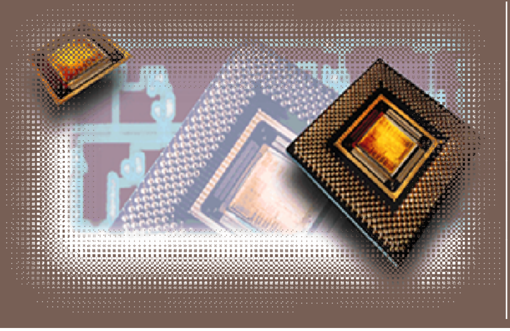
JOINT INSTITUTE
FOR NUCLEAR RESEARCH

National Institute for
Research and Development
of Isotopic and Molecular
Technologies

GRID Based High Performance Computing in Satellite Imagery. Case Study – Perona-Malik filter

Bogdan BELEAN | Adrian BOT | Carmen BELEAN |
Calin FLOARE | Codruta VARODI | Gheorghe ADAM



The 6th International Conference "Distributed Computing and Grid-Technologies
in Science and Education“, LIT-JINR Dubna, 30 June – 05 July 2014

Outline

A UNOSAT – An instance of Grid based approach to satellite imagery

B Elements of satellite imagery

C Essentials of Perona-Malik filter for digital image processing

D
- Integration of application specific hardware architectures for grid based satellite imagery
- FPGA based hardware architecture for Perona-Malik

E A Look into the Future

UNOSAT – An instance of Grid based approach to satellite imagery

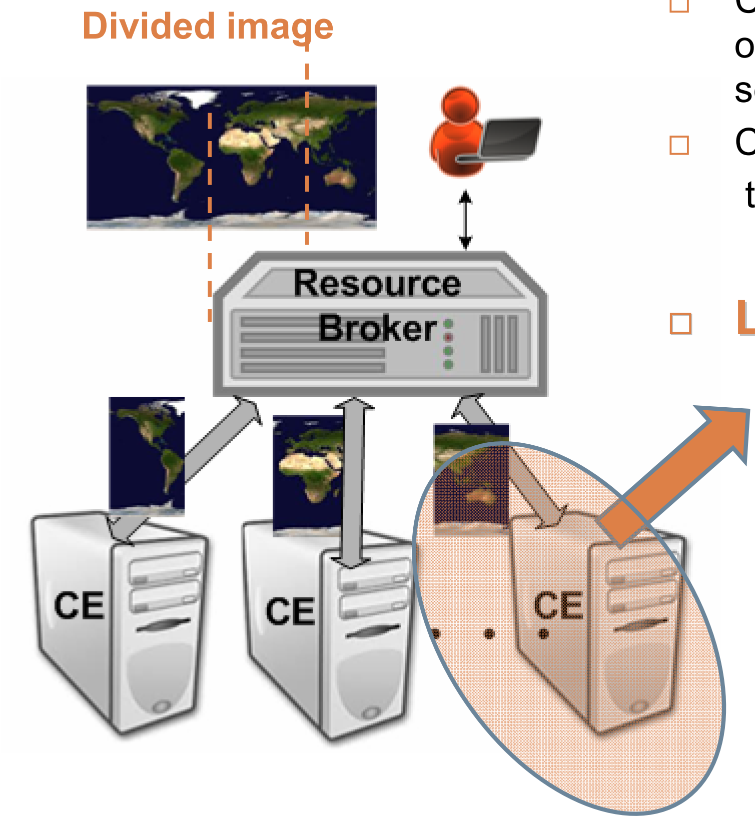
UNOSAT is the United Nations Institute for Training and Research (**UNI**TAR) **O**perational **Sat**ellite Applications Programme. Created in 2000, it provides the worldwide users with high quality satellite imagery and Geographic Information System (GIS) services. These serve for planning sustainable development or monitoring natural disasters.

Excessive loads on the UNOSAT website, putting strong pressure on the computing and storage resources are frequently encountered. Instances:

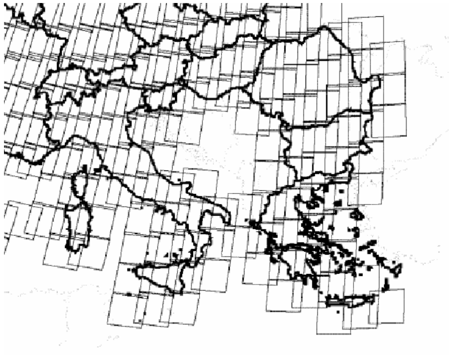
- *Use case 1.* During ***natural catastrophes*** and ***disasters***.
- *Use case 2.* ***Web visualization from mobile devices*** of the field workers (even though using compression and cropping when interrogating UNOSAT resources).
- *Use case 3.* Need of storage and computing resources ***from UNOSAT*** by ***users having slow internet connection.***
- *Use case 4.* The ***periodically performed updates*** by the UNOSAT administrators (asked by image uploading in the databases; own search and processing tasks on the satellite image databases).
- Since 2001, UNOSAT has been based at CERN and is supported by CERN's IT Department in the work it does.
- **CERN provides Grid approach** to the operations done on the satellite images:
  - Storage
  - Processing
  - Compression

