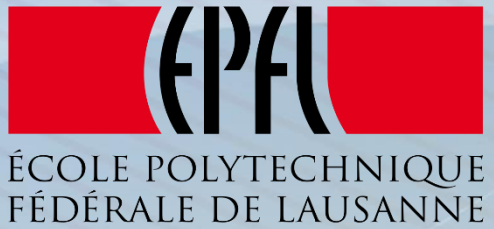
High-Performance Correlation Coefficient Calculator

Marcos Vinicius Silva Oliveira
Nikita Lazarev
Robin Malzac

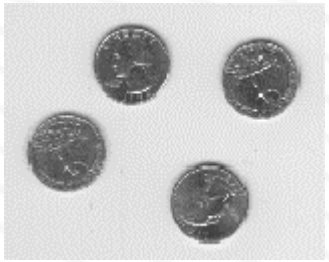


Summary

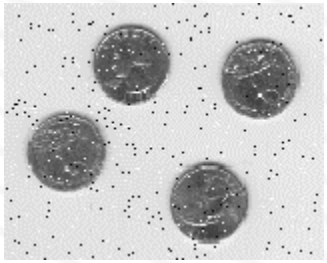
- Context
- Applications
- Impmementation
- Verification
- Performance analysis
- Conclusions

Applications

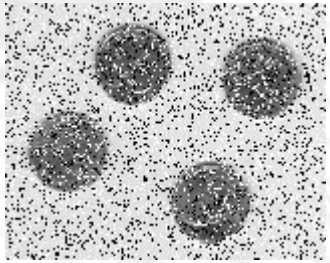
- Radar system
- Detect water leak
- Web filtering
- Risk management
- Telescope image analysis



