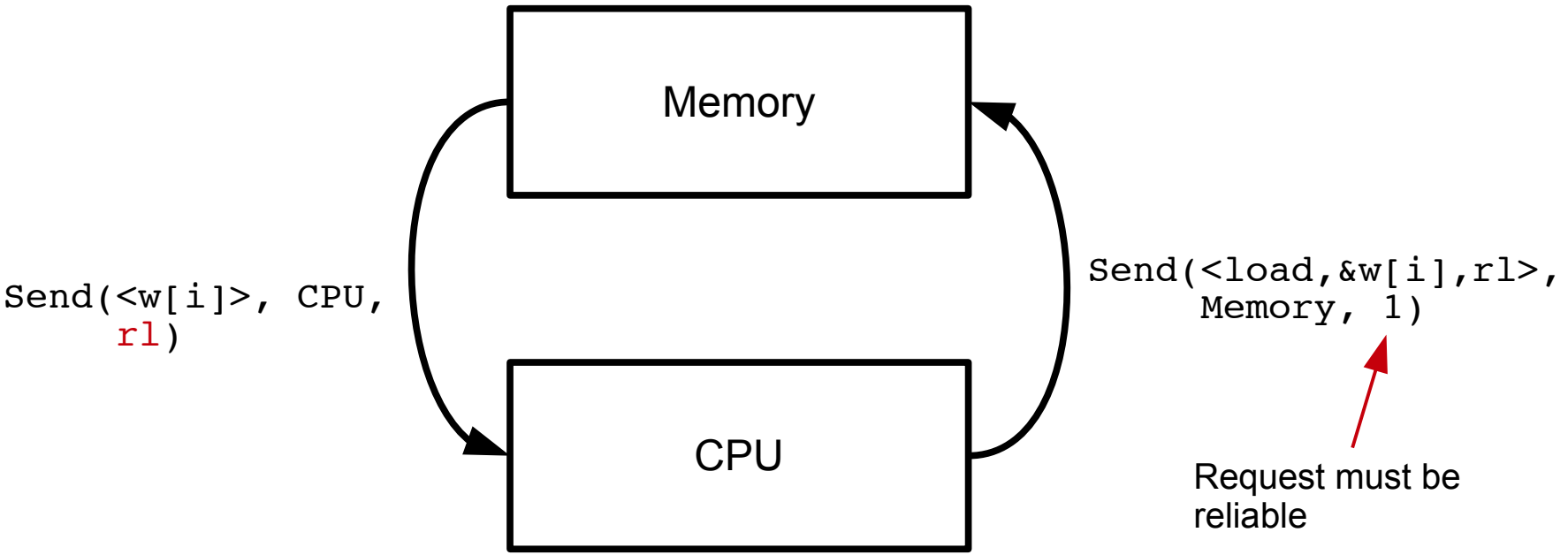


Implementation



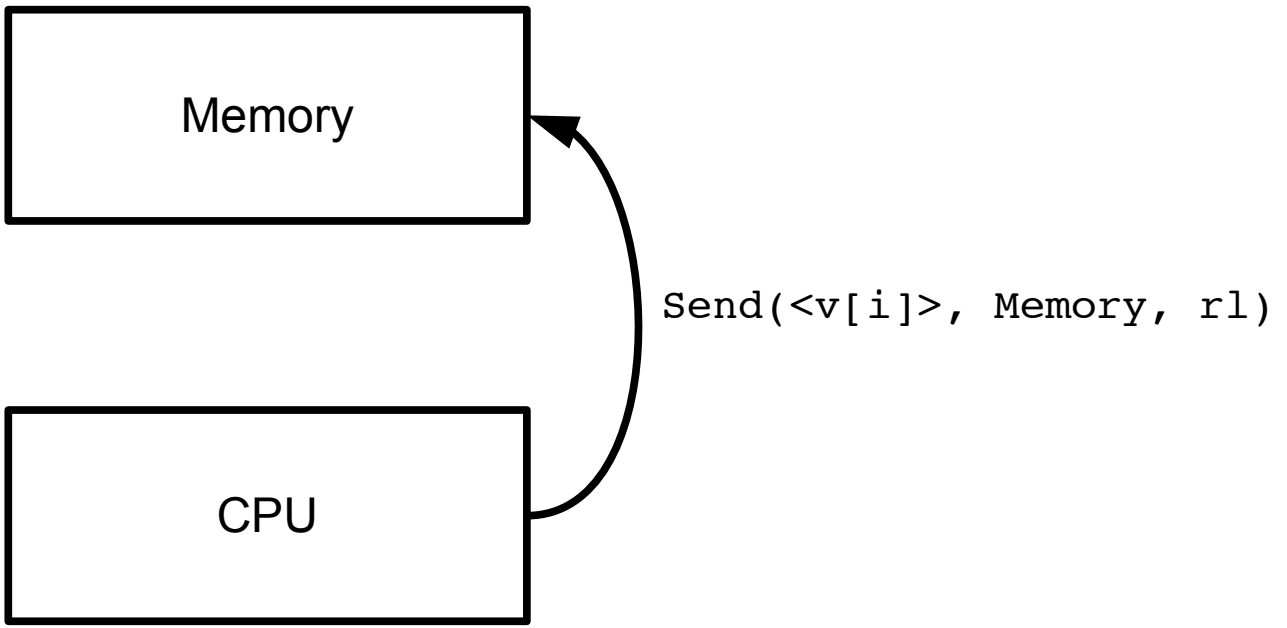
Programming Example

```
#pragma resilient(w, rl)
for (i=0; i<n; i++)
    v[i] = f(w[i]);
```

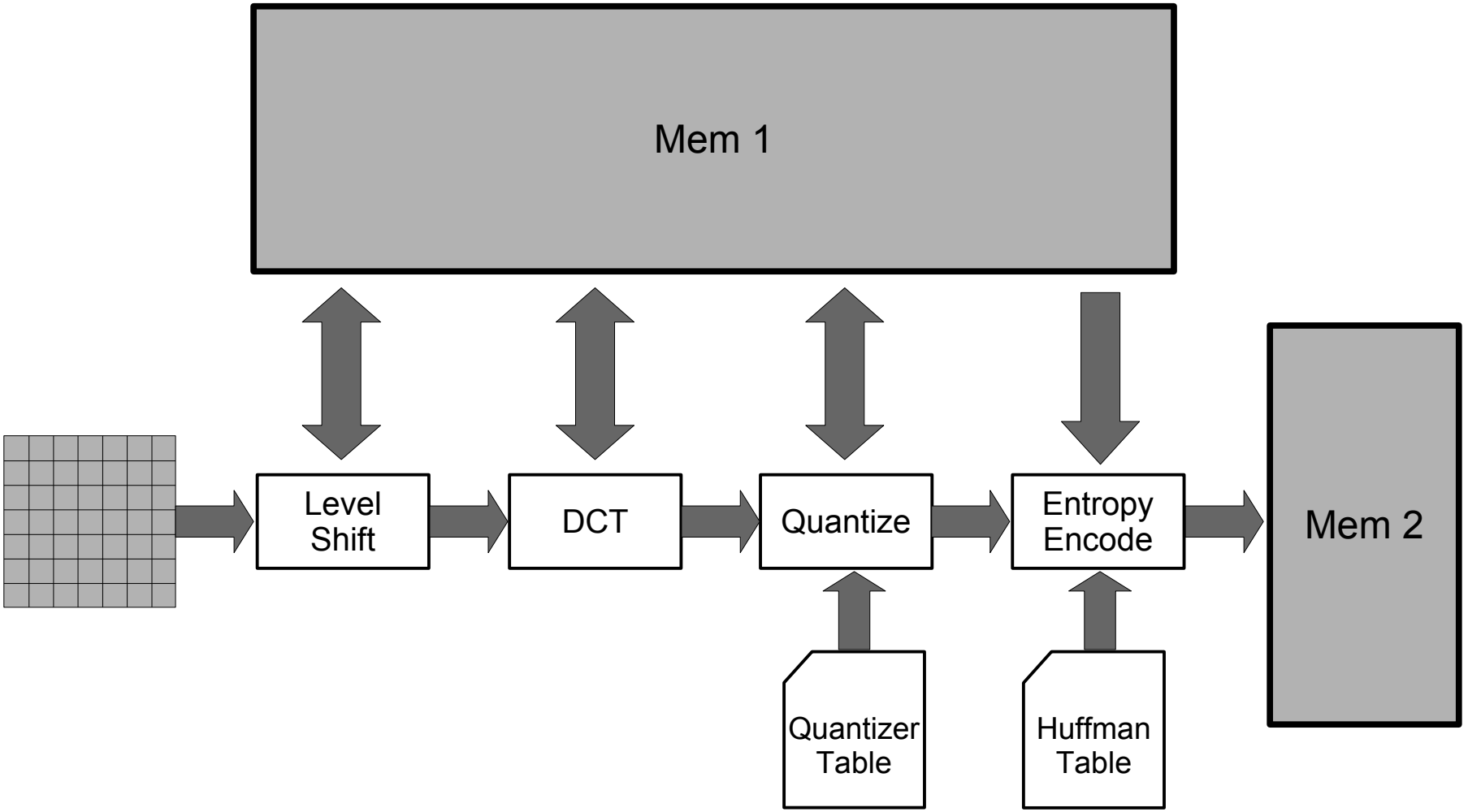


Programming Example

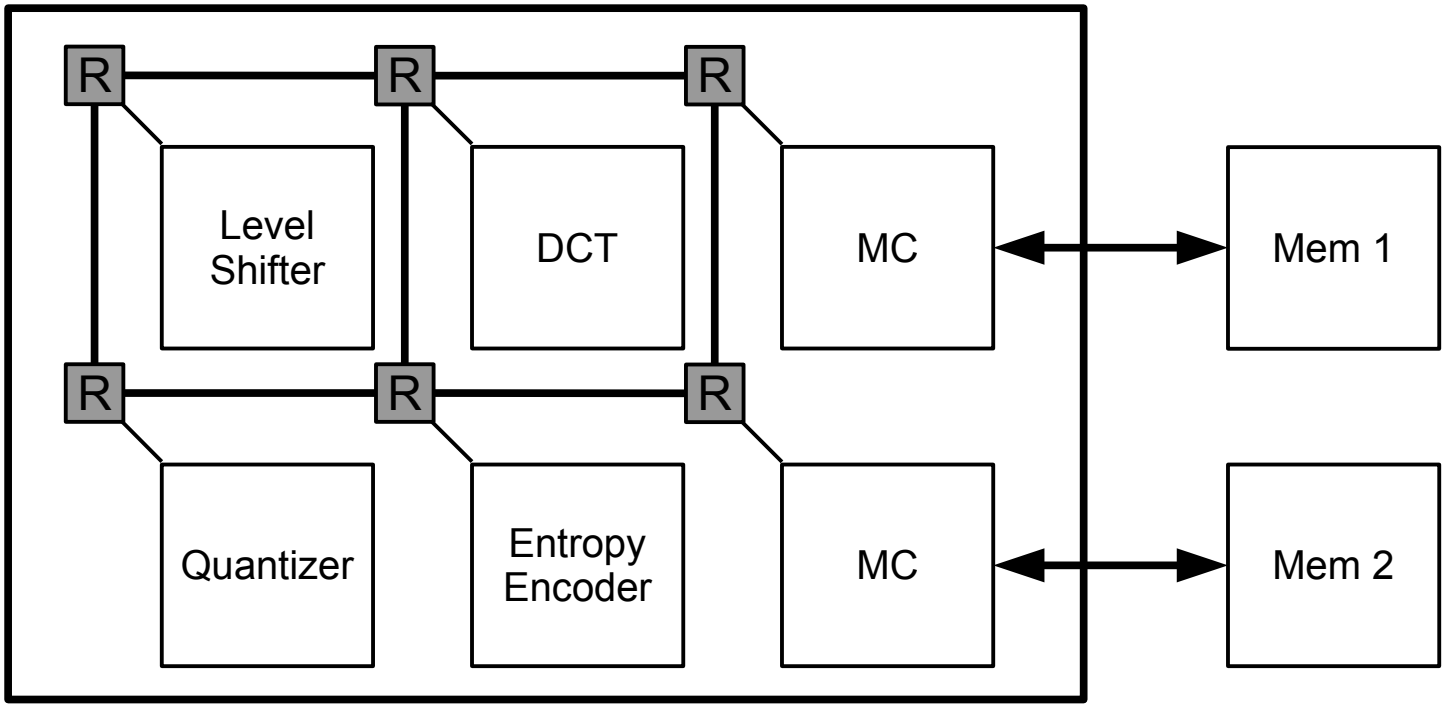
```
#pragma resilient(v, rl)
for (i=0; i<n; i++)
   v[i] = f(w[i]);
```