A

Specificity of GRID based satellite image processing



- Cf. [A1- A3]: The satellite images ***are divided in sub-images*** in order to reduce size to be processed. Each sub-image can be sent for processing to a different CE within the grid.
- Cf.[A1] ***Thiessen polygons*** are used to divide the satellite images



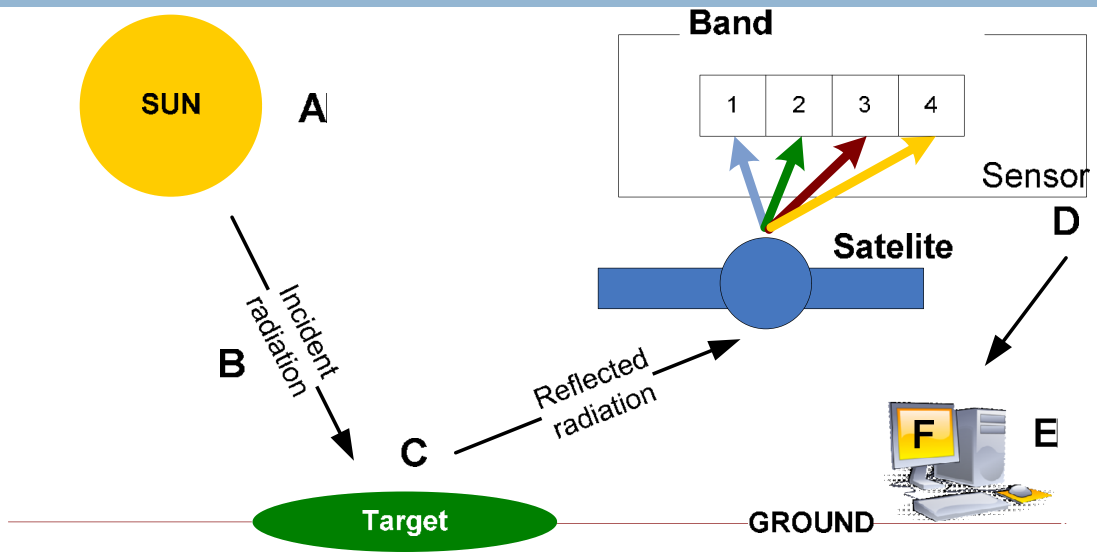
- **LIMITATIONS in case of iterative algorithms:**
  - The computation power is limited to one Computing Element (CE) while processing one sub-image
  - The computing elements are General Purpose Processors (GPPs) (e.g. **Intel® Xeon® Processor E5**)
  - The parallel processing strategies to be applied are limited by the available GPPs

[**B1**] F. Javier Gallego, *Stratified sampling of satellite images with a systematic grid of points*, **ISPRS Journal of Photogrammetry & Remote Sensing** 59 (2005) 369–376
[**B2**] Gregory Giuliani, Nicolas Ray, Anthony Lehmann, *Grid-enabled Spatial Data Infrastructure for environmental sciences: Challenges and opportunities*, **Future Generation Computer Systems**, 27 (2011) 292–303
[**B3**] Sauravjyoti Sarmah, Dhruba K. Bhattacharyya, *A grid-density based technique for finding clusters in satellite image***,** **Pattern Recognition Letters** 33 (2012) 589–604.

B

Elements of satellite imagery:

Physical processes involved in satellite image collection



- □ Propagation through the atmosphere of the incident radiation (B), emitted by an energy source (A)
- □ Interaction of the radiation with the surface of the target (C), depending on the surface characteristics and the wavelength of the incident radiation
- □ Radiation is reflected or scattered to a sensor (D) [placed on a satellite], which registers it and transmits it by remote means to a receiving station (E), where the information is transformed into digital images
- □ A visual interpretation of the digital image (F) is required to extract the information on the target surface (C)

Elements of satellite imagery:

Resolution characterization of a satellite image

- ***Spatial resolution*** – is described by the pixel size of the image

- ***Spectral resolution*** – is defined by the wavelength interval within the electromagnetic spectrum and the number of intervals measured by the sensor. Examples:

  visible images – come from sunlight reflected by the earth surface

  infrared images – come from infrared sensor measurements of the temperature of the earth surface