Original
r = 1

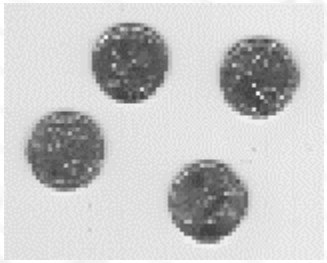


S & P (0.02)
r = 0.908

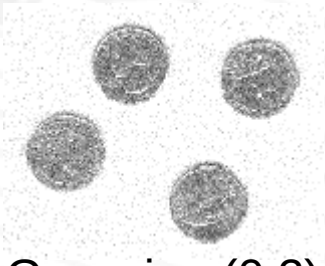


S & P (0.2)
r = 0.521

Flip
r = - 0.057



Resolution / 2
r = 0.956



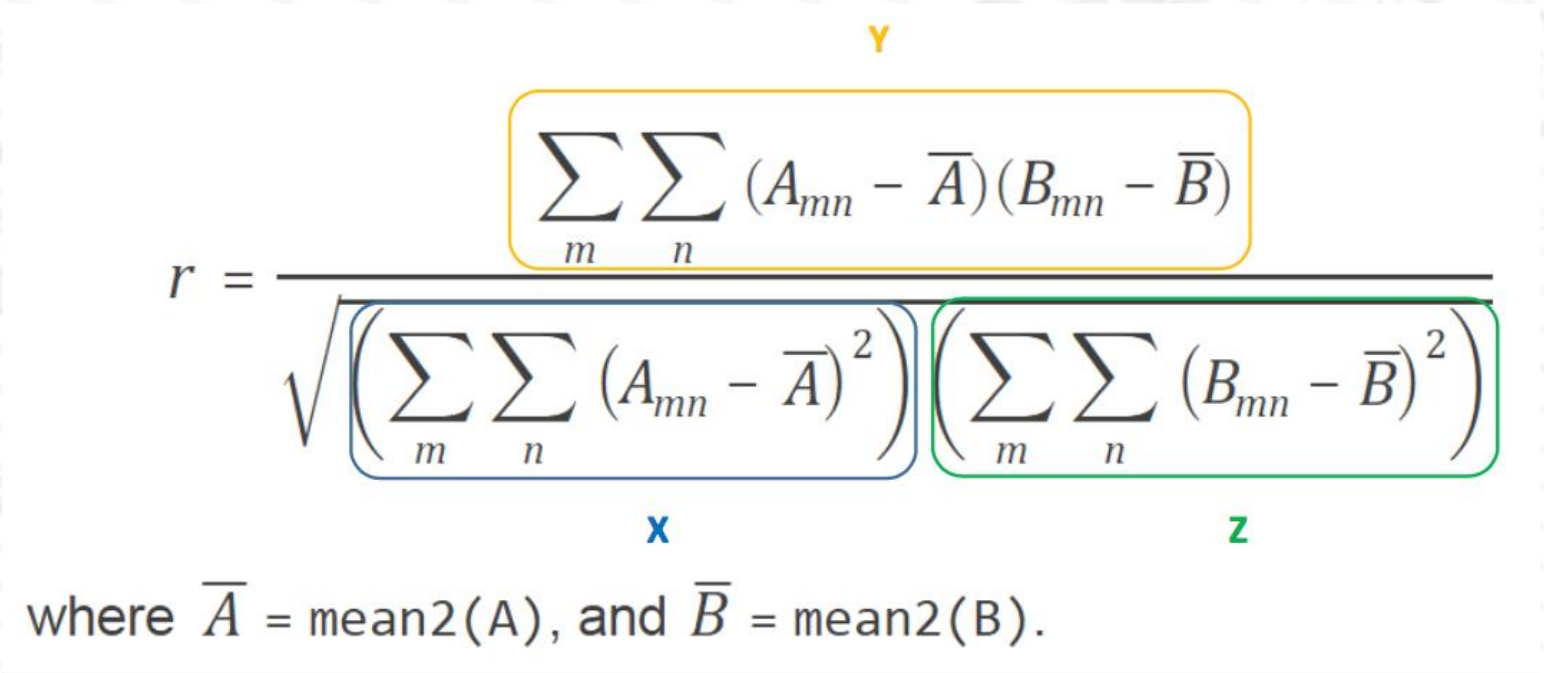
Gaussian (0.2)
r = 0.944

r = - 0.014

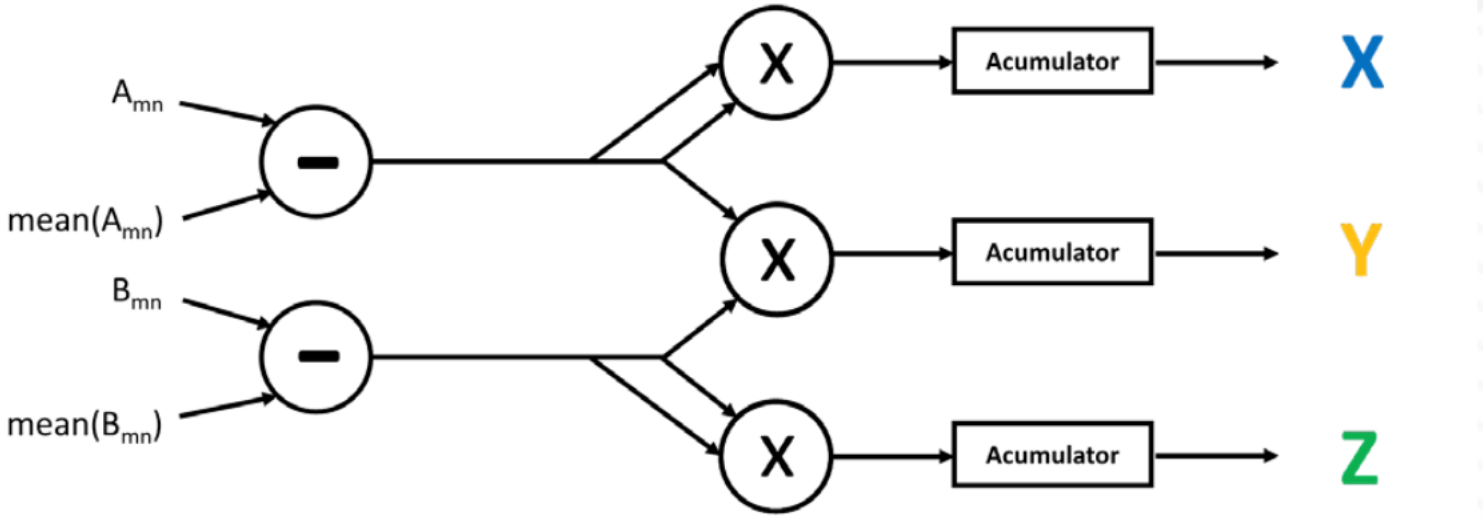
r = 0.063

r = - 0.083

Math & SW vs HW distribution