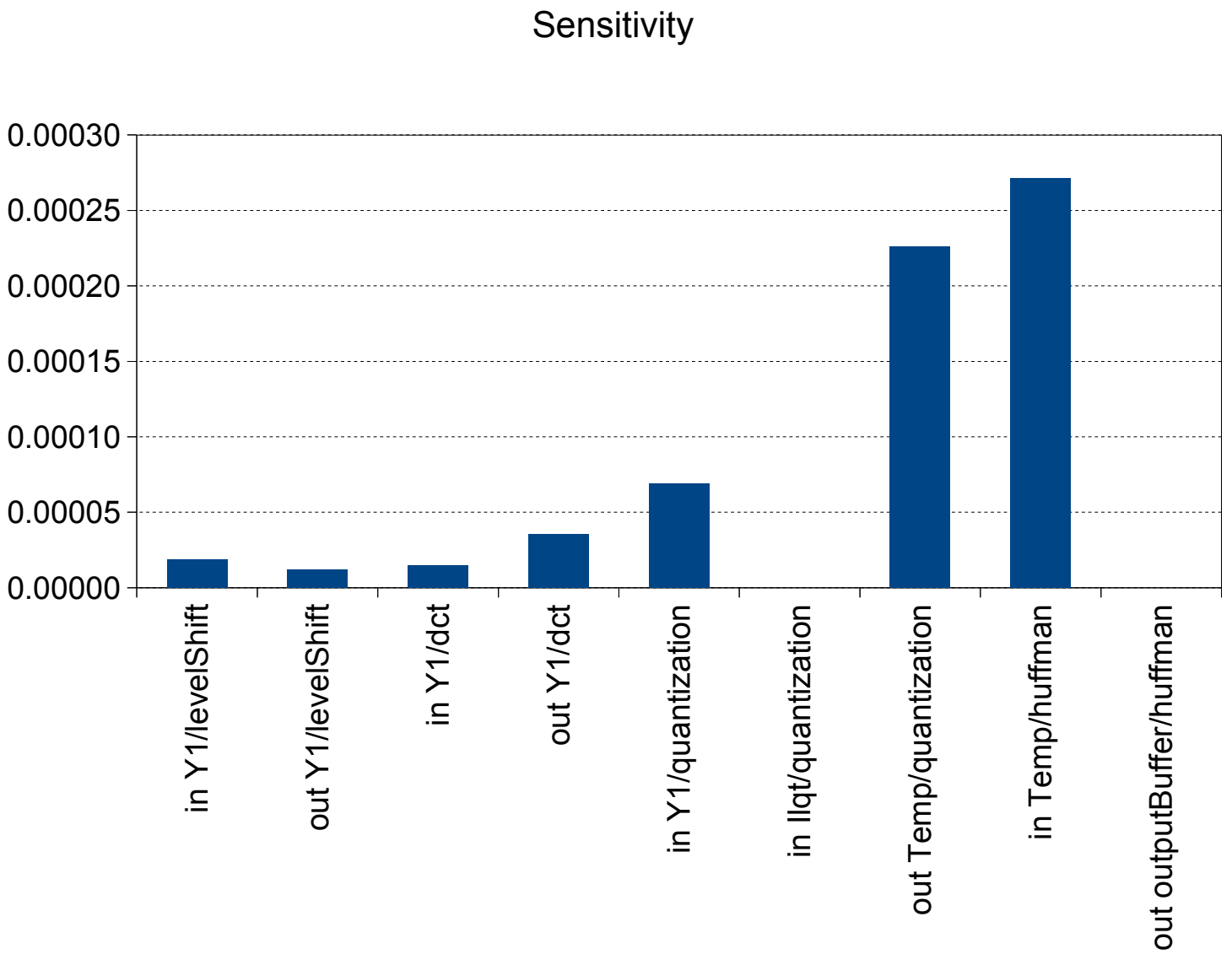
JPEG Encoder



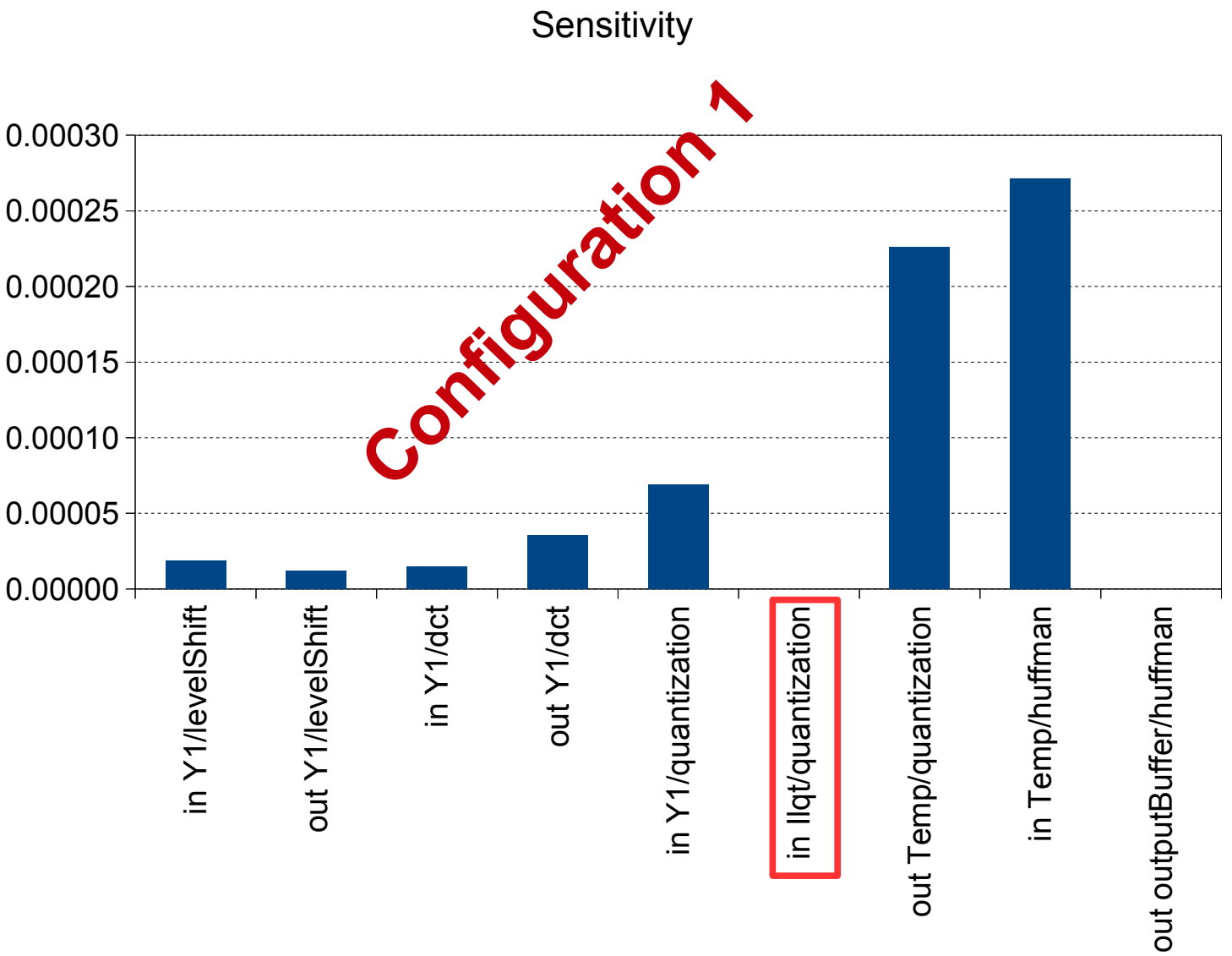
Experiments



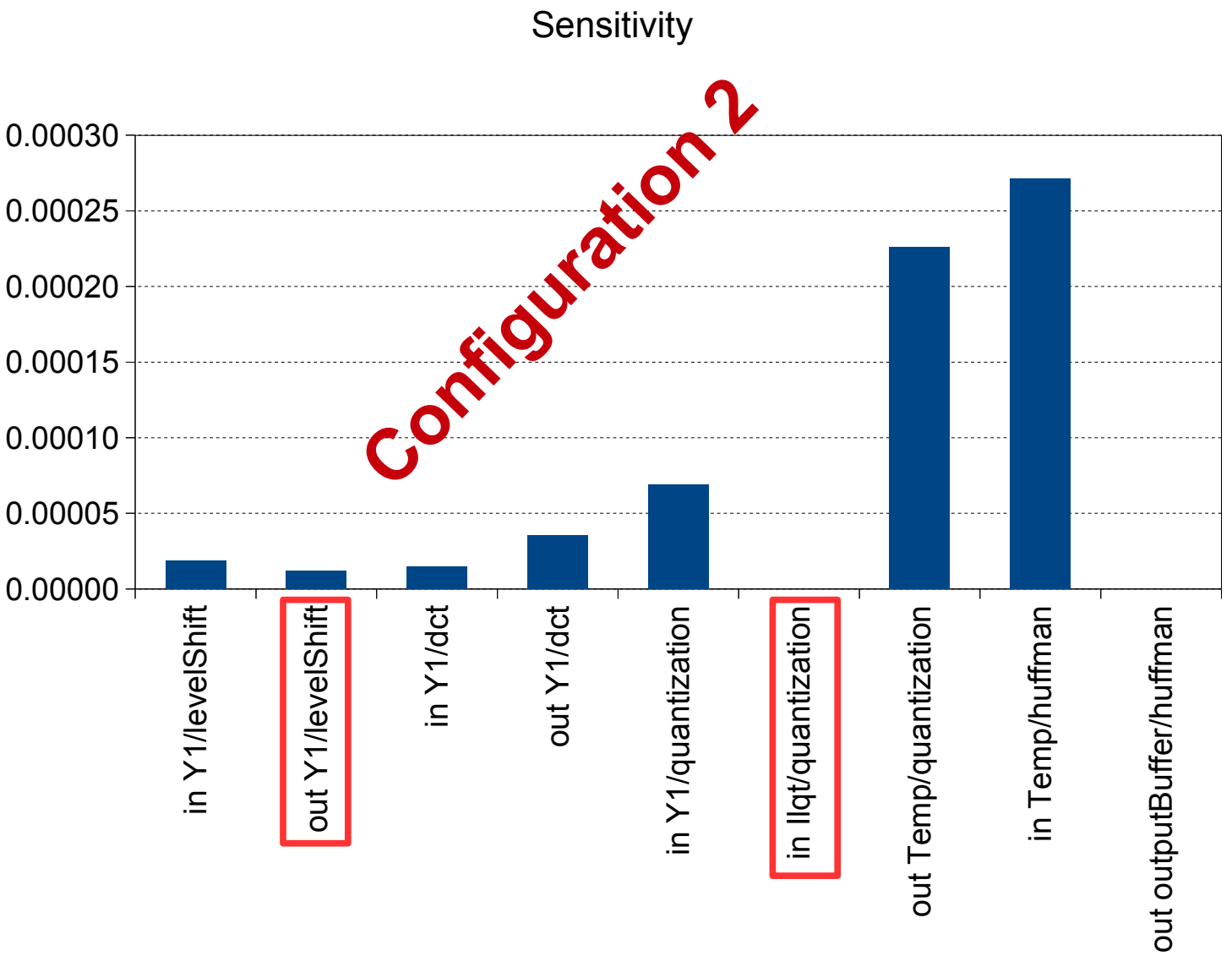
Sensitivity Analysis



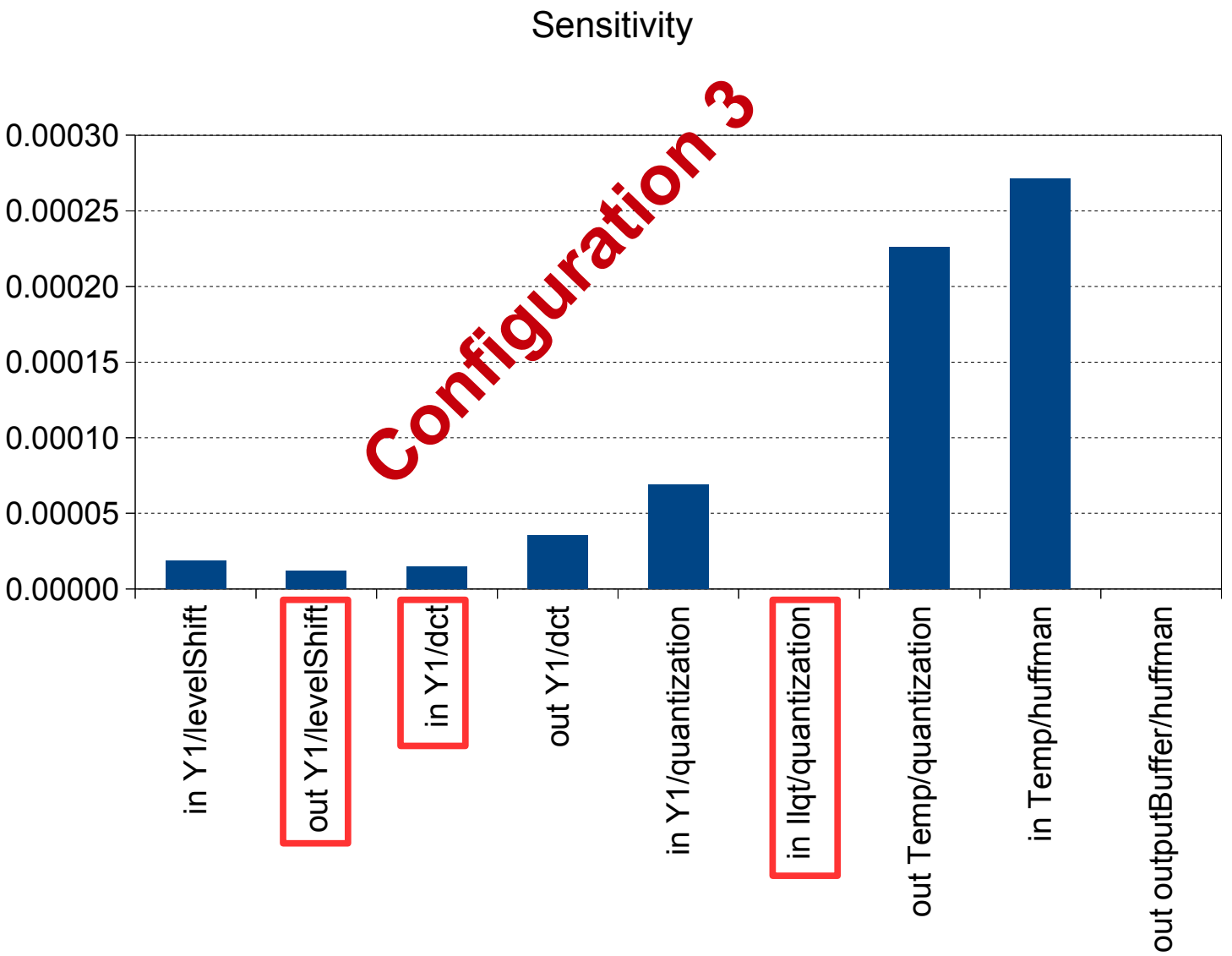
Sensitivity Analysis



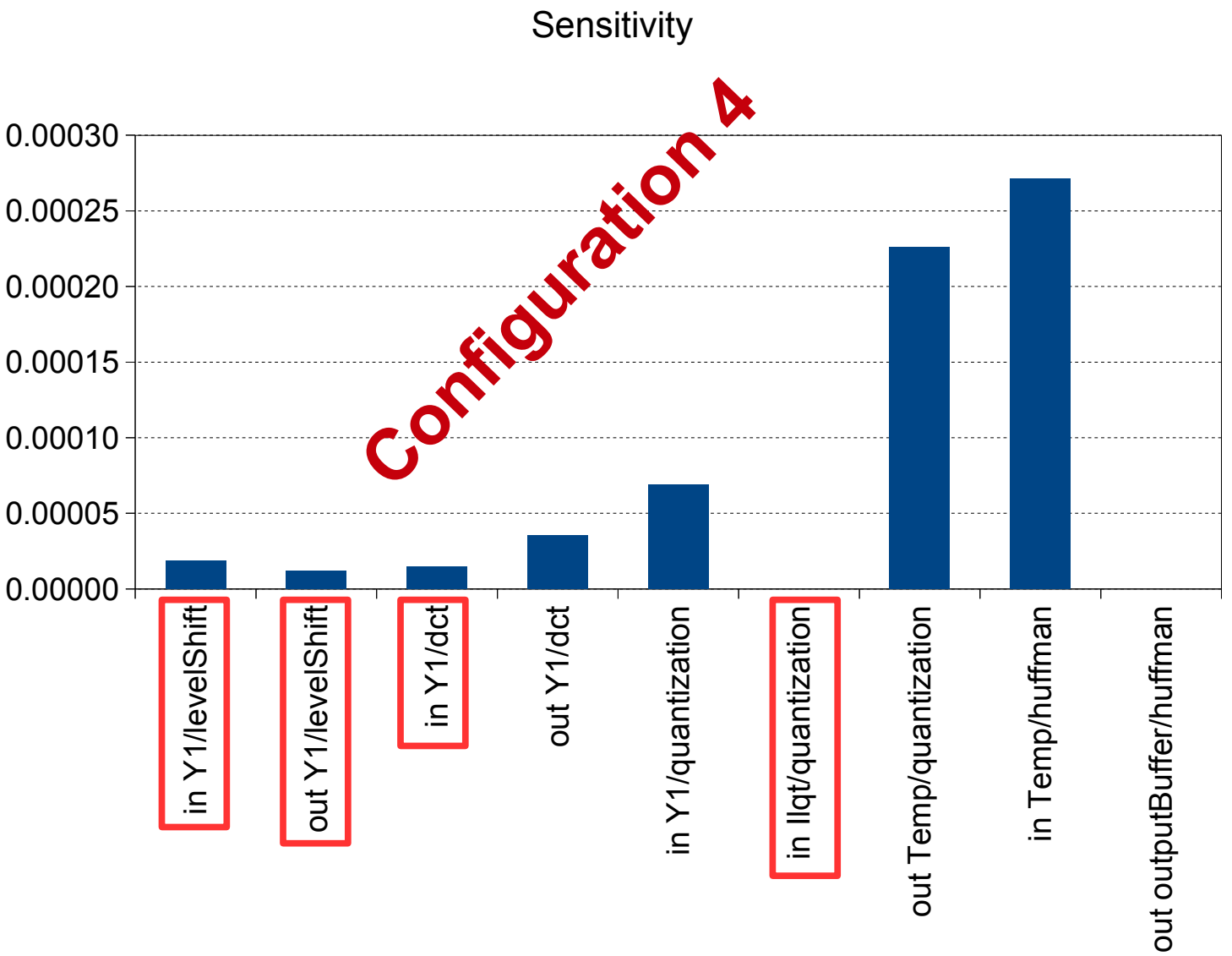
Sensitivity Analysis



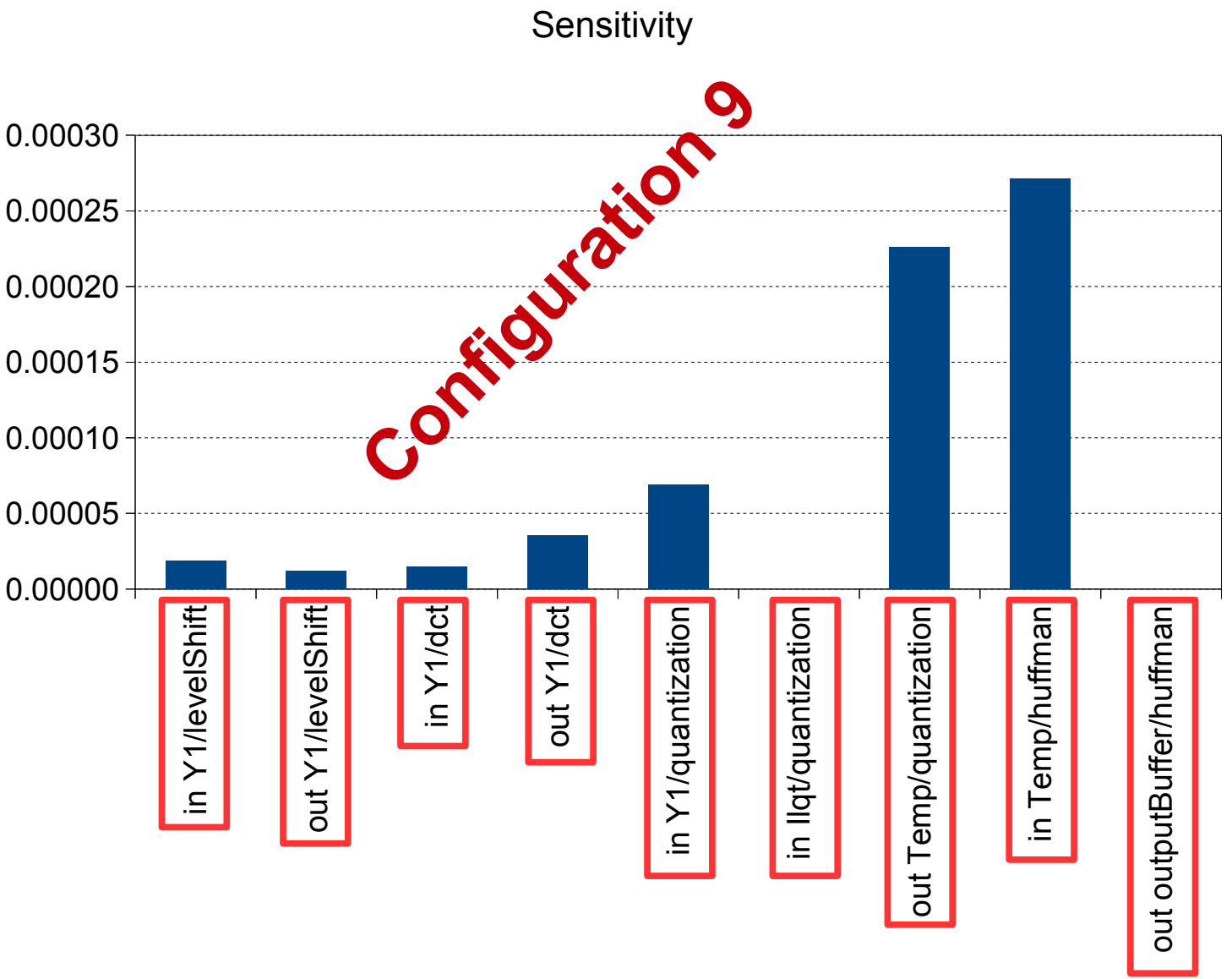
Sensitivity Analysis



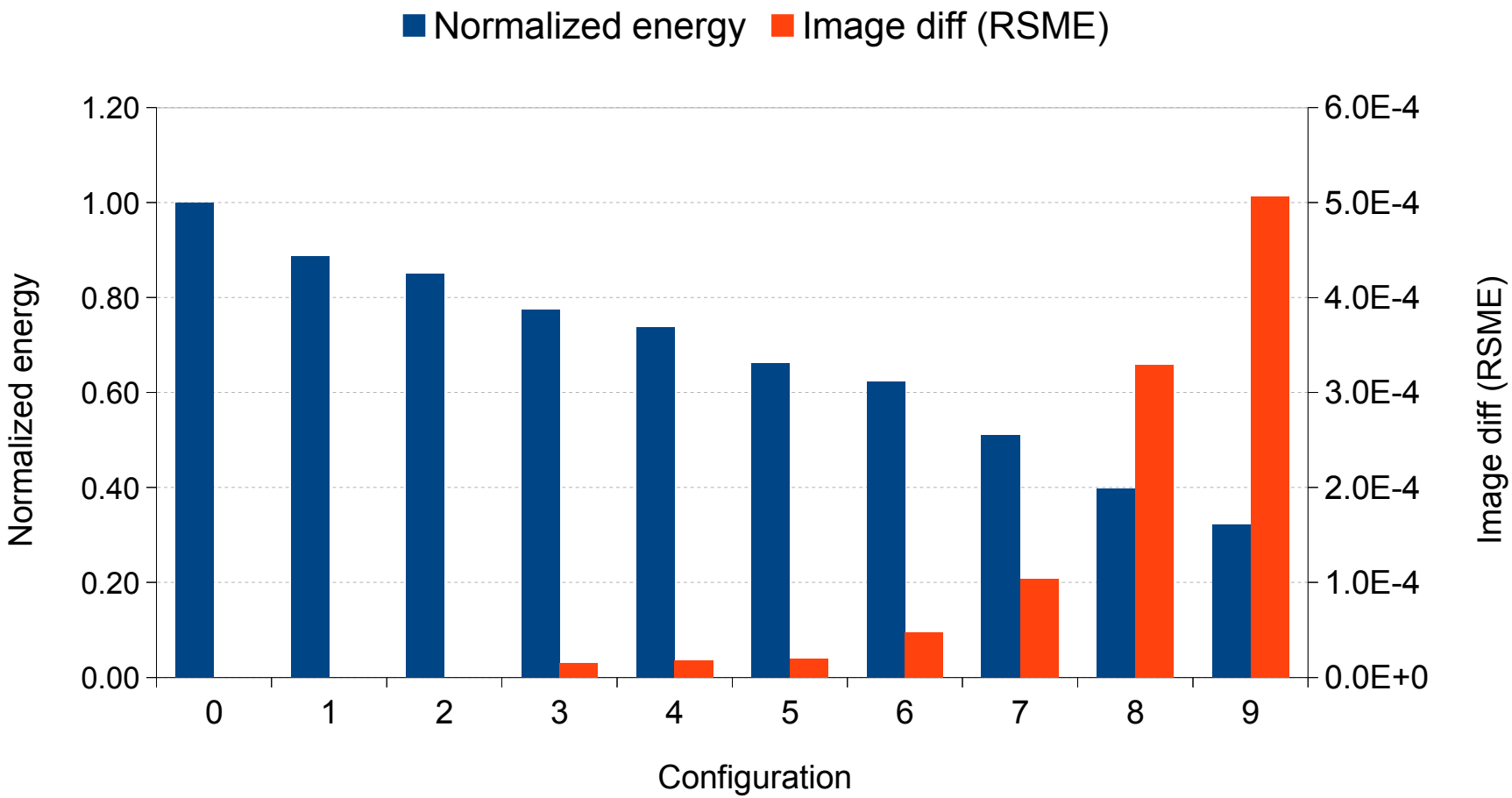
Sensitivity Analysis



Sensitivity Analysis