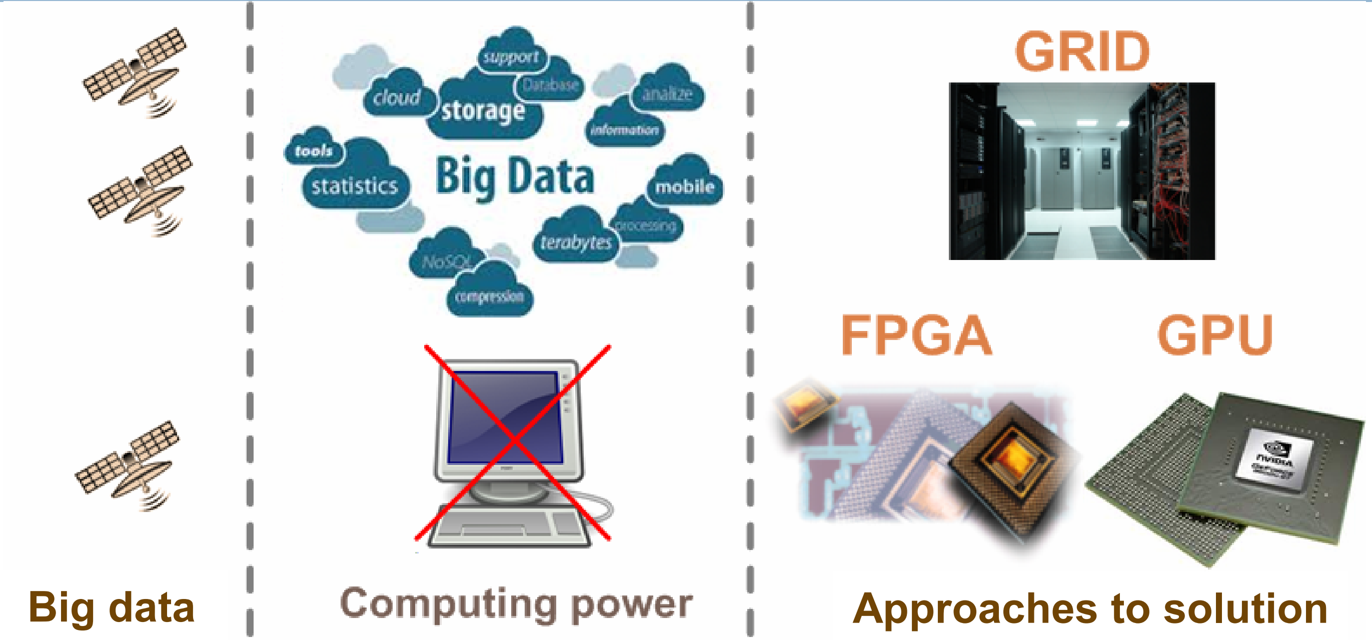
  water vapor images – infrared measurement of the temperature in a layer of the atmosphere about 6 – 10 km above the earth surface

- ***Temporal resolution*** – is defined by the time interval inbetween two consequent image acquisitions for a given surface location

- ***Radiometric resolution*** – is defined by the amount of levels of brightness recorded by the imaging system

B

Elements of satellite imagery:
Data accumulation and processing



- Big data outcomes – are consequences of increasing the number of high resolution satellites into orbit and the number of applications which use satellite images
- Local computing infrastructures are insufficient for data processing
- Possible approaches to solution:
  - use of GRID environment – implemented by CERN for UNOSAT
  - use of application specific hardware architectures (FPGA, GPU)

Elements of satellite imagery:

Fields of application

Satellite images are of fundamental interests in meteorology, agriculture, geology, forestry, landscape, biodiversity conservation, regional planning, education, intelligence and warfare.

Commercial applications:

- Insurance companies – damage estimates based on acquired images before and after disaster
- Mass Media – satellite imagery based news reports
- Software developers – image incorporation in flight simulators, games
- Combined with GPS for localization in geographic information systems, e.g., Google Earth and Google Earth Pro

To secure smooth continuous functioning of the system, a vital condition is to produce ***maximum relevant information*** under ***minimum system load***.

To this aim, the appropriate digital image processing is a must. The ***Perona-Malik filter*** offers such a functional tool.

Essentials of Perona-Malik filter for digital image processing (1)

- ***Two basic aims***
  - Eliminate disturbances, i.e., filter out noise
  - Retain and enhance essential information
- ***Basic idea:***
  - The above aims can be achieved by solving a Neumann boundary problem for a generalized diffusion equation

$$\partial_t u(x,t) = \nabla(g(u,x,t)\nabla u(x,t)) \qquad \text{(DEq)}$$

where the diffusivity function ***g*** is to be ***anisotropic***, such that, along some directions

$$g \gg 1 \qquad \text{[strong diffusion]}$$

while along other directions

$$g \ll 1 \qquad \text{[weak smoothing]}$$

- ***Why (DEq)?***
  - Under $g \equiv 1,$ the (DEq) describes the ***heat propagation.***
  - The solution of the ***heat equation*** is a ***convolution*** of the initial value $u(x,0) = f(x)$ with a ***Gauss distribution function***.
  - Since the latter is decreasing fast (both in the coordinate and the frequency spaces), the ***fast oscillations are cut out***, hence (DEq) acts as an effective ***low pass filter***.

Essentials of Perona-Malik filter for digital image processing (2)

- ***Using* (DEq) *approach to digital image processing***
  - There is a ***bounded*** domain of the digital image,

$$\Omega \subset \mathbb{R}^n \quad (n = 2, 3), \quad \text{of boundary } \partial\Omega \text{ of class } \mathbb{C}^1$$

  - The mapping

$$u : \Omega \to [0,1]$$

then achieves the correspondence from Ω to the ***gray level distribution*** (GLD) of a noisy image.

  - The numerical investigation of the time evolution of the GLD, performed through an ***iterative approach***, results in ***successive instances*** attempting at solving the filtering tasks.
- ***Two contradictory features of the Gaussian smoothing:***
  - ***Efficient*** noise filtering
  - ***Edge smearing*** (image blurring) results in ***quick loss*** of essential information contained in the original image.