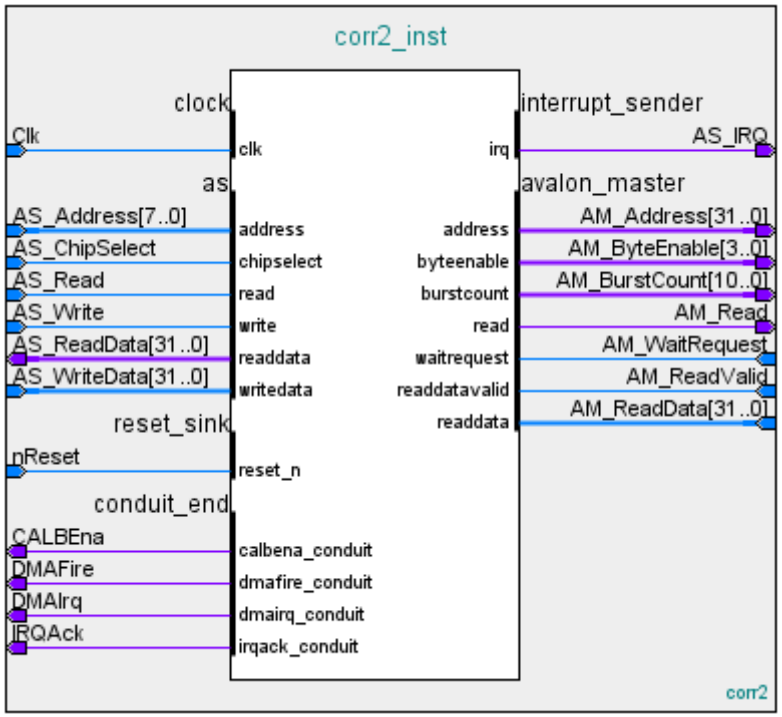


SW vs HW distribution

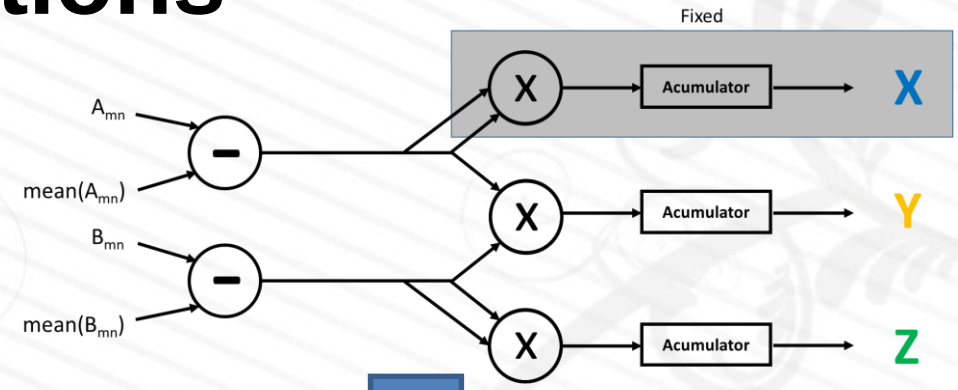


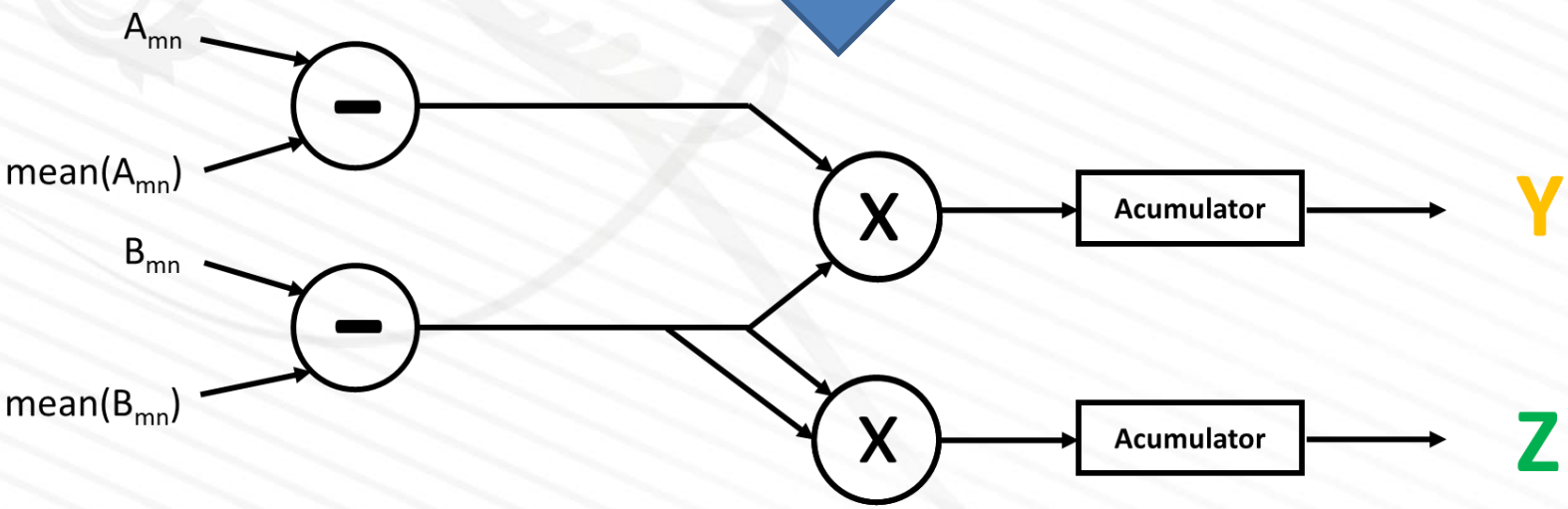
$$r = Y/\mathrm{sqrt}(X*Z)$$

Corr2 IP interfaces

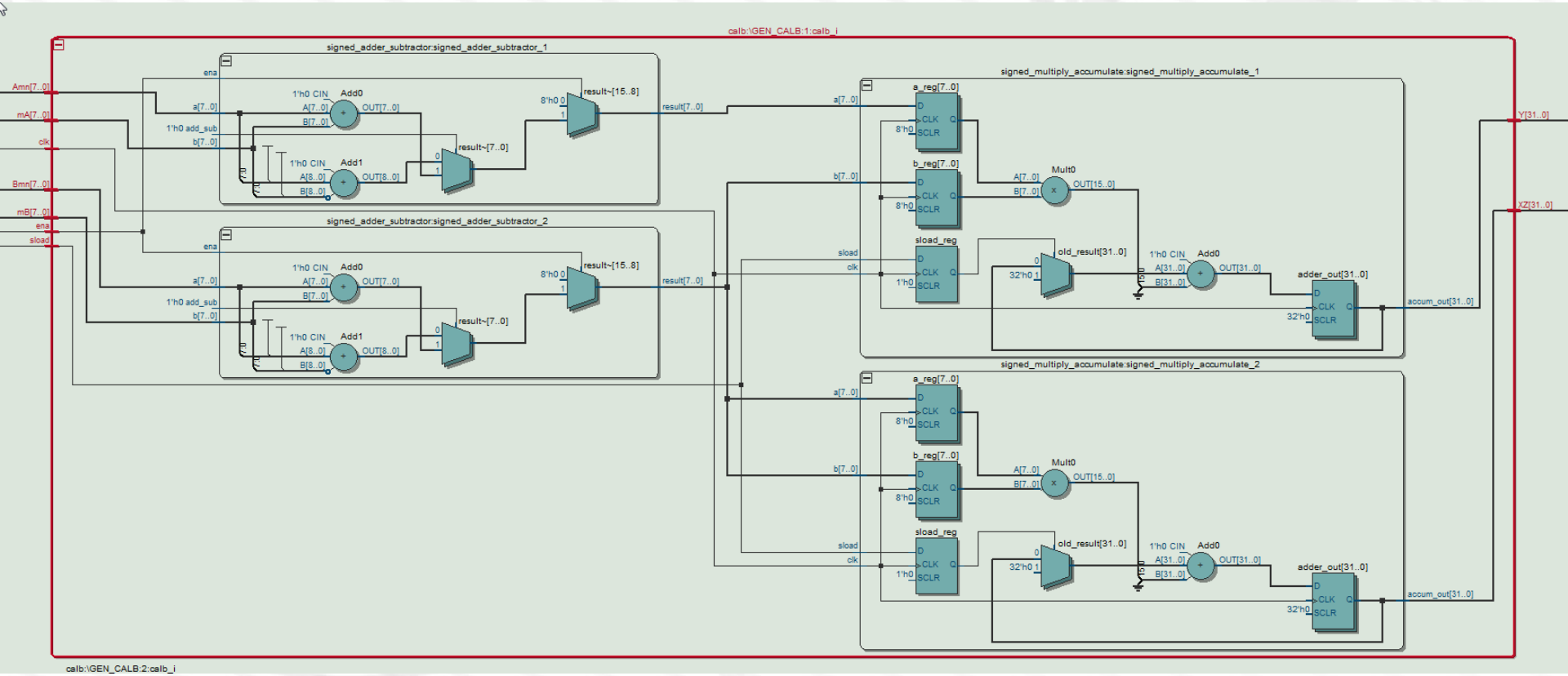


Custom Arithmetic Logic block simplifications



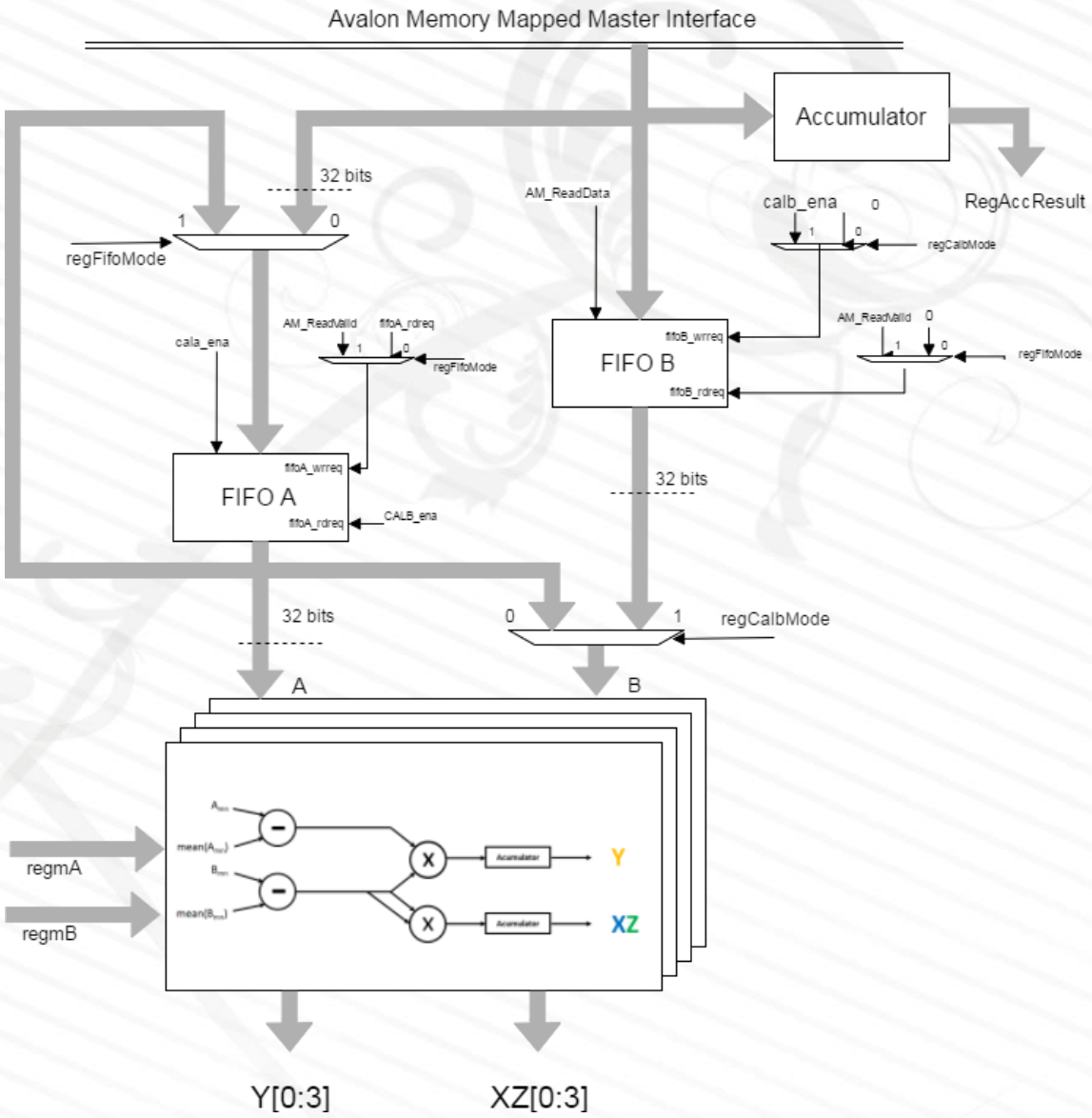


Custom Arithmetic Logic block



