eman ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

GRUPO DE DISENO EN ELECTRÓNICA DIGITAL

ADAPTIVE SCALABLE SVD UNIT FOR FAST PROCESSING OF LARGE LSE PROBLEMS

IÑAKI BILDOSOLA & UNAI MARTINEZ-CORRAL
KOLDO BASTERRETXEA

MOTIVATION

UPV EHU

-**Motivation**
-Selecting the Algorithm
-Selecting SVD Method
-Speaking About the Accuracy
-Improving Previous Work
-Results
- HW FPGA Implementation
-Why and How Scalable?
-Conclusion
-Future Work

- Previous Project
  - Computational Intelligence applications
  - Real Time Computation
  - LSE problems
  - Resulting Matrices:
    - Large-scale
    - Rank-deficient
    - Ill-conditioned matrices
  - Implementation in MicroBlaze → Too Much Delay
  - Need for acceleration → parallel processing & optimized faster implementation → FPGA

SELECTING THE ALGORITHM
WHY SVD ?

- We needed an algorithm numerically robust
- Struggling with deficient matrices
- Struggling with non-square matrices
- Avoid the Inverse calculation
- Obtain the Pseudoinverse
- Good Base for Problem Reduction (future Work)

SELECTING SVD METHOD
WHY ONE-SIDED JACOBI?

- Easily Parallelizable → ***Jacobi***
- What more?
  - Purely non-conflicting →***one-sided***
  - Optimizing the managed unit size → ***one-sided***
- Main features
  - Based on Column Pairs Orthogonalization
  - Given's rotations by Rutishauser formulas

$$\tan(2\theta_{ij}^{k}) = \frac{2 * (A_{:,i}^{k} * A_{:,j}^{k})}{||A_{:,j}^{k}||^2 - ||A_{:,i}^{k}||^2} \quad AV = W \quad A = USV^T$$

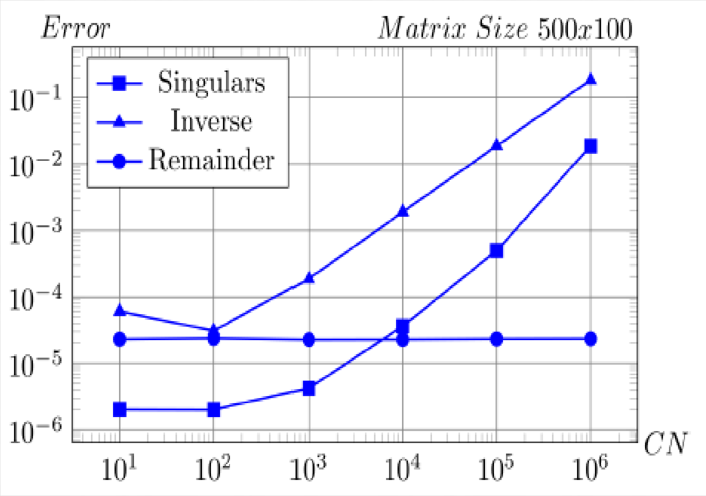
SPEAKING ABOUT THE ACCURACY IMPOSSED CONDITIONS

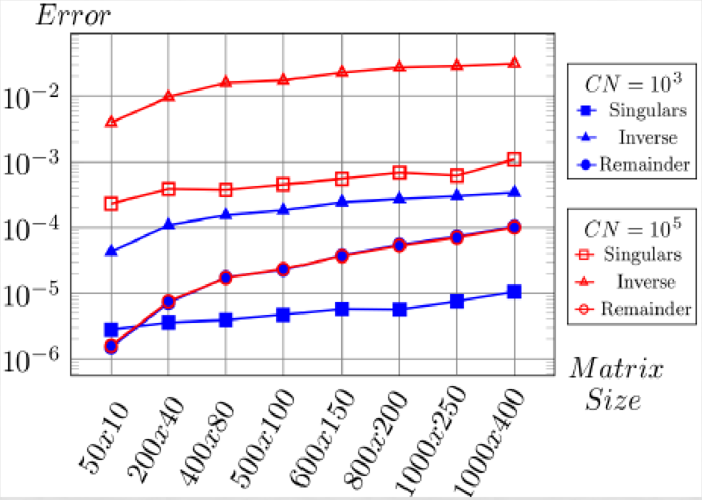
- Computing Precisions → Sets The Maximum
- Matrix Conditioning → Impacts on the Accuracy
  - K(A) = 10^k ; CP = 10^m ; Solution = 10^{m-k}.
  - Matrix Size → Posible Accumulated error.
  - K(A) & Matrix Size Impact close to CP → No Solution or very degraded