Exploration



Conclusions

- Extention of the approximate computing paradigm to the communication sub-system
- Application to a simple case study
- Current work
  - Assessment on other RMS applications
  - Extension to emerging WiNoC architectures

Approximate Communication in Networks-on-Chip based Architectures

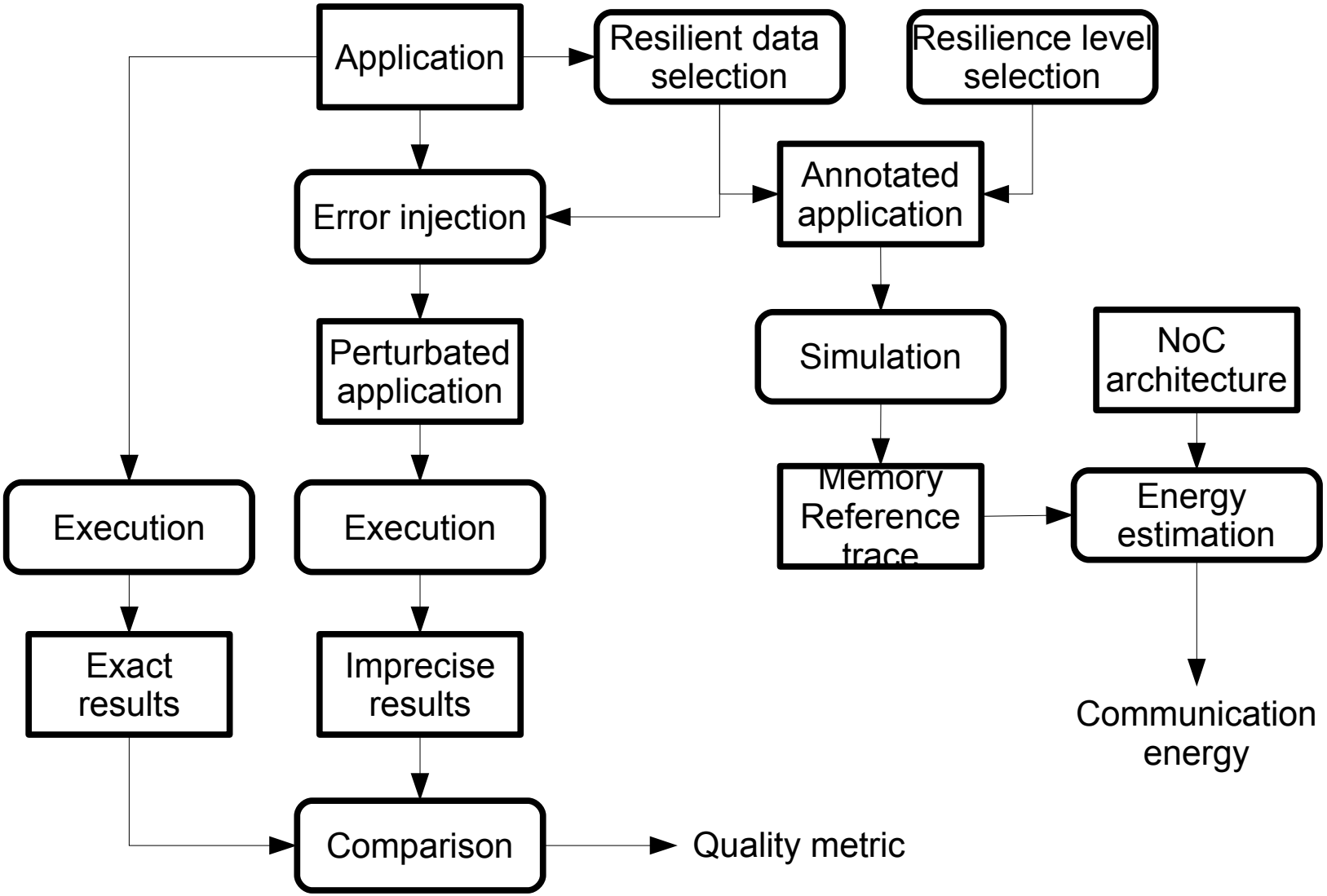
Giuseppe Ascia, Vincenzo Catania, Salvatore Monteleone,
Maurizio Palesi, Davide Patti