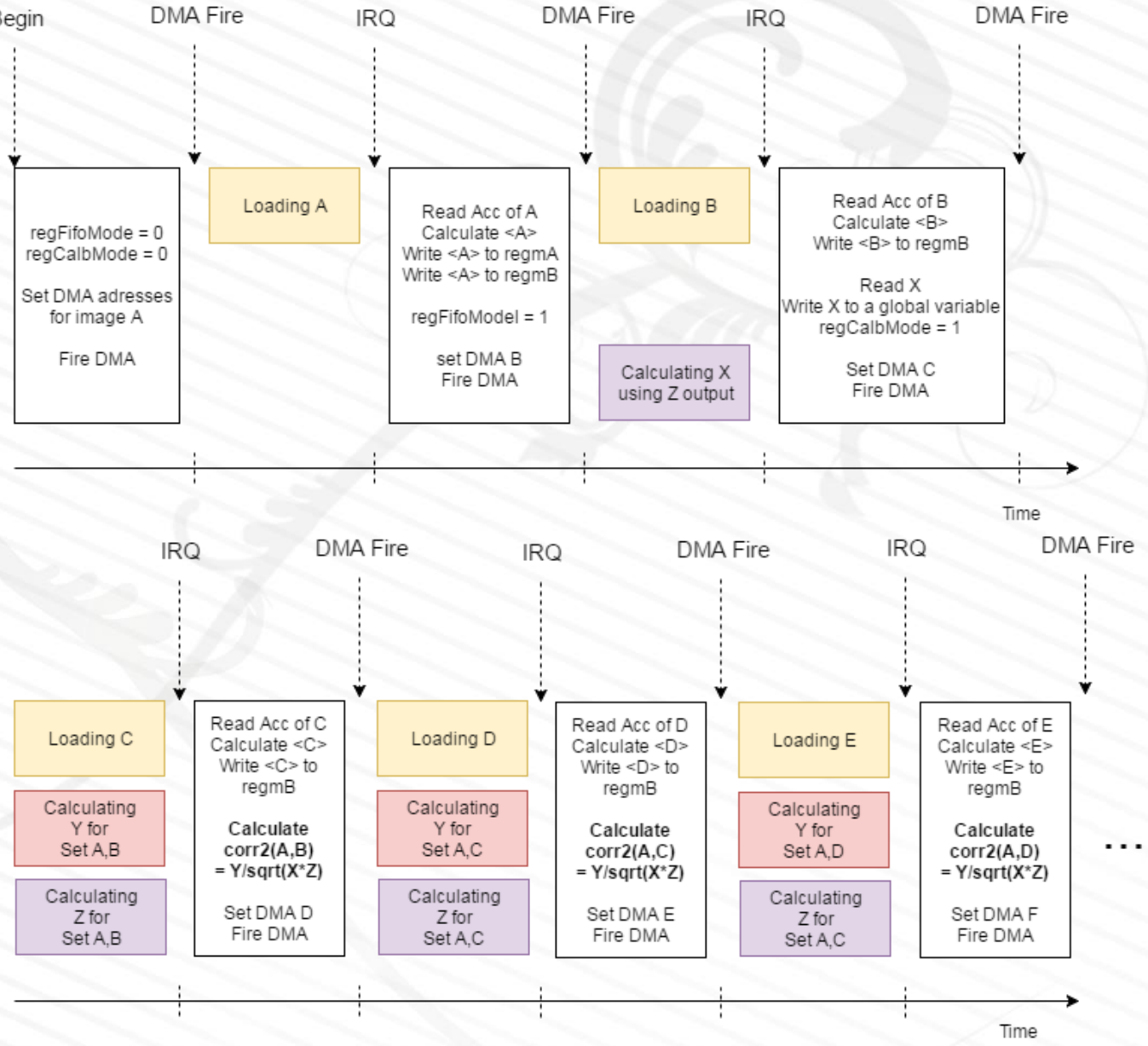
Corr2 IP

- Fifo A receives reference image (A)
  - After is loaded, the feedback path is enabled
- Accumulator calculates the sum of elements of the incoming image
- Fifo B receives image to be compared againt A.
  - The content is stored until all the elements have been summed up
- 4 instances of the CALB block proccess 4 pixels simuntaneously