SPEAKING ABOUT THE ACCURACY: IMPOSSED CONDITIONS

-Motivation
-Selecting the Algorithm
-Selecting SVD Method
-Speaking About the Accuracy
-Improving Previous Work
-Results
- HW FPGA Implementation
-Why and How Scalable?
-Conclusion
-Future Work

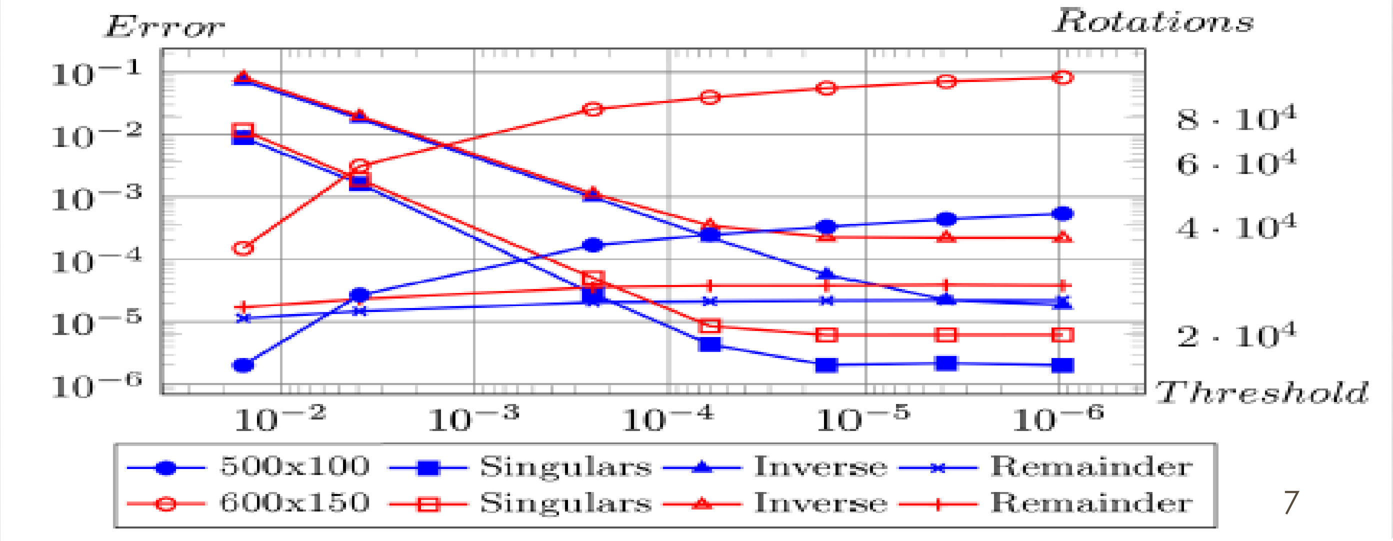
- Randomly generated matrices f(Size, k(A))
- Errors: Our Algorithm (′) in Single Vs Matlab(″) in double
  - Singulars = Maximum Normalized Error = $\frac{\sigma' - \sigma''}{\sigma''}$
  - Inverse = $\|A'_{inv} - A''_{inv}\|^2$
  - Remainder = $\|A - SVD\|^2$