Essentials of Perona-Malik filter: The Perona-Malik equation (1)

- ***Perona-Malik hypothesis:*** the diffusivity coefficient in (DEq) is to be a function of the ***norm of the local gray level distribution gradient.*** This results into the diffusion Perona-Malik problem:

$$\begin{aligned} &\partial_t u(x,t) = \nabla(g(|\nabla u|^2)\nabla u(x,t)) && \text{in} \quad \Omega\times(0,+\infty) \\ &\partial u/\partial n = 0 && \text{in} \quad \partial\Omega\times(0,+\infty) \\ &u(x,0) = f(x) && \text{in} \quad \Omega. \qquad \text{(DPMEq)} \end{aligned}$$

- ***(DPMEq) basic features***. It is an ***ill-posed*** problem, which ***does not*** admit a weak solution. A solution could, nonetheless, be defined in the sense of distributions. However, under ***spatial discretization by finite differences***, a ***well-conditioned problem*** arises, with an unexpectedly good filtering efficiency. This is usually called the ***Perona-Malik paradox*** in the digital image processing.

- An important quantity is ***the flux function***,

$$\Phi(s) = s\cdot g(s^2) > 0 \qquad \text{for} \qquad s\in(0,+\infty)$$

which is asked to ***vary smoothly with s*** and to have a maximum on $(0,+\infty)$, at some characteristic value $s_0 = \lambda > 0.$

Essentials of Perona-Malik filter: The Perona-Malik equation (2)

□ ***The diffusivity function*** $g(s^2)$ enabling such properties of $\Phi(s)$ should be infinitely continuous differentiable and to decrease monotonically from 1 to 0 while s^2 varies from 0 to $+\infty$.

□ ***An expression*** of $g(s^2)$ inspired by the Gaussian distribution function is

$$g(s^2) = \exp(-s^2/2\lambda^2), \quad \lambda > 0.$$

Then $\Phi(s)$ has a maximum at $|s| = \lambda,$ with $\Phi'(s) > 0$ for $|s| < \lambda$ and $\Phi'(s) < 0$ for $|s| > \lambda.$

□ In the ***two-dimensional case***, let ξ and η denote the local coordinates in directions perpendicular and parallel to ∇u respectively.

Then the Perona-Malik equation can be rewritten as

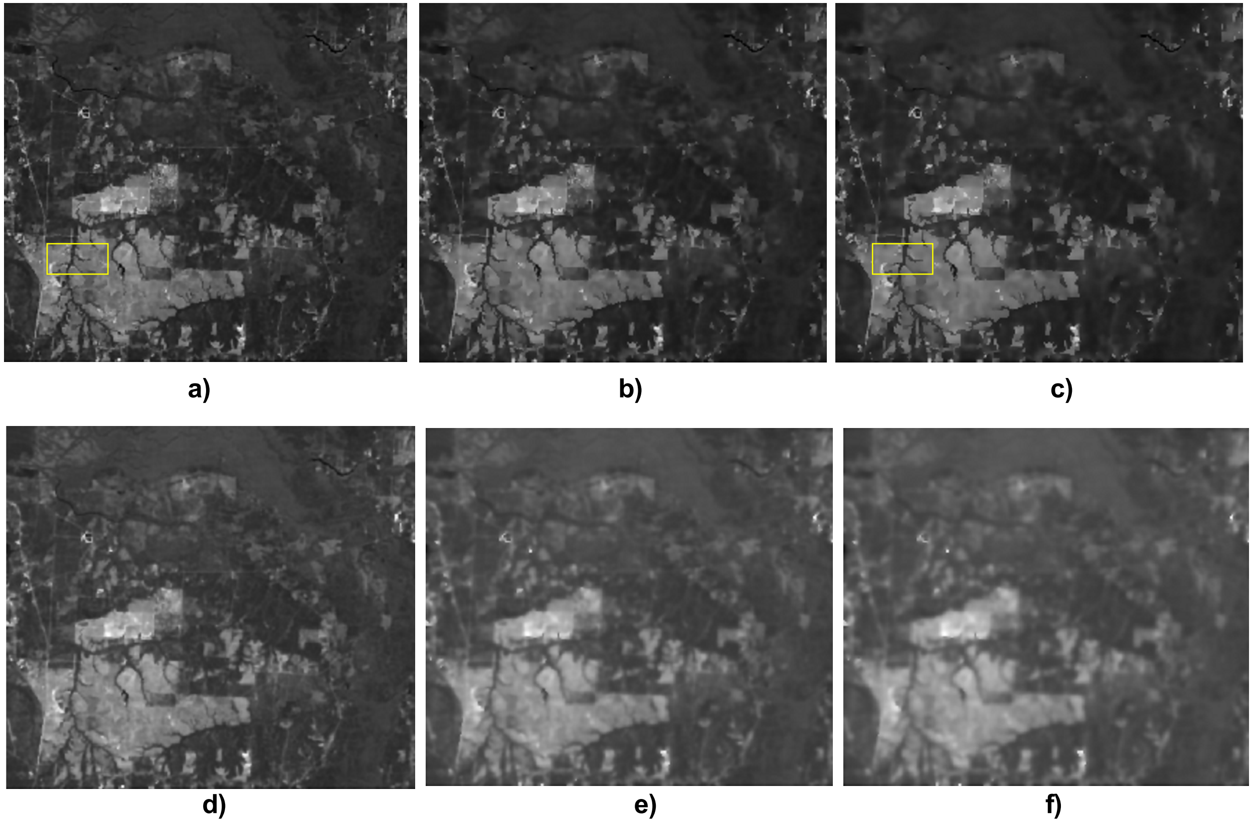
$$\partial_t u = g(|\nabla u|^2) u_{\xi\xi} + \Phi'(|\nabla u|) u_{\eta\eta}.$$

The coefficient of $u_{\xi\xi}$ is always positive, hence (DPMEq) acts as a smearing filter washing details along the contour lines of the function u.