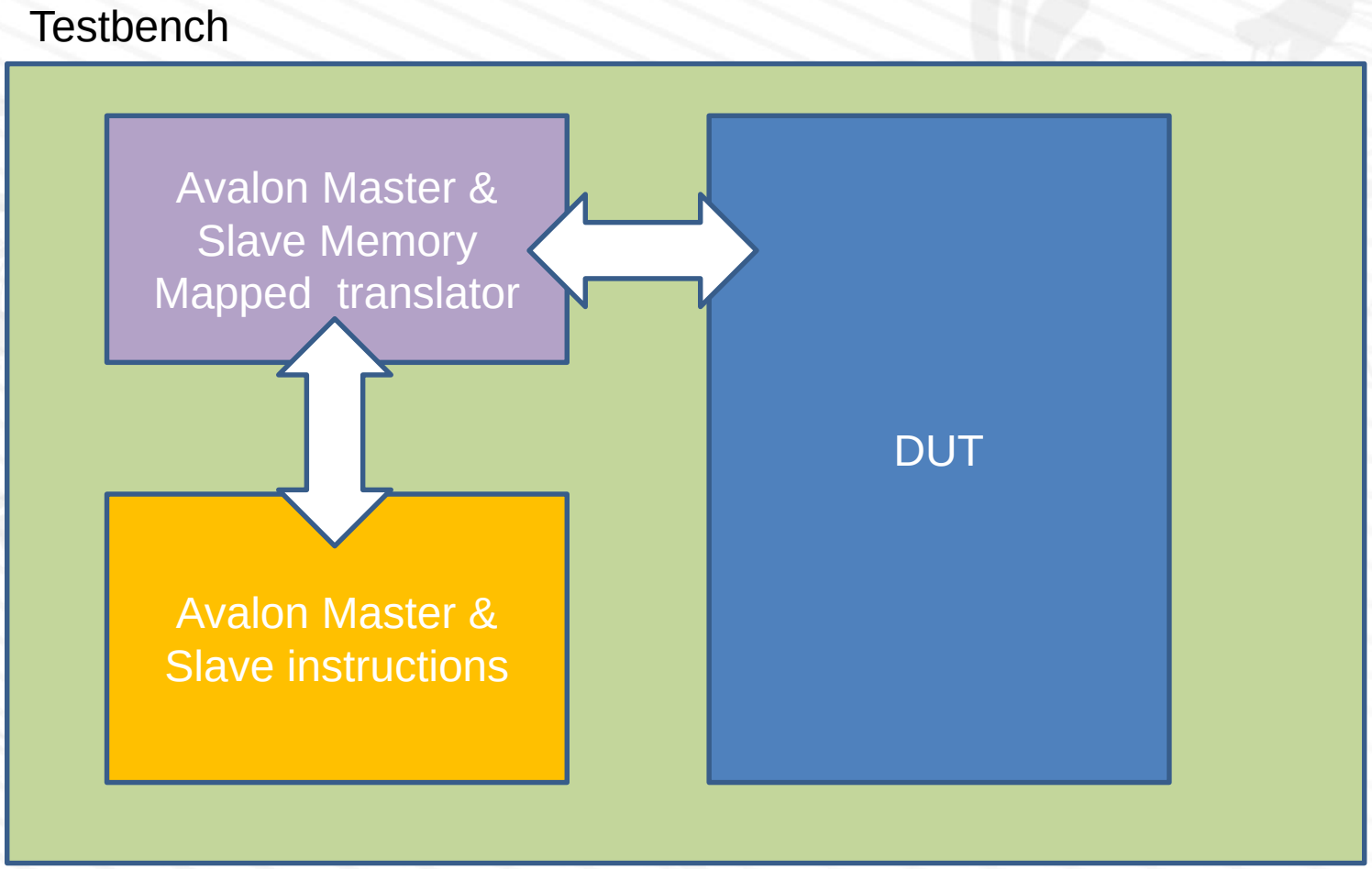


Corr2 IP Data flow timeline

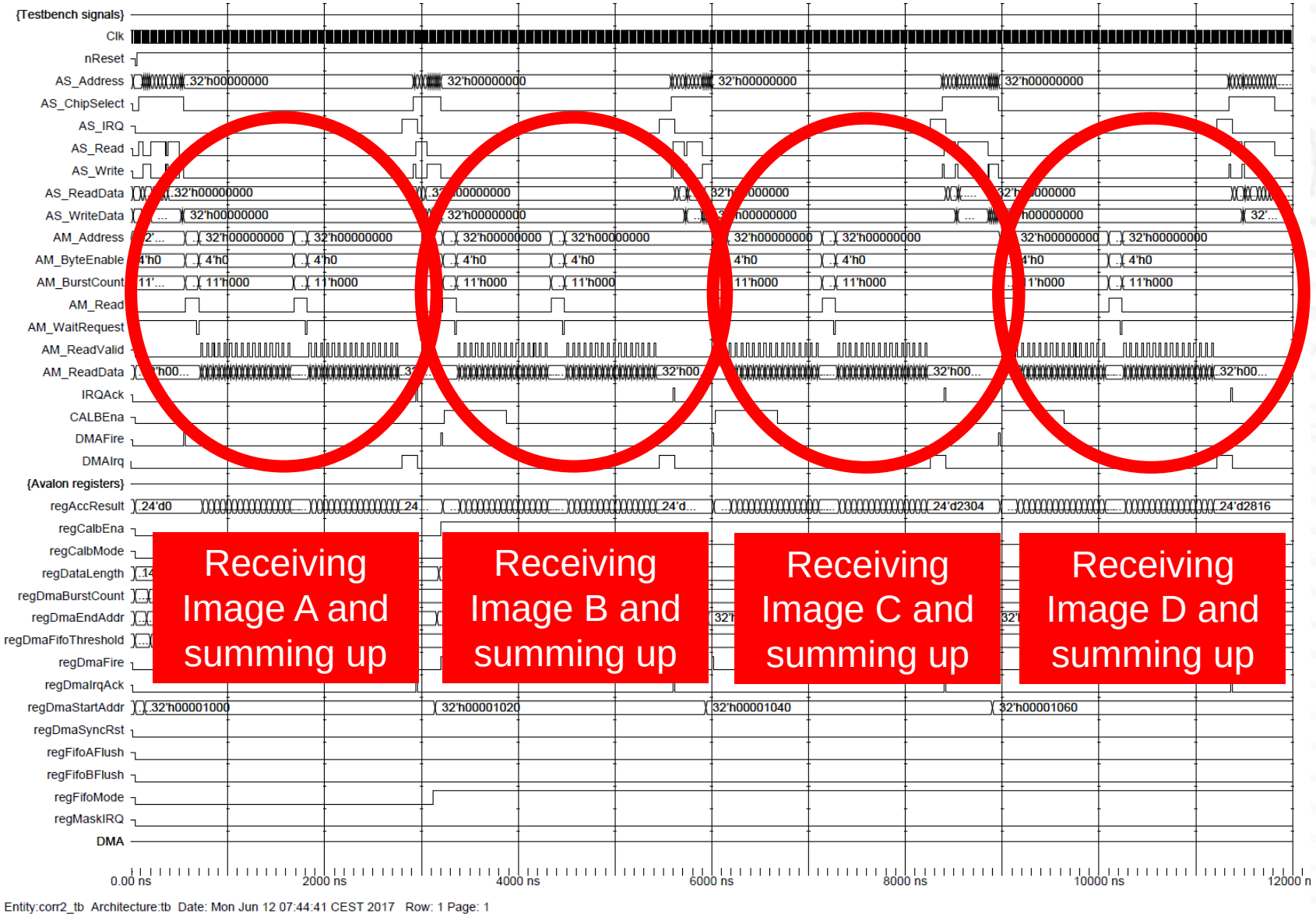
- **Perfomance boost thanks to:**
  - **DMA access to memory**
  - **Reading image A only once for all coeficient calculations**
  - **Loading data, calculating sum, calculating Y, calculating Z, simultaneously**
  - **Processing 4 pixels at the same time**