DIEEI, University of Catania

3rd Italian Workshop on Embedded Systems
Department of Information Engineering and Mathematics
University of Siena
September 13-14, 2018

Backup Slides

Overall Evaluation Flow

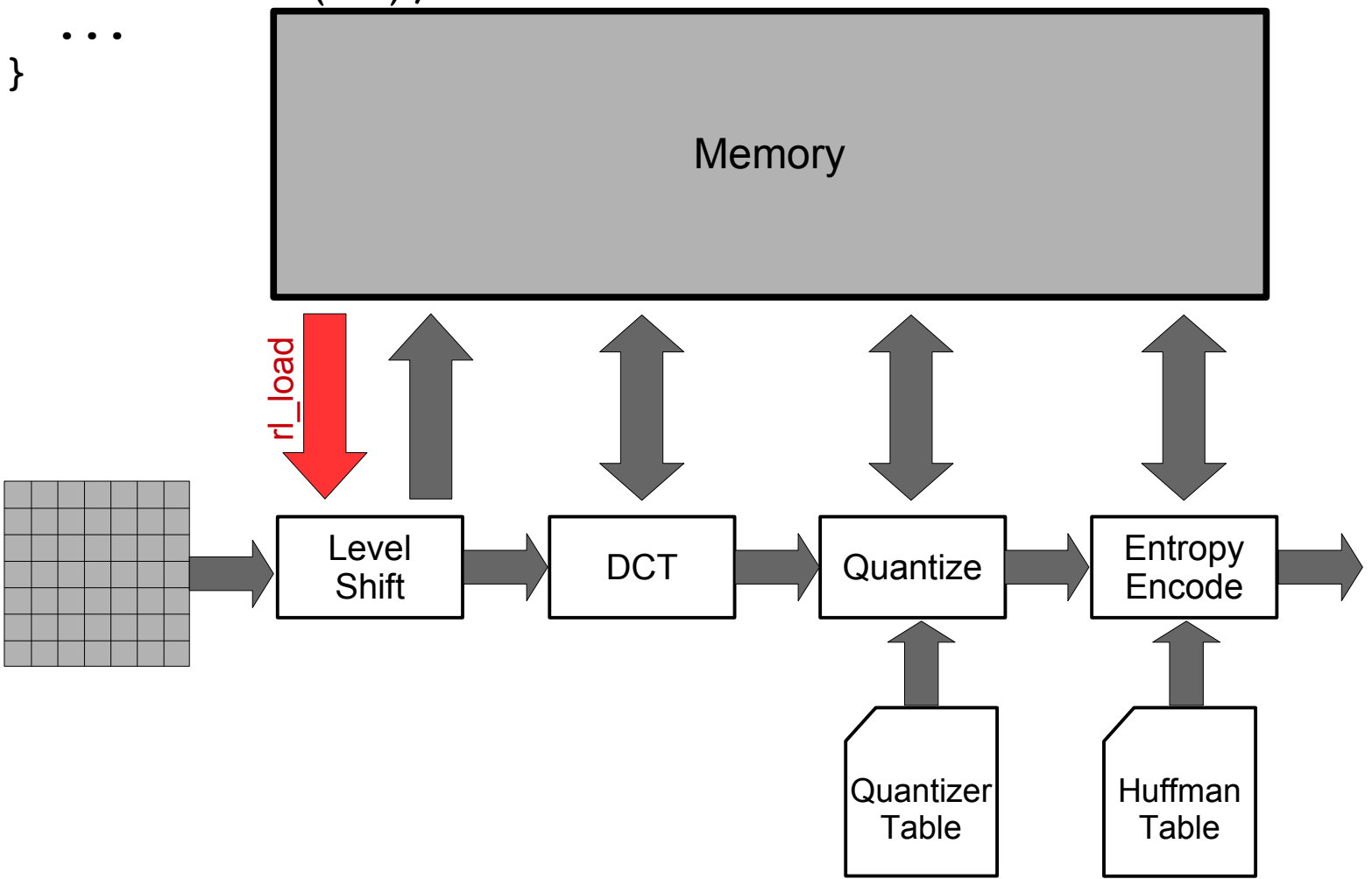


Experiments

```
UINT8* encodeMcu(UINT32 imageFormat,
                 UINT8 *outputBuffer)
{
  levelShift(Y1);
  dct(Y1);
  quantization(Y1, ILqt);
  outputBuffer = huffman(1, outputBuffer);
  return outputBuffer;
}
```