The coefficient of $u_{\eta\eta}$ may be both positive and negative, hence in the perpendicular (gradient) direction, slow gradient values are smeared out, while ***large gradient values*** (like ***edges***) are sharpened instead of being blurred.

Essentials of Perona-Malik filter: An instance of usage for digital image processing

- Parameters are: the number of iterations *Num_Iter*, integration constant *Delta_T* which is set usually to maximum value and the gradient modulus threshold that controls the conduction denoted by *λ*.



Anisotropic diffusion applied for edge enhancement in case of original image *Florida* :
a) *Num_iter = 5, λ = 10*, b) *Num_iter = 15, λ = 10*, c) *Num_iter = 25, λ = 10,*
d) *Num_iter* = 5, *λ* = 30*, e) Num_iter* = 15, *λ* = 30*,* f) *Num_iter = 25, λ = 30.*

D

Integration of ASHA - Application Specific Hardware Architectures - for grid based satellite imagery

- GPU and FPGA represent a solution for parallel processing of satellite images
  - They can be used in conjunction with the **grid based approach** for fast processing by implementing both **spatial** parallelism and **temporal** parallelism

Perona-Malik filter: computational steps

□ Given an initial image which is to be processed within **N iterations** (empirically N = 10 to 20)

N iterations

for each *p(i,j)*

- a) **STEP 1** - compute finite differences

  η_N, $\eta_{S,}$ $\eta_{E,,}$ … η_{NW} using

  the following computational masks

  hN = [0 1 0; 0 -1 0; 0 0 0];

  hS = [0 0 0; 0 -1 0; 0 1 0];

  ……………………..

  hNW = [1 0 0; 0 -1 0; 0 0];

  - equivalent with 8 additions

- b) **STEP 2** - compute diffusion function

$$\underbrace{c_N = e^{-(\eta_N/k)^2} \qquad c_S = e^{-(\eta_S/k)^2} \qquad \ldots.. \qquad c_{NW} = e^{-(\eta_{NW}/k)^2}}_{\textit{- 8 computations of the exponential function}}$$

- c) **STEP 3** - the p(i,j) pixel within the resulted image after each iteration is computed as follows

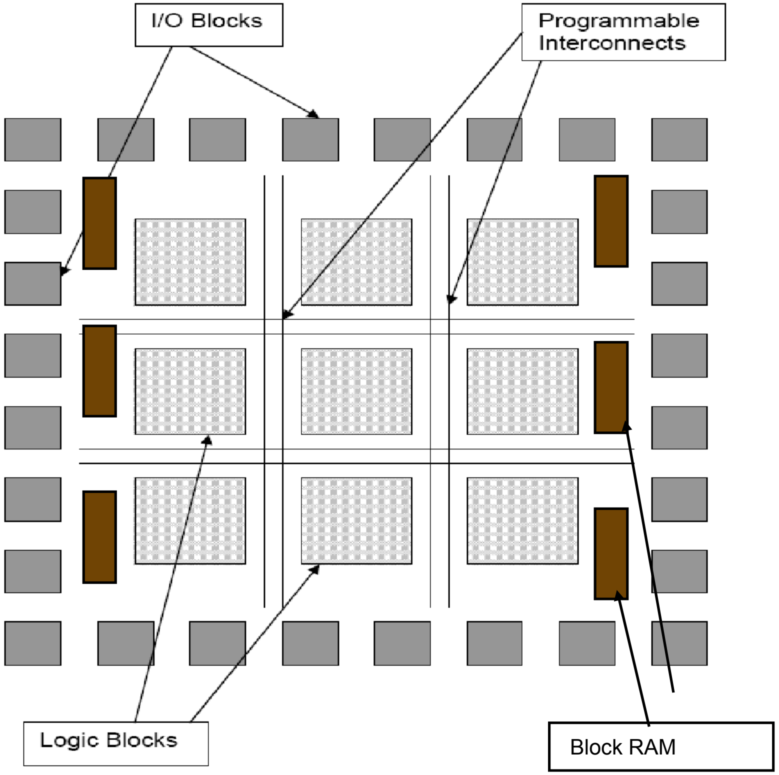
$$p(i,j) = p(i,j) + c_N\eta_N + c_S\eta_S + c_E\eta_E + c_W\eta_W +$$
$$+0.5c_{NE}\eta_{NE} + 0.5c_{SE}\eta_{SE} + 0.5c_{NW}\eta_{NW} + 0.5c_{SW}\eta_{SW}$$

- 8 multiplications
- 8 additions

□ **STEP 1, STEP 2, and STEP 3 are to be parallelized for efficient computation**

Hardware Architecture for Perona-Malik filter:
Field Programmable Gate Arrays

- Are digital logic chips containing:
  - Configurable Logic Blocks (CLB)
  - Programmable interconnects
  - I/O Blocks (programmable)
  - Block RAMs
  - Processors (Power PC)