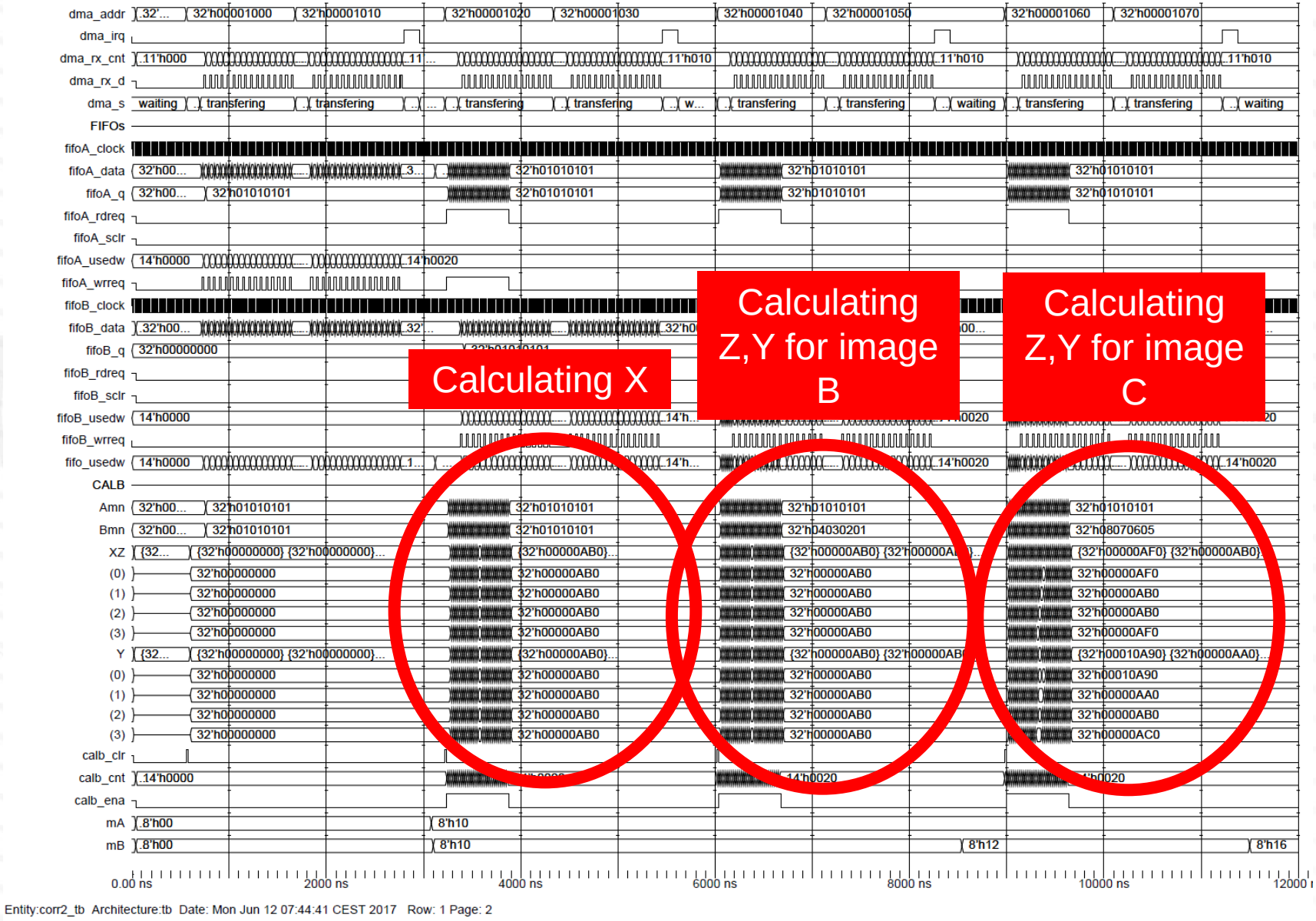
Functional Simulation / Verification



Functional Simulation / Verification

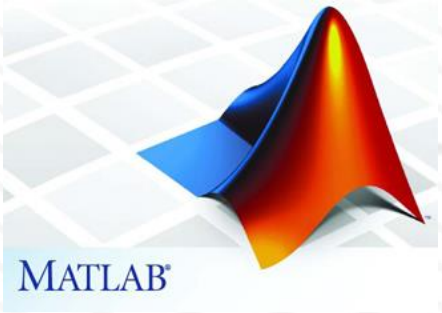


Functional Simulation / Verification



Functional Simulation / Verification

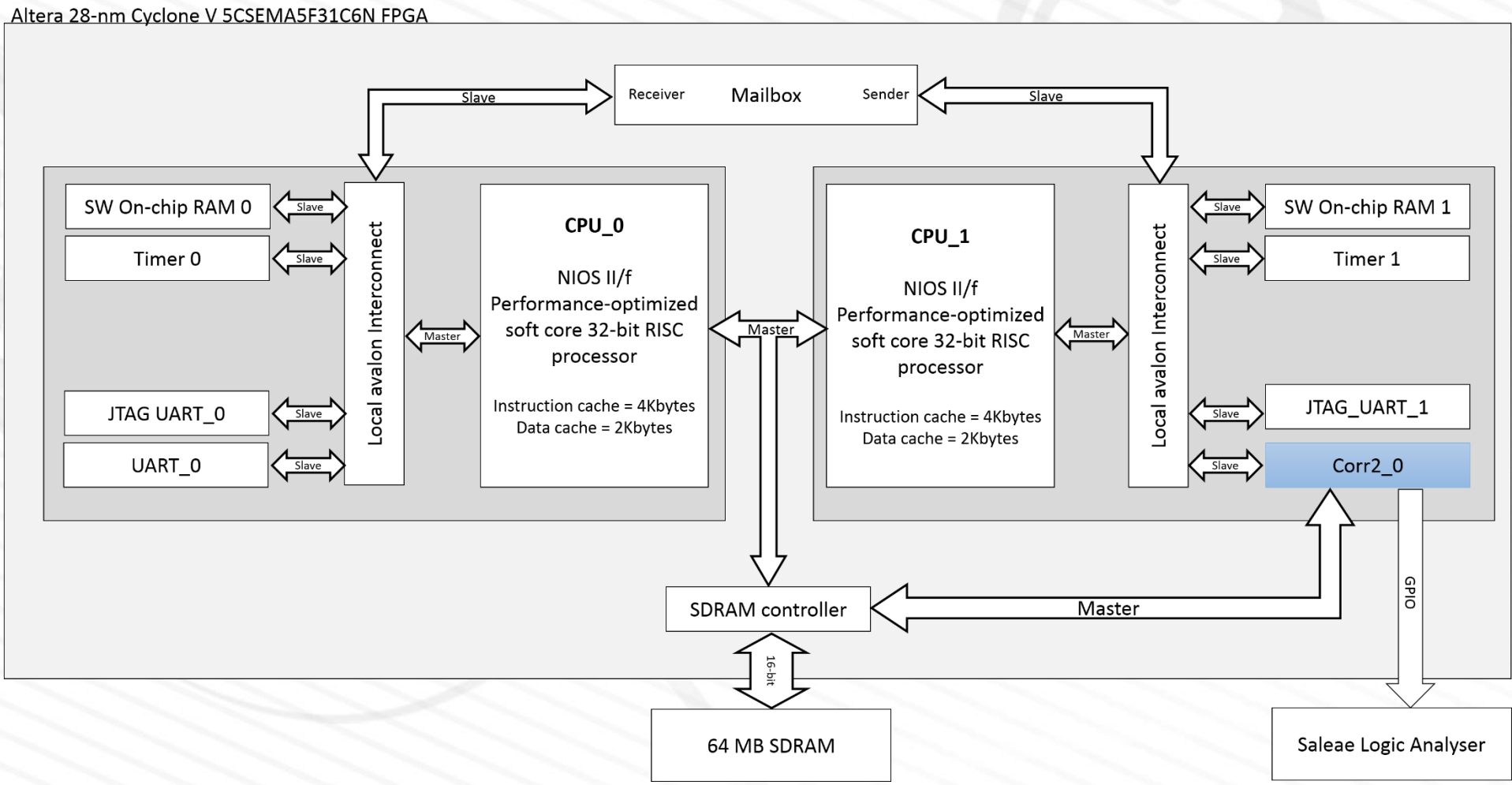
ModelSim



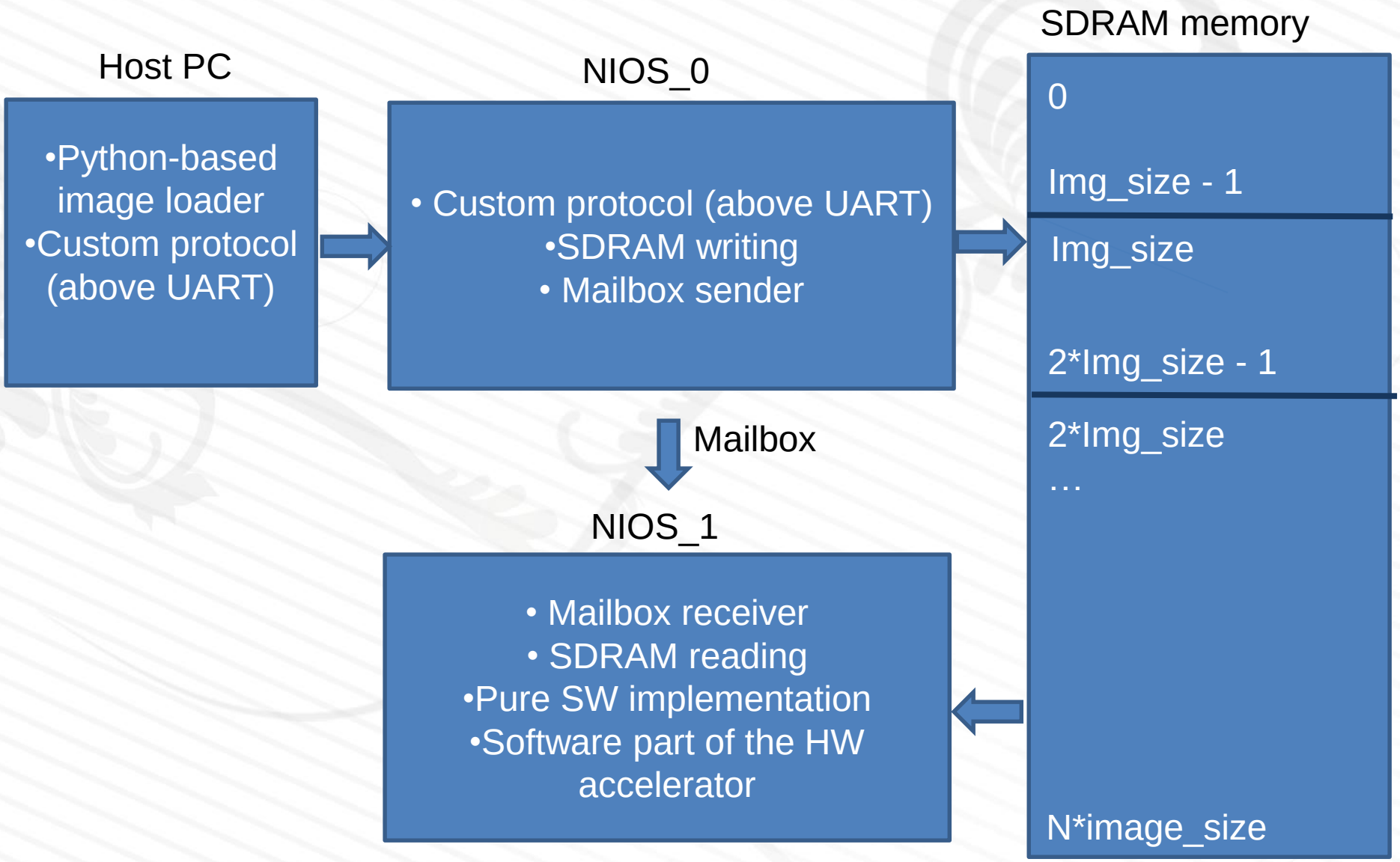

```
ModelSim> run 15 us
# ** Note: Firmware Version: 0xABCD0001
#    Time: 100 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: DMA Start Addr: 0x00001000
#    Time: 220 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: DMA End Addr: 0x00001010
#    Time: 260 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: DMA Burst Count: 0x00000010
#    Time: 300 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: DMA Fifo Threshold: 0x00003FFF
#    Time: 340 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Fifo / Calb register: 0x00000000
#    Time: 400 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: FIFOA Used words: 0x00000000
#    Time: 440 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: FIFOB Used words: 0x00000000
#    Time: 480 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Accumulator result: 0x00000840
#    Time: 2960 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: FIFOA Used words: 0x00000020
#    Time: 3 us  Iteration: 1  Instance: /corr2_tb
# ** Note: FIFOB Used words: 0x00000000
#    Time: 3040 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Accumulator result: 0x00000840
#    Time: 5620 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: FIFOA Used words: 0x00000020
#    Time: 5660 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: FIFOB Used words: 0x00000020
#    Time: 5700 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected XZ[0]=0xAB0 | Actual: 0x00000AB0
#    Time: 5760 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected XZ[1]=0xAB0 | Actual: 0x00000AB0
#    Time: 5800 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected XZ[2]=0xAB0 | Actual: 0x00000AB0
#    Time: 5840 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected XZ[3]=0xAB0 | Actual: 0x00000AB0
#    Time: 5880 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Accumulator result: 0x00000900
#    Time: 8420 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: FIFOA Used words: 0x00000020
#    Time: 8460 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: FIFOB Used words: 0x00000020
#    Time: 8500 ns  Iteration: 1  Instance: /corr2_tb
```