SPEAKING ABOUT THE ACCURACY DECIDED CONDITIONS

- **Threshold Value**
  - Iterative Algorithm Finisher → Orthogonalization
  - User Defined Parameter → Time & Accuracy Trade-off
  - Error Saturation Phenomenon → (Imposed Conditions)



IMPROVING PREVIOUS WORK
LEARNING FROM OTHERS

- **Brent and Luck**
  - Highlighted the column norm importance
  - Normalized the Threshold → Adapting to the columns' norm → Actually calculating the cosine: $\frac{A_i \cdot A_j^T}{\|A_i\|\|A_j\|} < Threshold$
- **Hestenes**
  - Swap the columns → Active Sorting → f(column norm)

IMPROVING PREVIOUS WORK ADDING OUR TOUCH

- Increased Adaptability
  - Realizing that the "Inverse Error" lies on small columns
  - Being Fussier with them → Harder Threshold
  - With Easier Threshold → Same Solution Accuracy
  - Not rotating in vain the big columns
  - AMN:
  - $\frac{A_i \cdot A_j^T}{\|A_i\|\|A_j\|} = \cos(A_i, A_j) < Threshold \cdot \min(\|A_i\|, \|A_j\|)$
  - AAMN:
  - $\frac{A_i \cdot A_j^T}{\|A_i\|\|A_j\|} = \cos(A_i, A_j) < Threshold \cdot \|A_j\|$

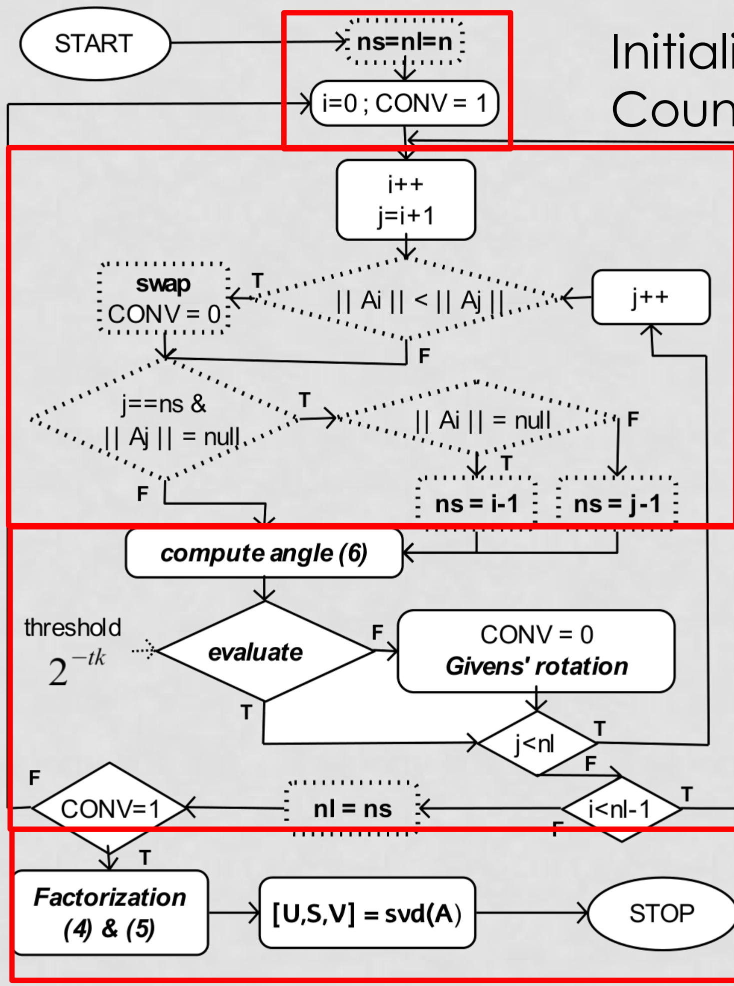
IMPROVING PREVIOUS WORK BEING HW FRIENDLY