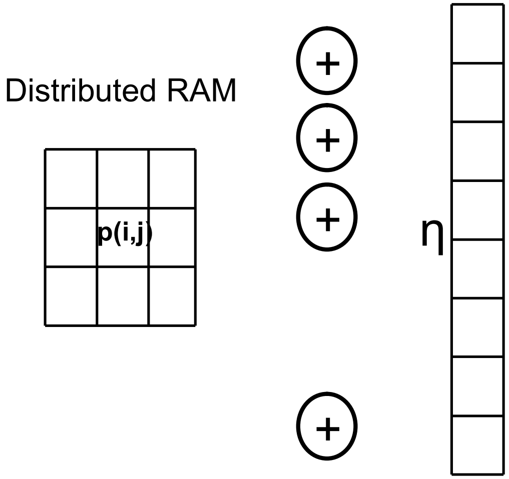


- Parallel computing capabilities; possibilities to exploit:
  - Spatial parallelism (e.g multiple computing units)
  - Temporal parallelism (pipeline approaches)

D

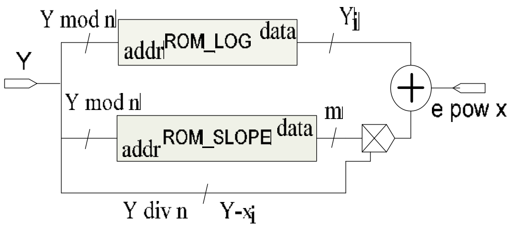
Hardware Architecture for Perona-Malik filter: Spatial parallelism

STEP 1

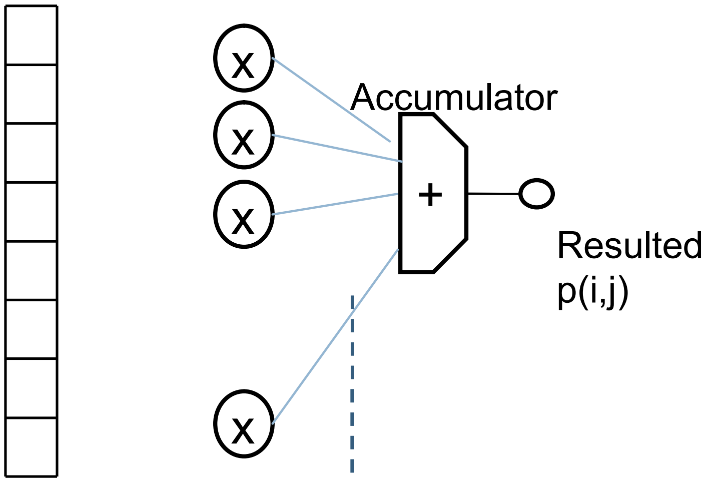


STEP 2

Compute e^x using linear approximations (8 architectures as the following one are used)