```
# ** Note: Expected XZ[0]=0xAB0 | Actual: 0x00000AB0
#    Time: 8560 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected XZ[1]=0xAB0 | Actual: 0x00000AB0
#    Time: 8600 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected XZ[2]=0xAB0 | Actual: 0x00000AB0
#    Time: 8640 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected XZ[3]=0xAB0 | Actual: 0x00000AB0
#    Time: 8680 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected Y[0]=0xAB0 | Actual: 0x00000AB0
#    Time: 8720 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected Y[1]=0xAB0 | Actual: 0x00000AB0
#    Time: 8760 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected Y[2]=0xAB0 | Actual: 0x00000AB0
#    Time: 8800 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected Y[3]=0xAB0 | Actual: 0x00000AB0
#    Time: 8840 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Accumulator result: 0x00000B00
#    Time: 11380 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: FIFOA Used words: 0x00000020
#    Time: 11420 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: FIFOB Used words: 0x00000020
#    Time: 11460 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected XZ[0]=0xAF0 | Actual: 0x00000AF0
#    Time: 11520 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected XZ[1]=0xAB0 | Actual: 0x00000AB0
#    Time: 11560 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected XZ[2]=0xAB0 | Actual: 0x00000AB0
#    Time: 11600 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected XZ[3]=0xAF0 | Actual: 0x00000AF0
#    Time: 11640 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected Y[0]=0xA90 | Actual: 0x00010A90
#    Time: 11680 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected Y[1]=0xAA0 | Actual: 0x00000AA0
#    Time: 11720 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected Y[2]=0xAB0 | Actual: 0x00000AB0
#    Time: 11760 ns  Iteration: 1  Instance: /corr2_tb
# ** Note: Expected Y[3]=0xAC0 | Actual: 0x00000AC0
#    Time: 11800 ns  Iteration: 1  Instance: /corr2_tb
```