UPV EHU

- Initially Two Angles Calculation:
  - The Decision → f(cosine)
  - The Rotation → f(Rutishauser)
- Cos & RutisHauser → both f(columns and its norms)
- Killing two bird with one stone Decision and rotation → f(RutisHauser)
- Readaptation → More sensitive
- AARH: $\theta_{ij} < Threshold \cdot \|A_j\|^2$
- Avoiding root squares

RESULTS
THE MODIFIED ONE-SIDED JACOBI

- -Motivation
- -Selecting the Algorithm
- -Selecting SVD Method
- -Speaking About the Accuracy
- -Improving Previous Work
- **-Results**
- - HW FPGA Implementation
- -Why and How Scalable?
- -Conclusion
- -Future Work



Initialization: Problem Size, Flag & Counters

Iteration : Swap &Null Columns Management

Iteration : Decision, Rotation & Actualization

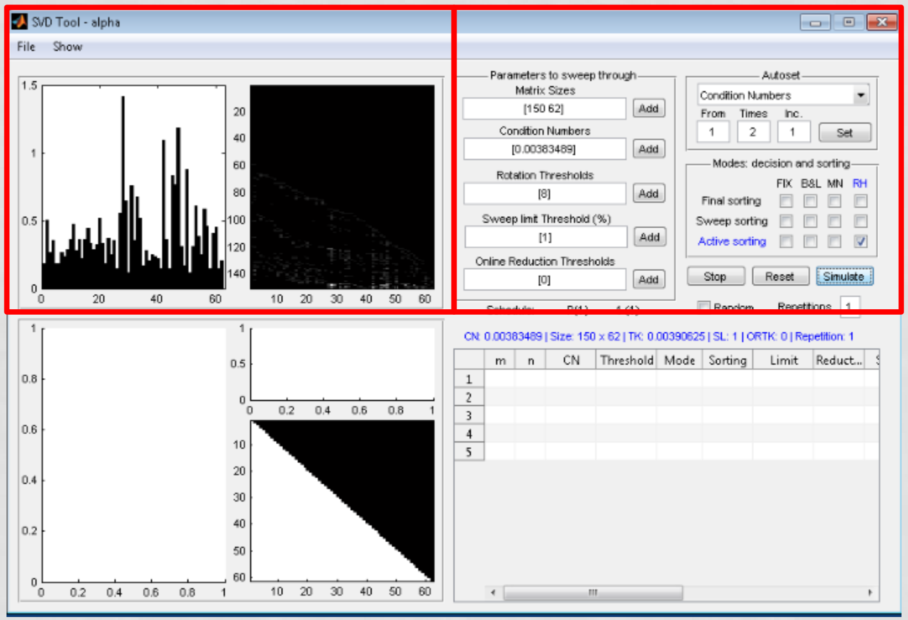
Finish: Matrix Factorization

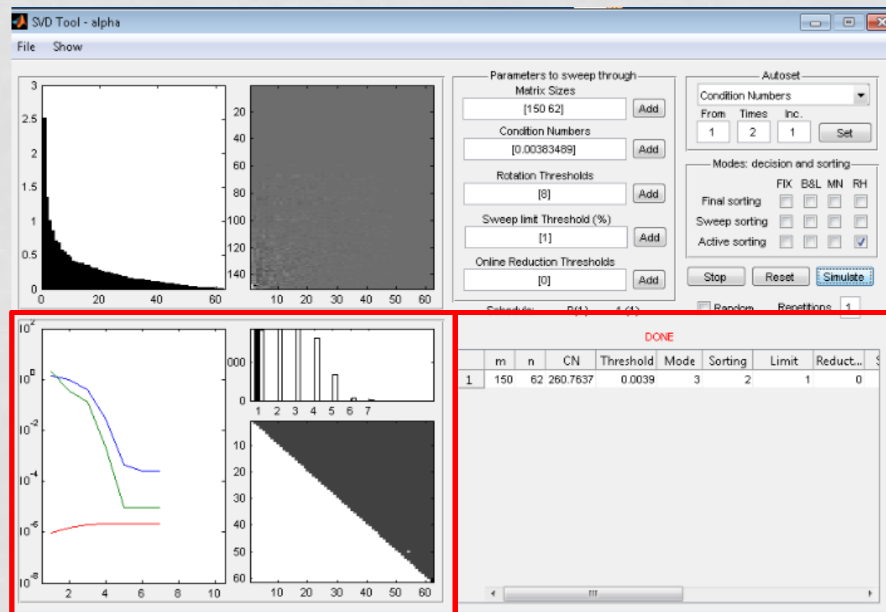
$$\sigma_i = \|W_{(:,i)}\| \qquad U_{(:,i)} = \frac{W_{(:,i)}}{\sigma_i}$$