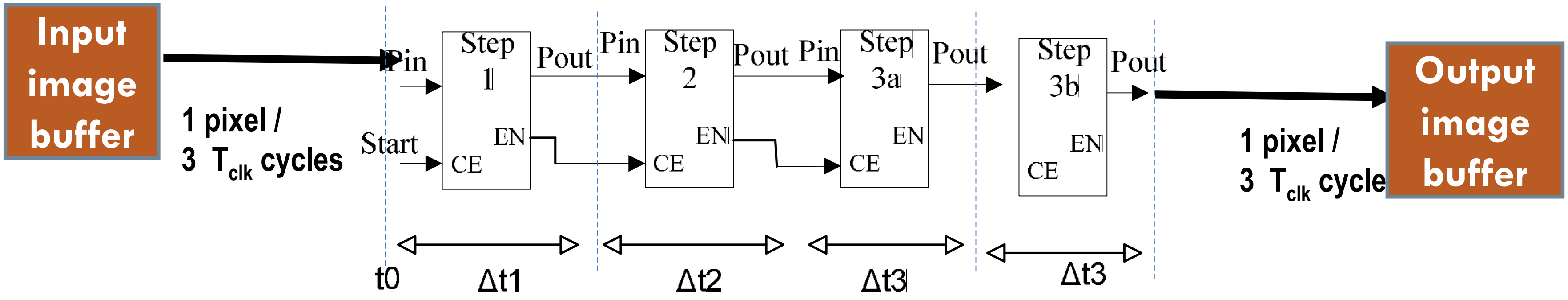
STEP 3

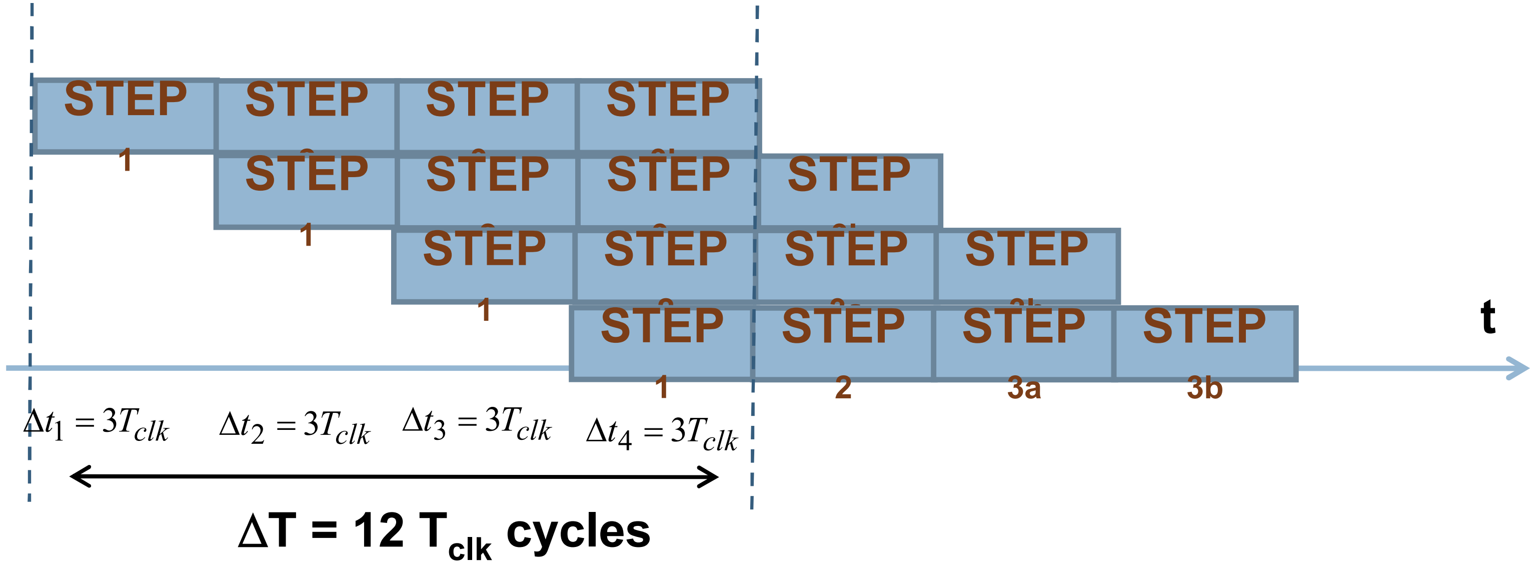


Computational time	$\Delta t_1 = 3T_{clk}$	$\Delta t_2 = 3T_{clk}$	$\Delta t_3 = 3T_{clk}$ STEP 3a	$\Delta t_4 = 3T_{clk}$ STEP 3b
Hardware resources	-1 Distributed RAM memory -8 adders - 1 register	-2 ROM memories - 8 adders - 8 multipliers	-1 Register -8 multipliers -1 accumulator	

Hardware Architecture for Perona-Malik filter:
Temporal parallelism

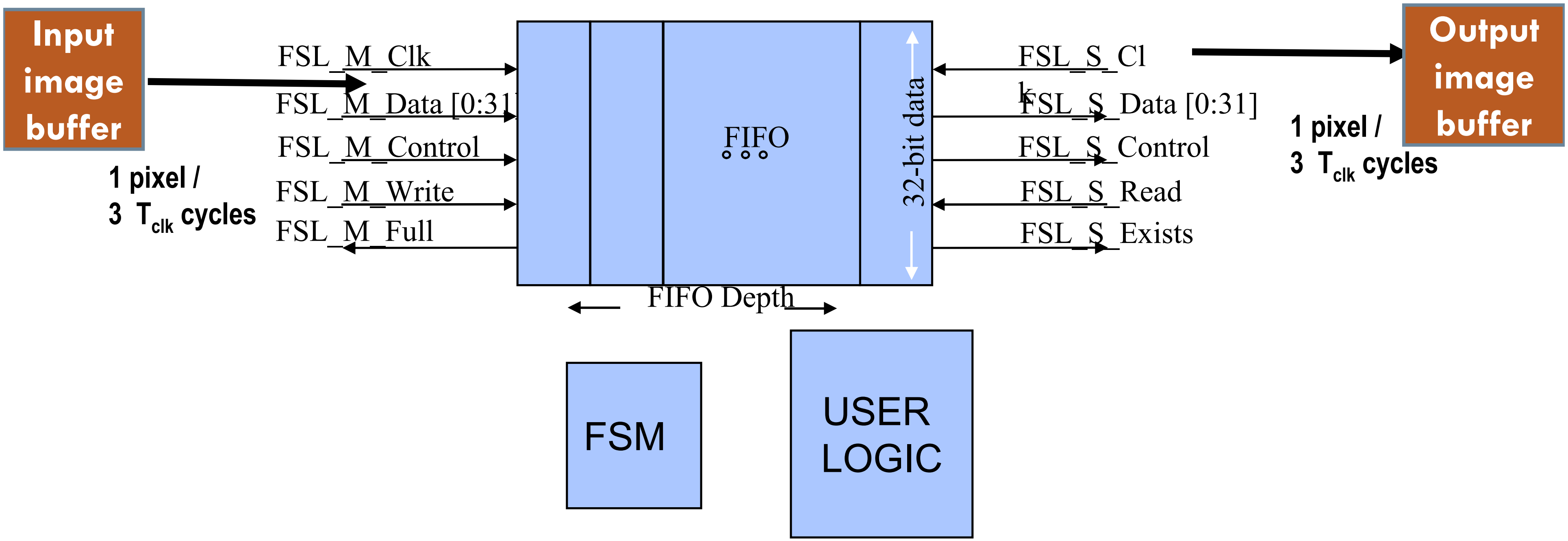
- Previously described computational steps are arranged in a **pipeline** architecture:
  - Each computational step has assigned a FPGA based architecture