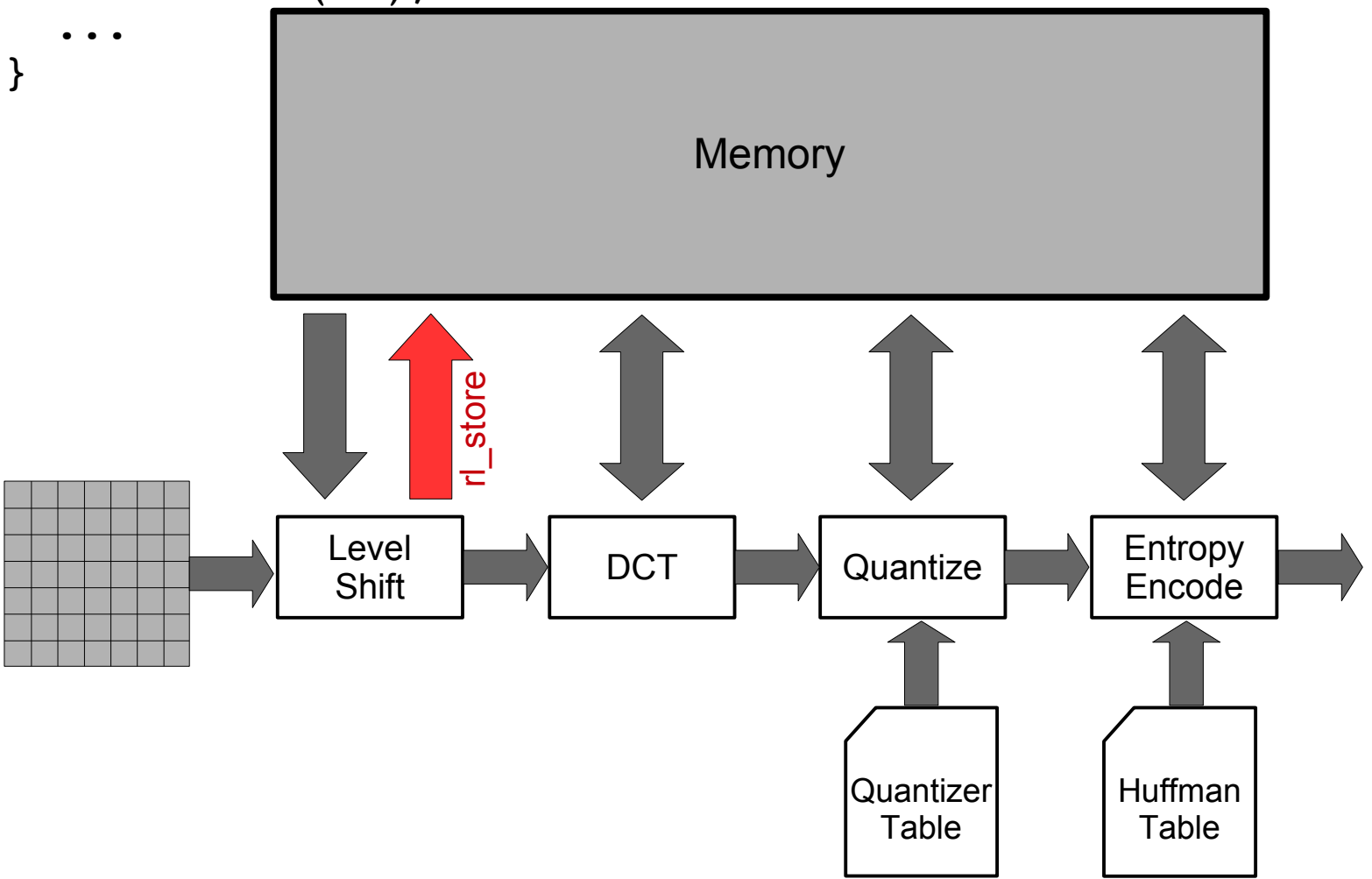
Experiments

```
UINT8* encodeMcu(UINT32 imageFormat,
                 UINT8 *outputBuffer)
{
  #pragma resilient_load(Y1, rl_load)
  levelShift(Y1);
  ...
}
```



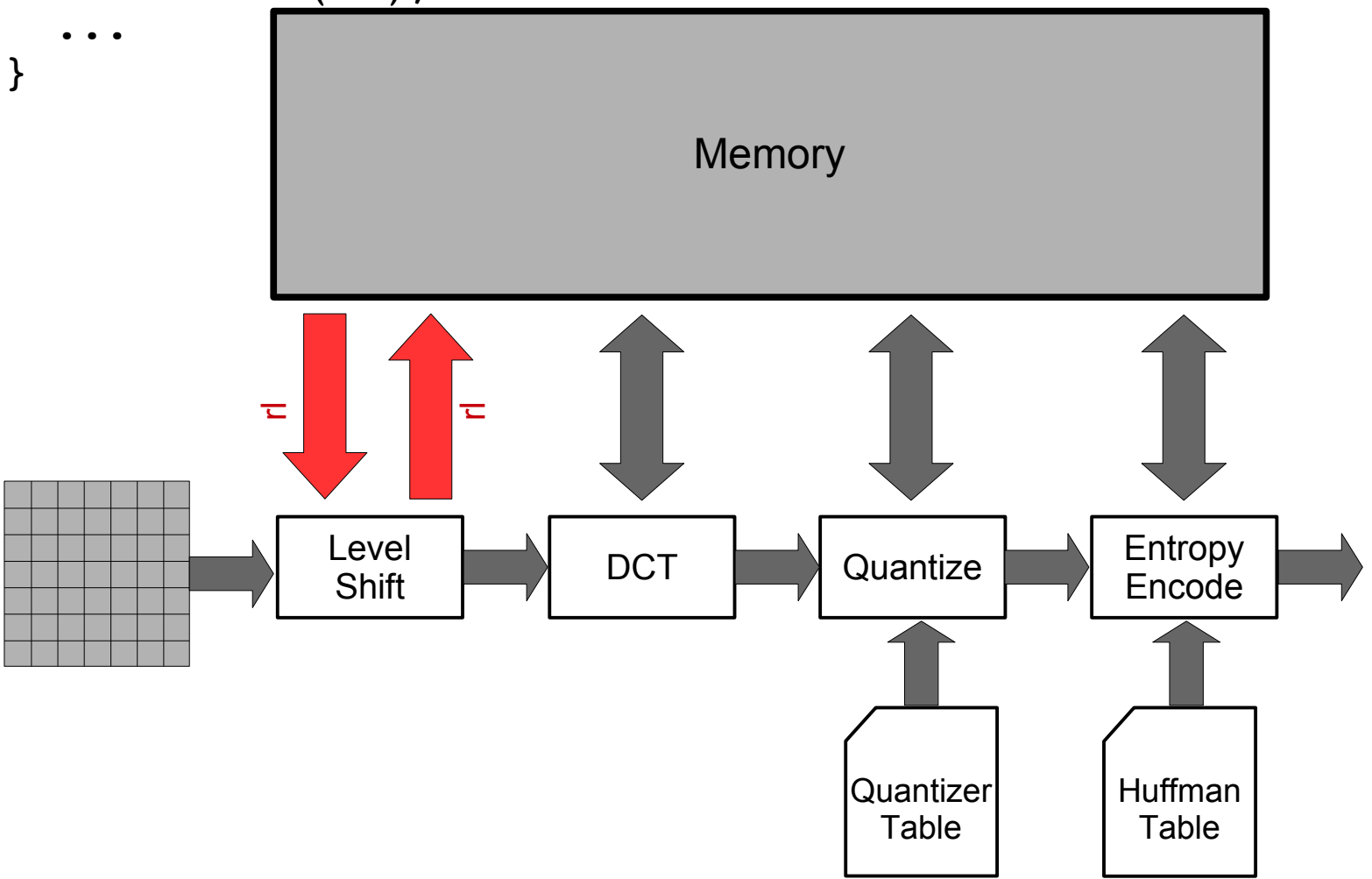
Experiments

```
UINT8* encodeMcu(UINT32 imageFormat,
                 UINT8 *outputBuffer)
{
  #pragma resilient_store(Y1, rl_store)
  levelShift(Y1);
  ...
}
```



Experiments

```
UINT8* encodeMcu(UINT32 imageFormat,
                 UINT8 *outputBuffer)
{
  #pragma resilient(Y1, rl)
  levelShift(Y1);
  ...
}
```

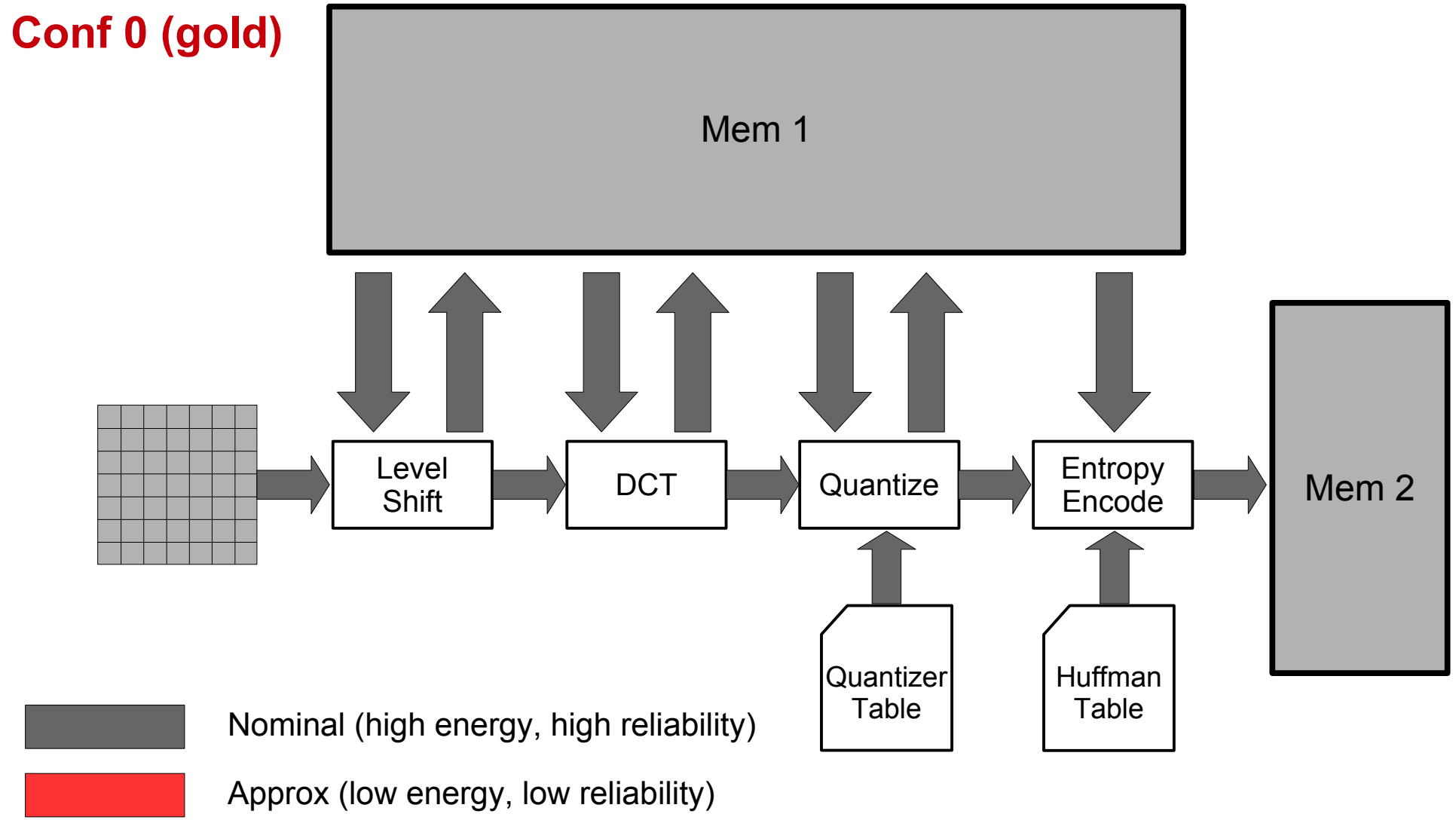


Experiments

- Two voltage swing levels
  - Nominal 1.1 V → BER: 10^{-17}, Ebit: 512 fJ
  - Low 0.6 V → BER: 10^{-6}, Ebit: 152 fJ