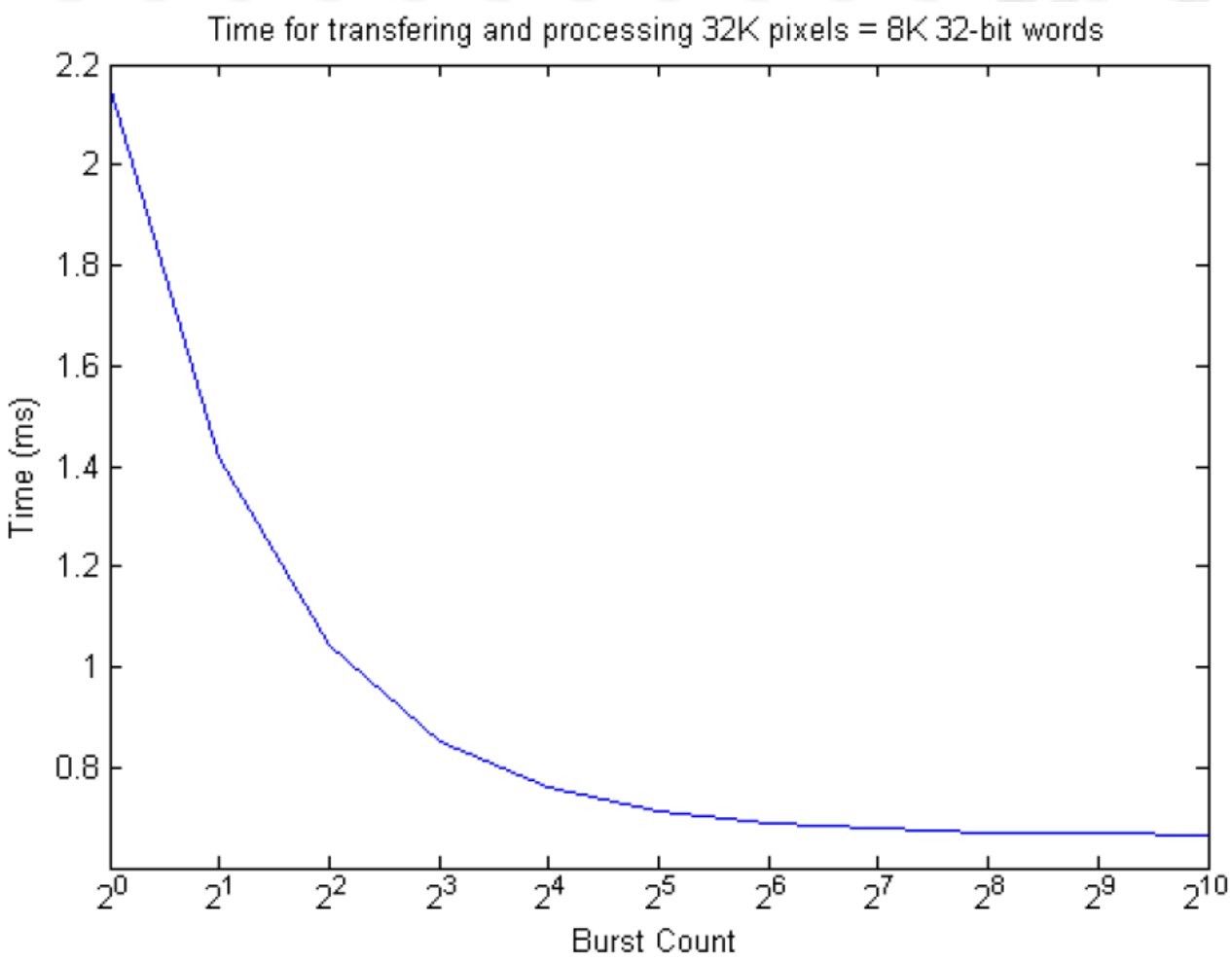
Top level block diagram



Software structure



Performance measurements



Time for loading 32K pixel image into the FIFO vs burst count

Not worth using DMA bursts transfers longer than 64 words

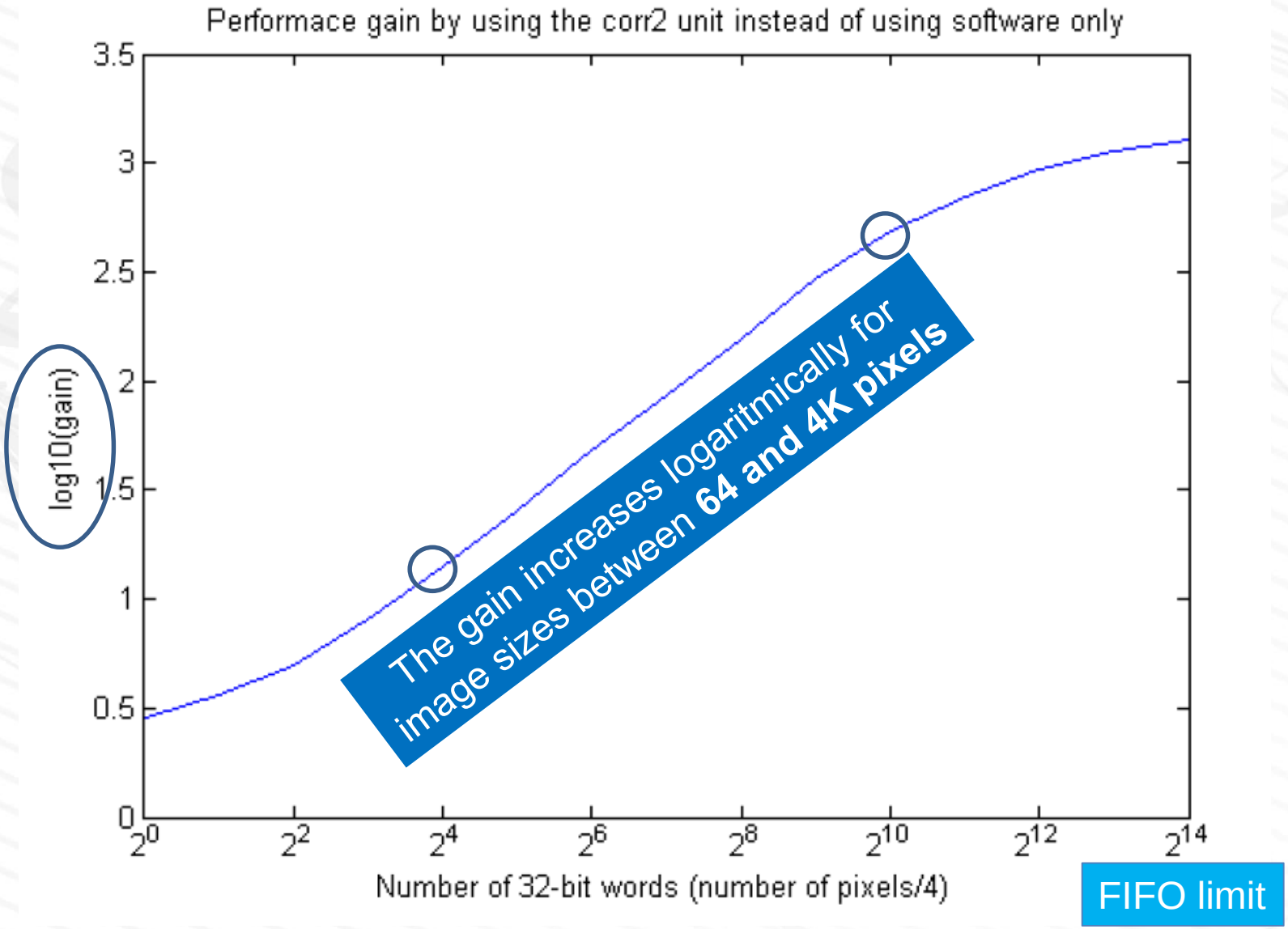
(for the SoC being investigated in this study)

Performance measurements

32-bit words	Burst count	Time for software (s)	Time for hw (s)	Gain
1	1	0.001363	0.000482	2.827801
2	2	0.001793	0.000494	3.629555
4	4	0.002437	0.000487	5.004107
8	8	0.004009	0.000494	8.115385
16	16	0.006943	0.000496	13.997984
32	32	0.012623	0.000494	25.552632
64	64	0.024511	0.000513	47.779727
128	64	0.046164	0.000542	85.173432
256	64	0.091527	0.000578	158.351211
512	64	0.180916	0.000620	291.800000
1024	64	0.360766	0.000739	488.181326
2048	64	0.708648	0.001019	695.434740
4096	64	1.42377	0.001529	931.177240
8192	64	2.864359	0.002523	1135.298851
16320	64	5.818144	0.004555	1277.309330

The corr2 ip always outperforms the processor-only solution

Performance measurements



Conclusions

- There is a gain of 5X in loading 32K pixel images using DMA burst transfer of 64 words instead of single cycle transfers.
- The corr2 ip always outperforms the processor-only solution for any image size, including the minimum size of 4 pixels.
- The gain increases logaritmically for image sizes between **64 and 4K pixels**, ranging from **14X to 488X** respectively.
- The gain reaches **1277X** for images with size of **64K pixels**, which corresponds to the FIFO limit.

- **The study presented here shows that the use of hardware aceleration is very interesting for image processing algorithms such as 2D correlation.**