Hardware Architecture for Perona-Malik filter:
FSL data bus

- Unidirectional point-to-point FIFO-based communication
- Dedicated (unshared) and nonarbitrated architecture
- Dedicated MicroBlaze™ C and ASM instructions for easy access
- High speed, access in as little as two clocks on processor side, 600 MHz at hardware interface
- Available in Xilinx Platform Studio (XPS) as a bus interface library core from **Hardware → Create or Import Peripheral Wizard**



Input image buffer

1 pixel / 3 T_{clk} cycles

FSL_M_Clk

FSL_M_Data [0:31]

FSL_M_Control

FSL_M_Write

FSL_M_Full

FIFO

32-bit data

FIFO Depth

FSL_S_Cl k

FSL_S_Data [0:31]

FSL_S_Control

FSL_S_Read

FSL_S_Exists

Output image buffer

1 pixel / 3 T_{clk} cycles

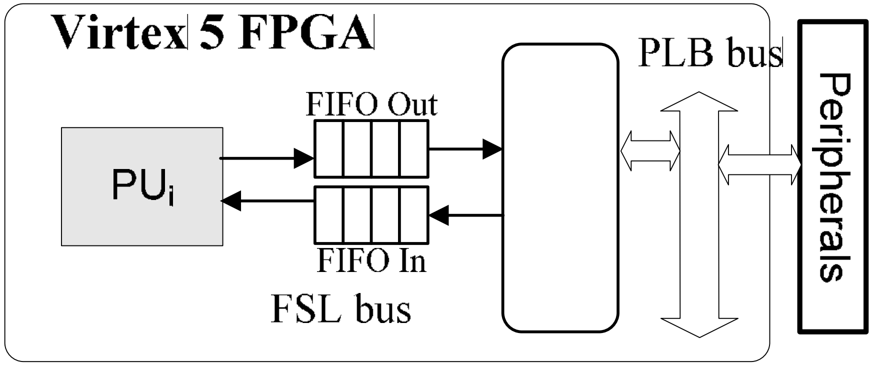
FSM

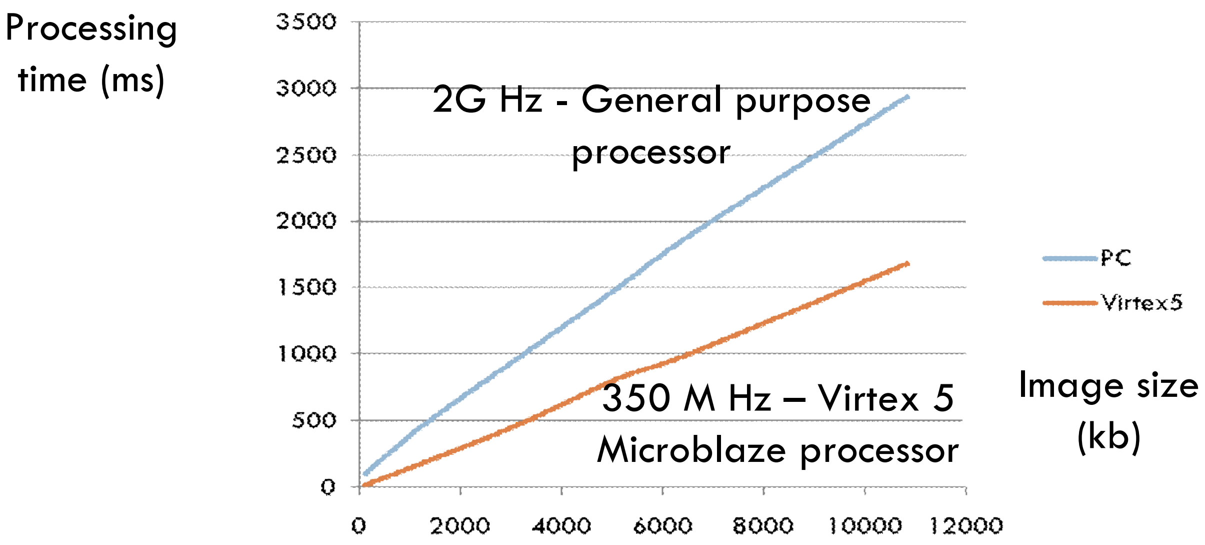
USER LOGIC

Hardware Architecture for Perona-Malik filter

Implementation of computational process and result

- *The microarray image is delivered pixel by pixel to the computing unit PU with the help of a MICROBLAZE processor, trough the FSL data bus*





E

A Look into the Future

- In future attempts to implement solutions for specific iterative algorithms, we intend to intensively use Field Programmable Gate Arrays
- The reported results suggest that the general purpose processors are surpassed by Application Specific Hardware Architectures as it concerns the computing time
- In the next future, we plan to derive an GPU implementation of the Perona-Malik filter and to compare it with the FPGA implementation
- In the processing of grid Big Data sets, extensive use of Application Specific Hardware Architectures might offer efficient solutions.

Thank you for your attention!