Experiments

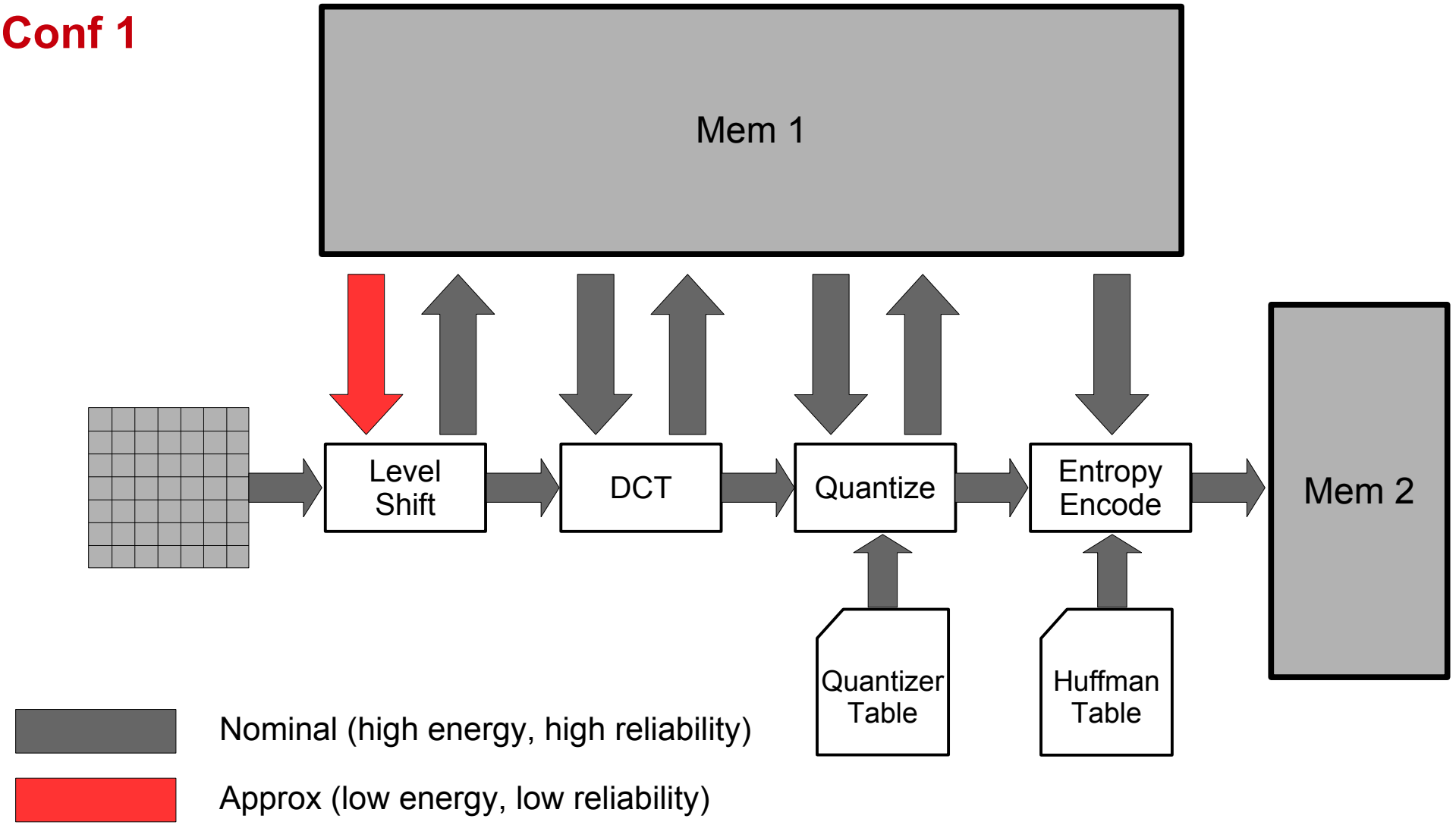
Conf 0 (gold)



Experiments

Experiments

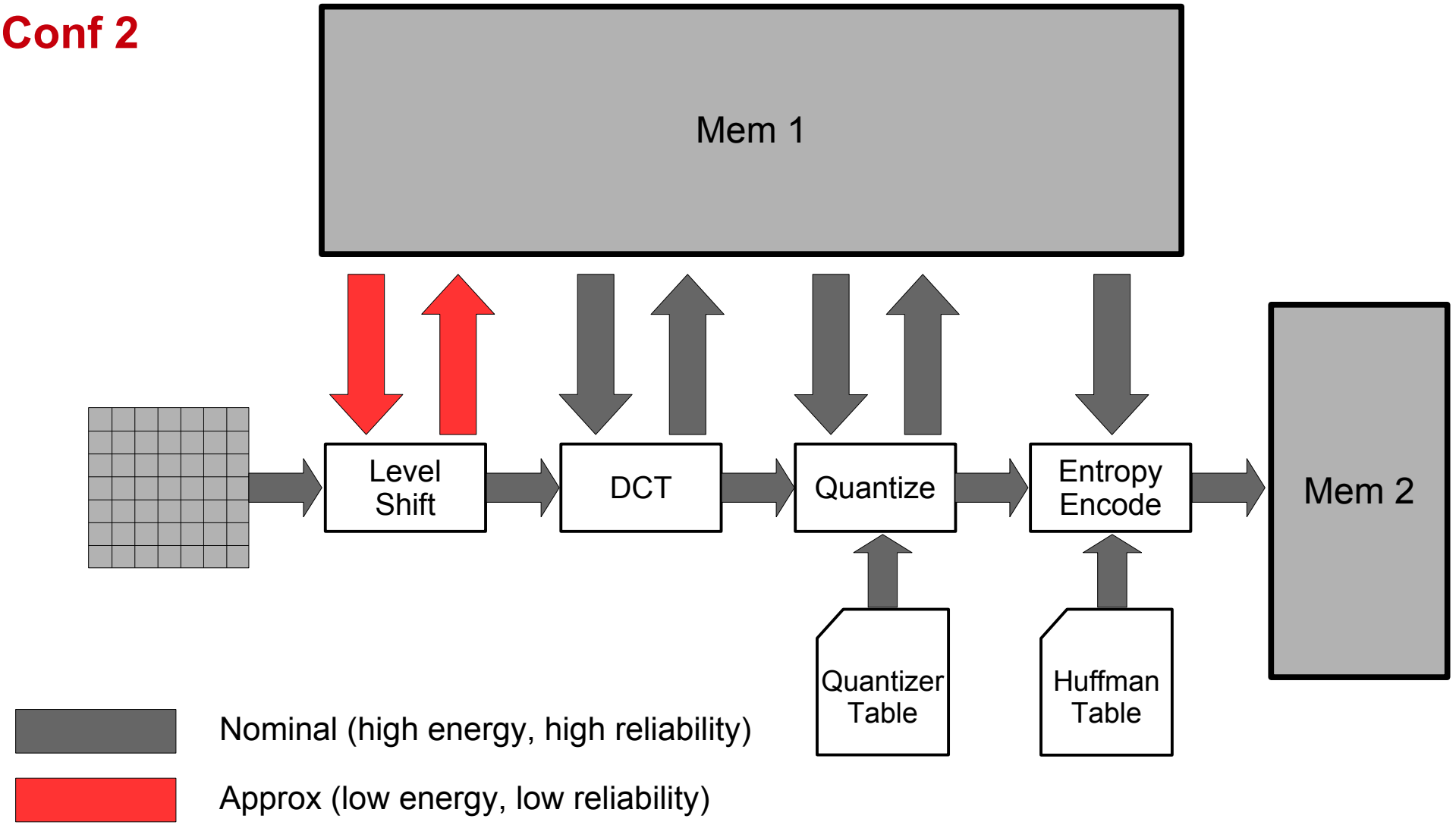
Conf 1



Experiments

Experiments

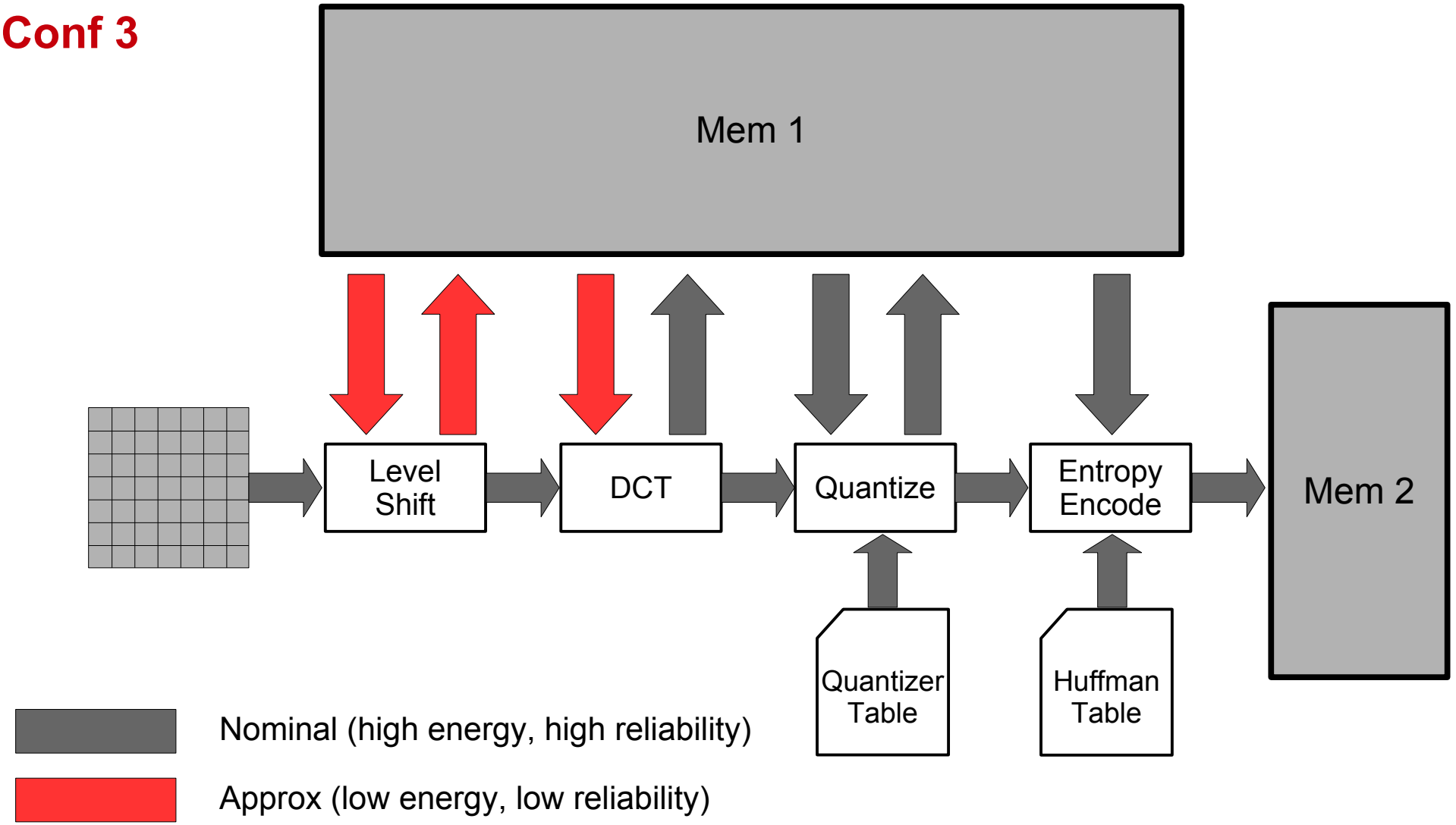
Conf 2



Experiments

Experiments

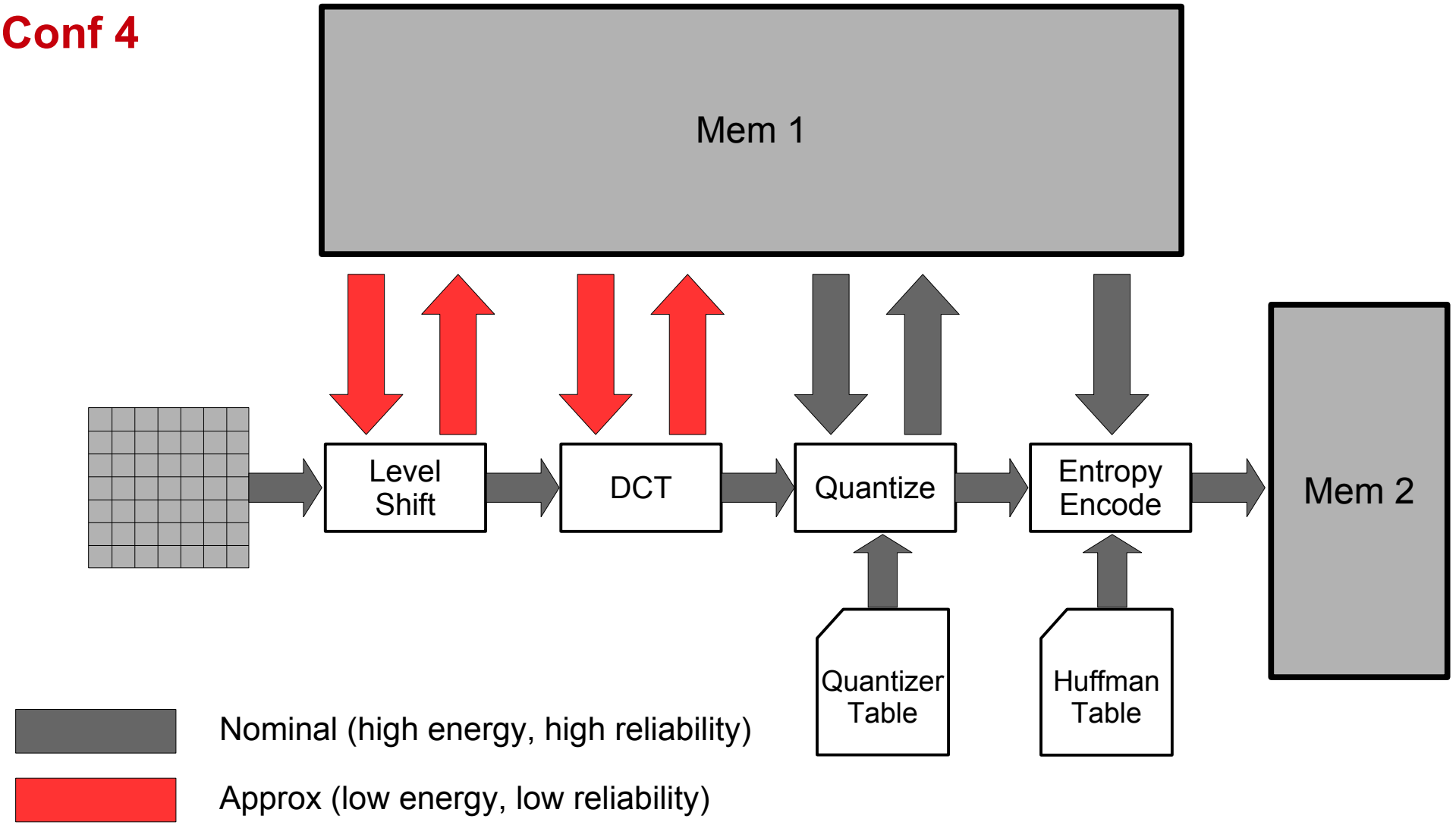
Conf 3



Experiments

Experiments

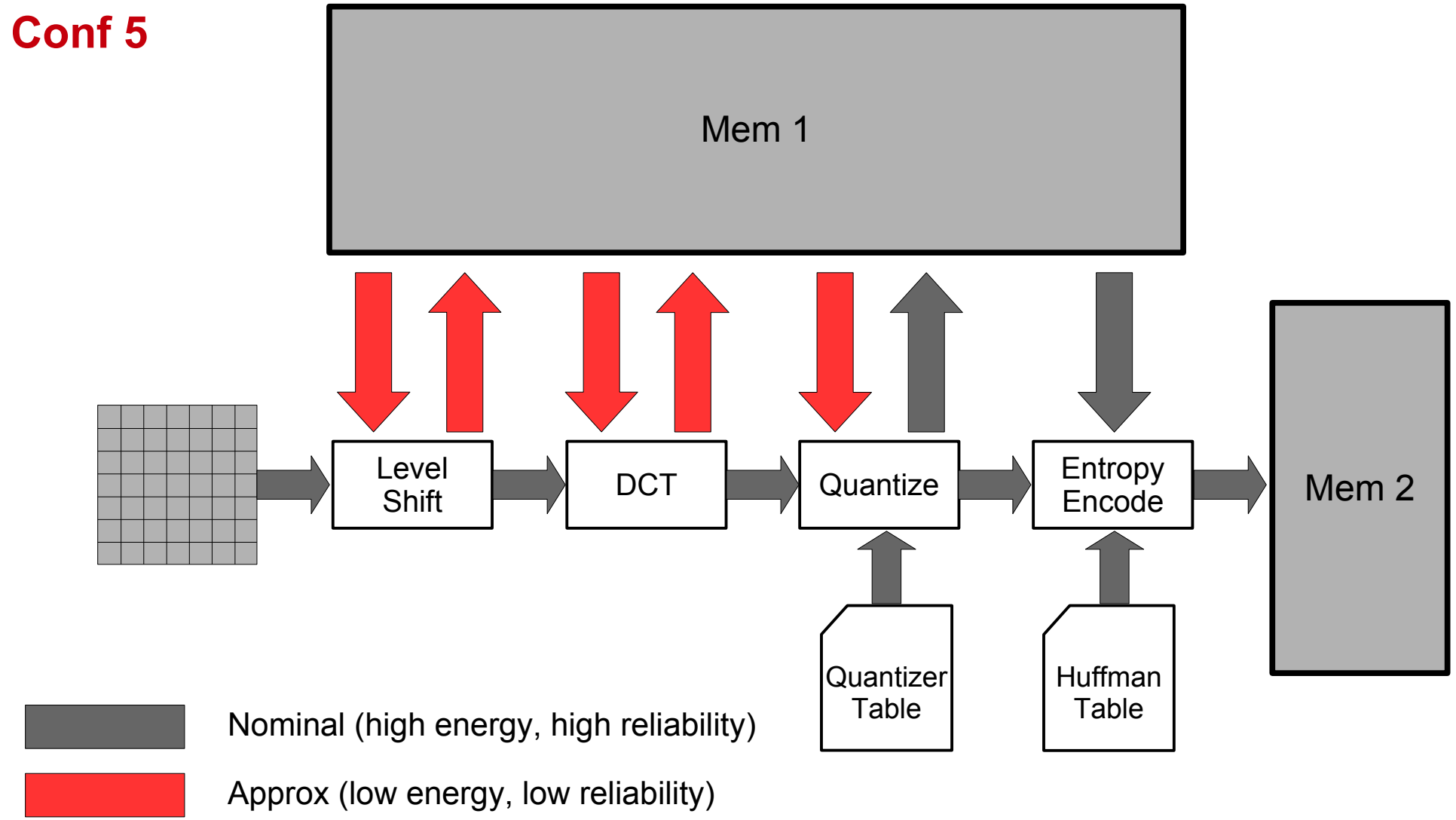
Conf 4



Experiments

Experiments

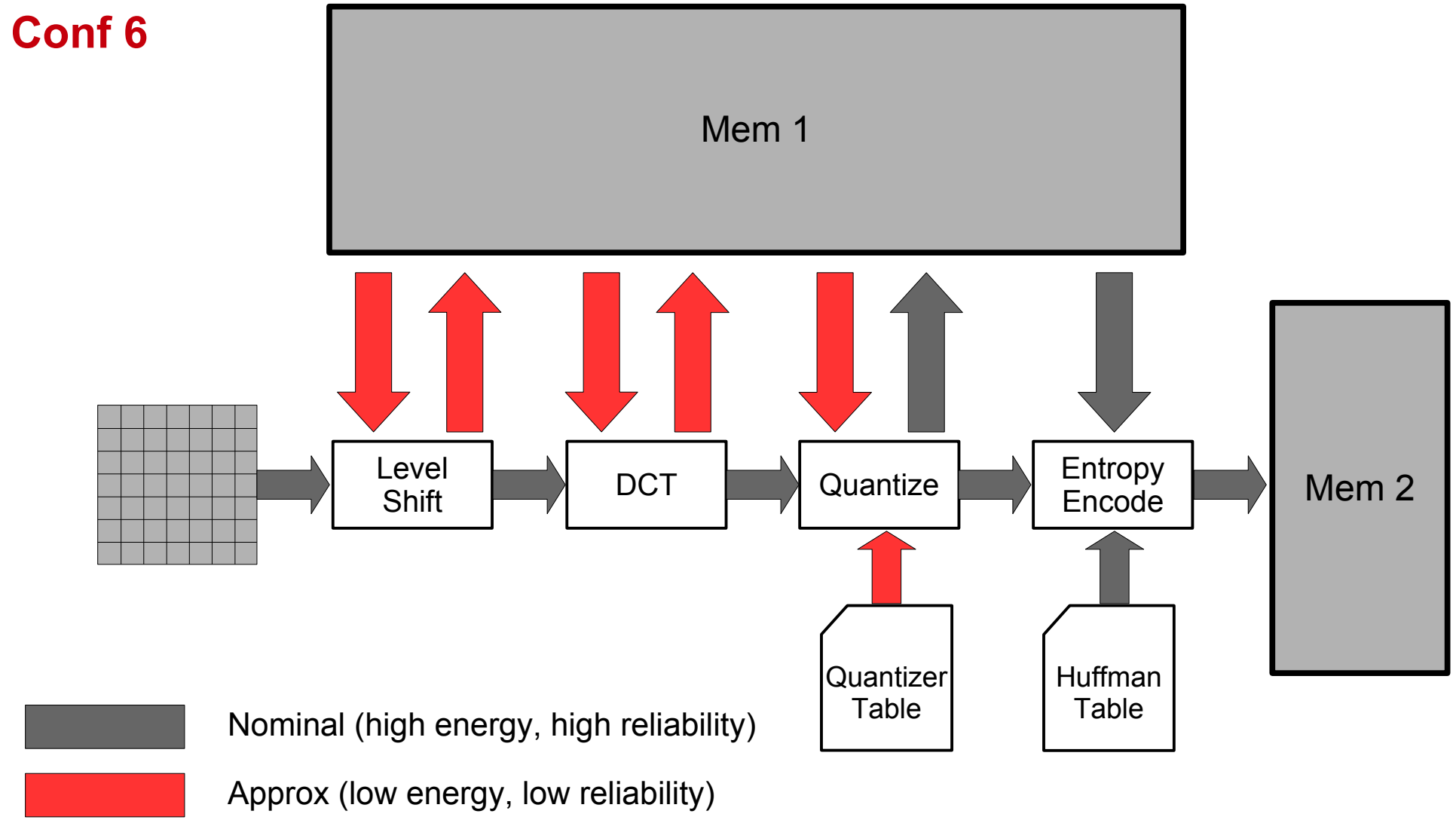
Conf 5



Experiments

Experiments

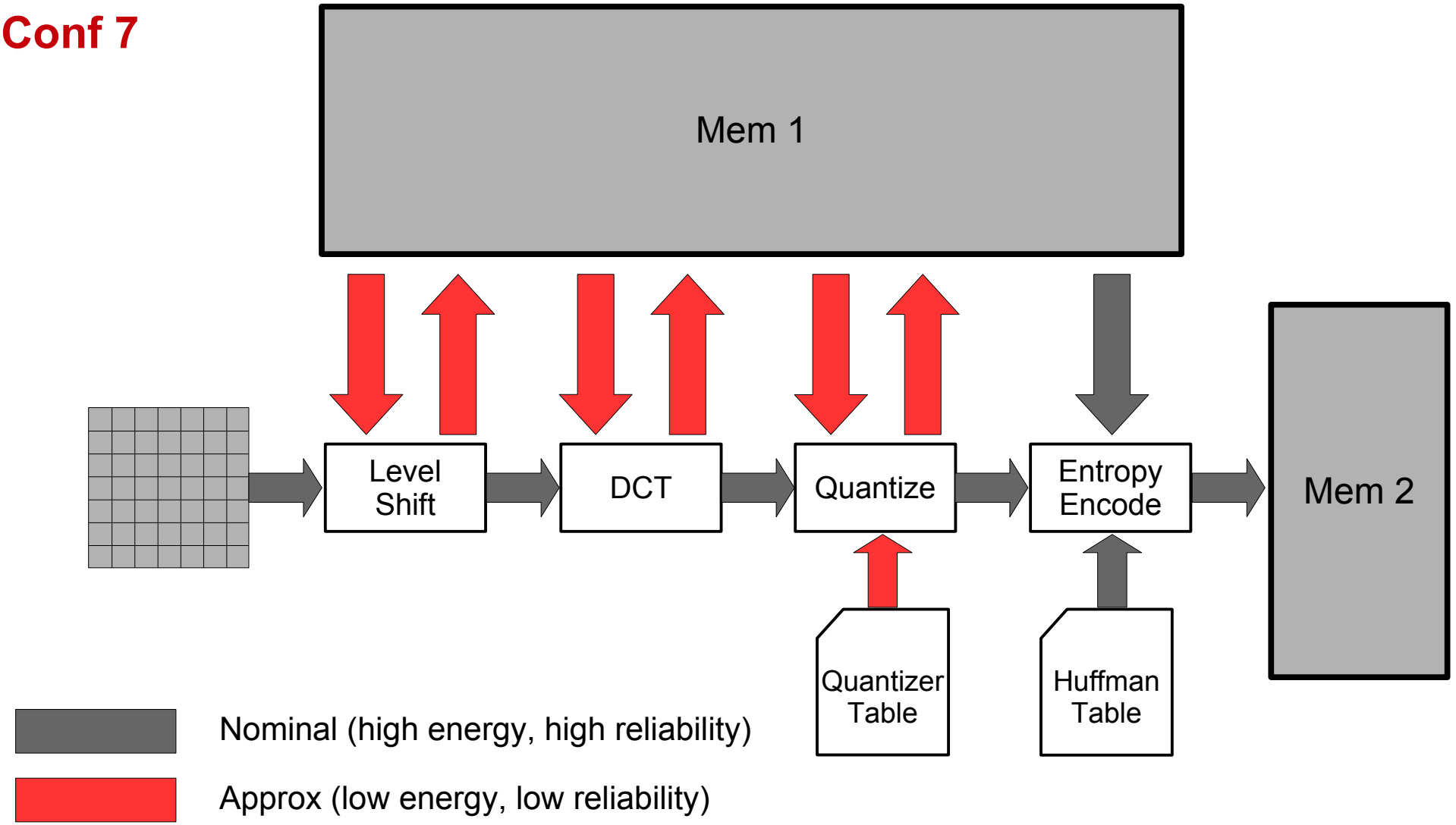
Conf 6



Experiments

Experiments

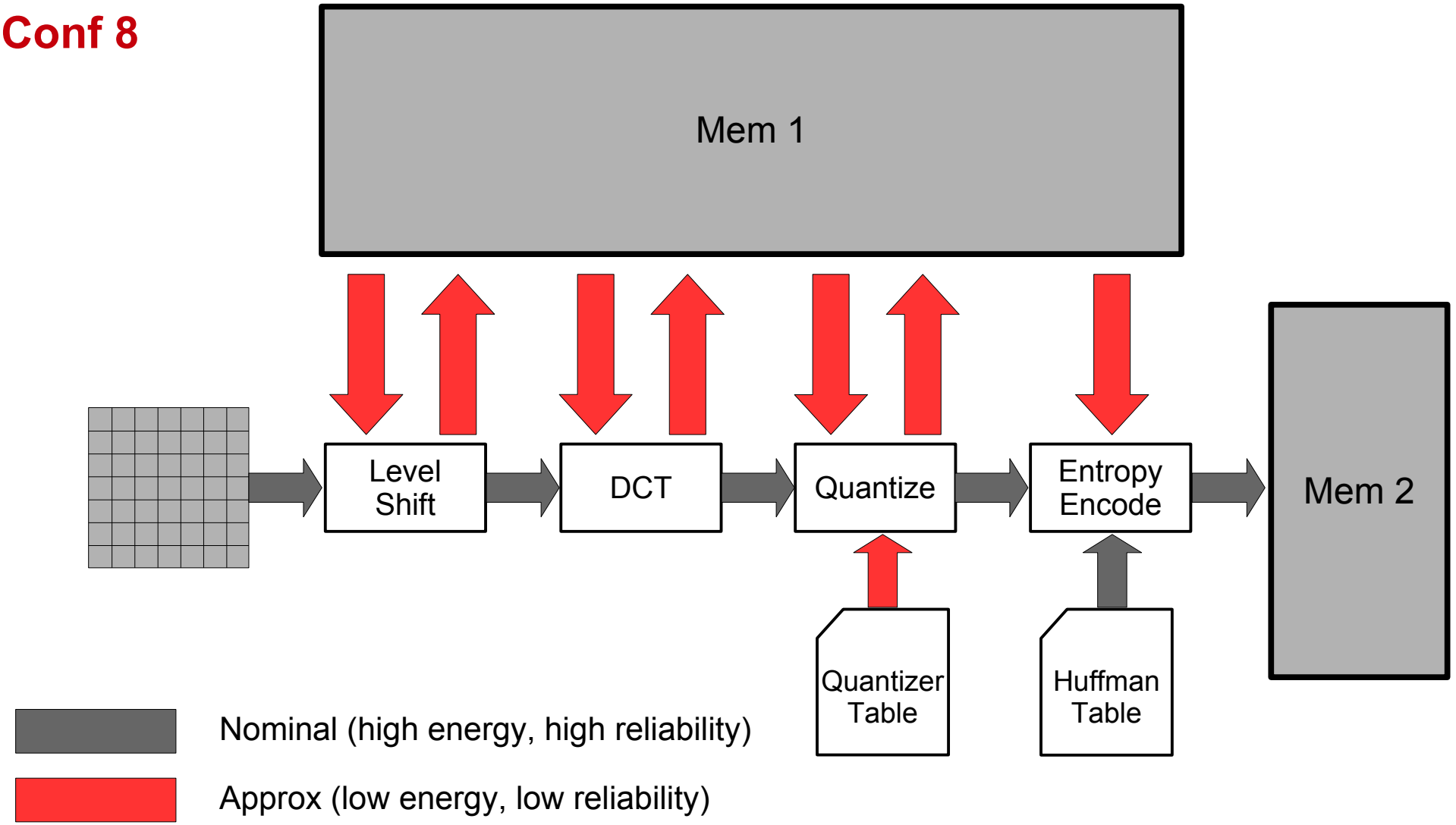
Conf 7



Experiments

Experiments

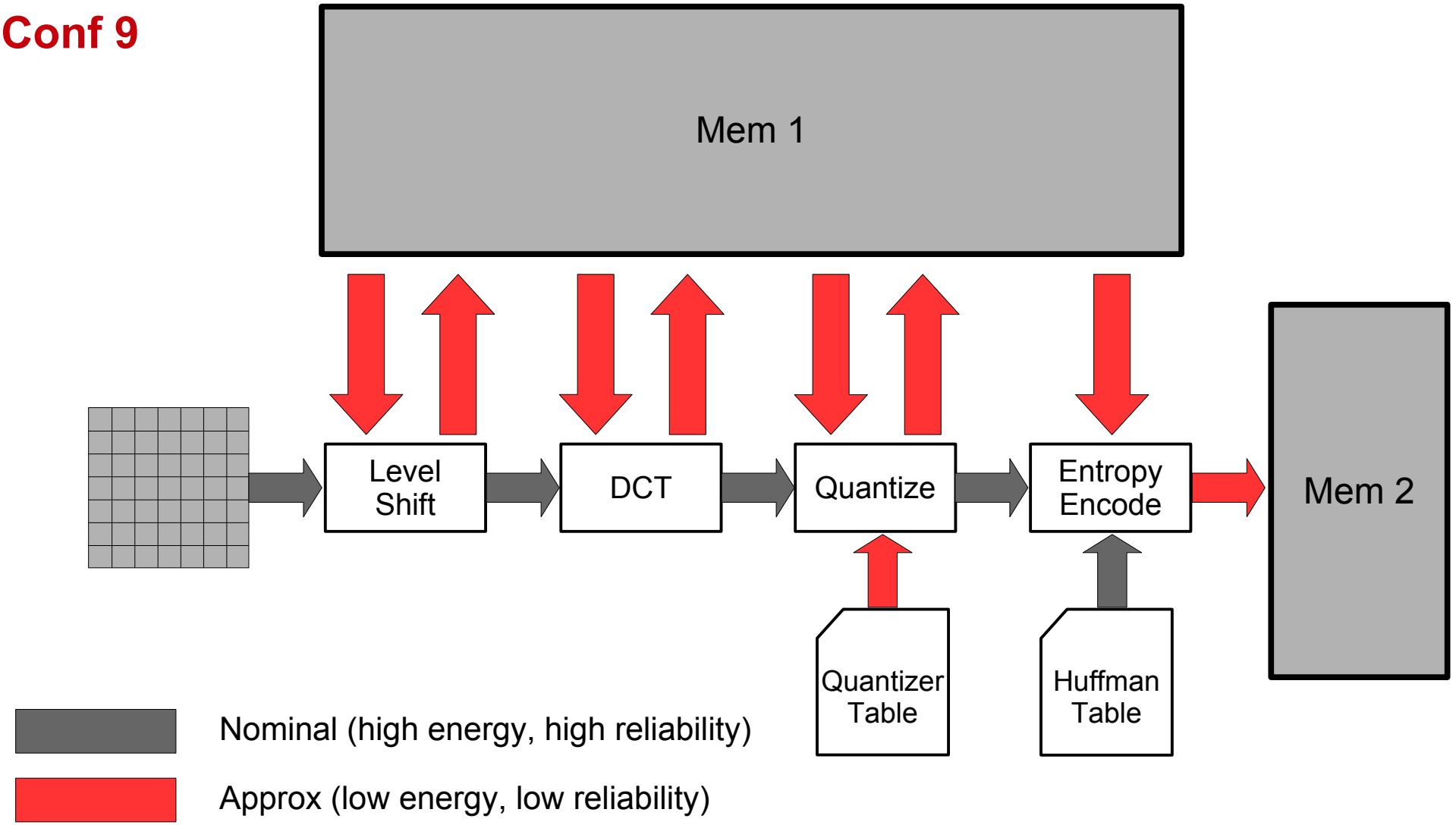
Conf 8



Experiments

Experiments

Conf 9



Experiments