COMPARATION ANALYSIS TOOLBOX

Column Arrangement

Selection of the Comparation





Error Evolution

Results

RESULTS
COMPARING WITH THE REST

UPV EHU

- -Motivation
- -Selecting the Algorithm
- -Selecting SVD Method
- -Speaking About the Accuracy
- -Improving Previous Work
- **-Results**
- - HW FPGA Implementation
- -Why and How Scalable?
- -Conclusion
- -Future Work

TABLE I: Comparative of algorithm performance

	Inverse Error | Threshold			
Fixed	4.19E-6 | 24	2.11E-5 | 24	2.54E-4 | 24	1.89E-3 | 24
B&L	3.98E-6 | 22	1.93E-5 | 18	1.62E-4 | 16	1.87E-5 | 16
ABL	3.68E-6 | 22	1.83E-5 | 18	1.63E-4 | 16	1.78E-3 | 12
AAMN	5.60E-6 | 20	1.76E-5 | 16	1.68E-4 | 10	1.34E-3 | 8
AARH	6.77E-6 | 20	1.87E-5 | 16	1.76E-4 | 10	1.41E-3 | 8
$\kappa(A)$	**1,00E+01**	**1,00E+02**	**1,00E+03**	**1,00E+04**
Fixed	33,696 | 11.3	33,530 | 10.50	33,905 | 10.50	34,004 | 10.75
B&L	32,303 | 9.70	30,911 | 10.20	29,858 | 10.15	26,482 | 10.13
ABL	26,096 | 8.20	24,356 | 8.00	23,121 | 8.05	19,927 | 7.73
AAMN	25,275 | 8.13	23,209 | 8.05	18,353 | 8.05	15,327 | 8.30
AARH	25,512 | 8.40	23,345 | 8.50	18,960 | 8.53	16,078 | 8.35
	Rotations | Sweeps			

Obtaining Same Accuracy → Easier Threshold

Obtaining Same Accuracy → Less Rotations

Obtaining a Better Result → The Higher the K(A)

TABLE II
TEST MATRICES RESULTS

Inverse Error | $2^{-Threshold}$; Rotations | Sweeps

Size | κ Name	201 × 47 | 7.5E1 JGD_Kocay/Trec9	235 × 216 | 1.7E3 JGD_Forest/TF11	200 × 200 | 2.4E3 Bai/bwm200
ABL	1.99E6 | 22 5,004 | 7	1.4E-3 | 12 97,281 | 9	2.96E4 | 14 95,902 | 9
AAMN	1.96E-6 | 18 4,903 | 8	5.3E4 | 6 75,433 | 9	1.43E4 | 8 61,981 | 9
AARH	2.21E6 | 16 4,908 | 9	8.03E-4 | 4 76,854 | 9	5.64E-4 | 6 57,675 | 10

Testing With Real Matrices & Obtaining Expected Results

RESULTS
COMPARING WITH THE REST

- Savings in Number of Rotations

CN	Fixed to AARH	B&L to AARH	ABL to AARH
1,00E+01	24,29%	21,02%	2,24%
1,00E+02	30,38%	24,48%	4,15%
1,00E+03	44,08%	36,50%	18,00%
1,00E+04	52,72%	39,29%	19,32%

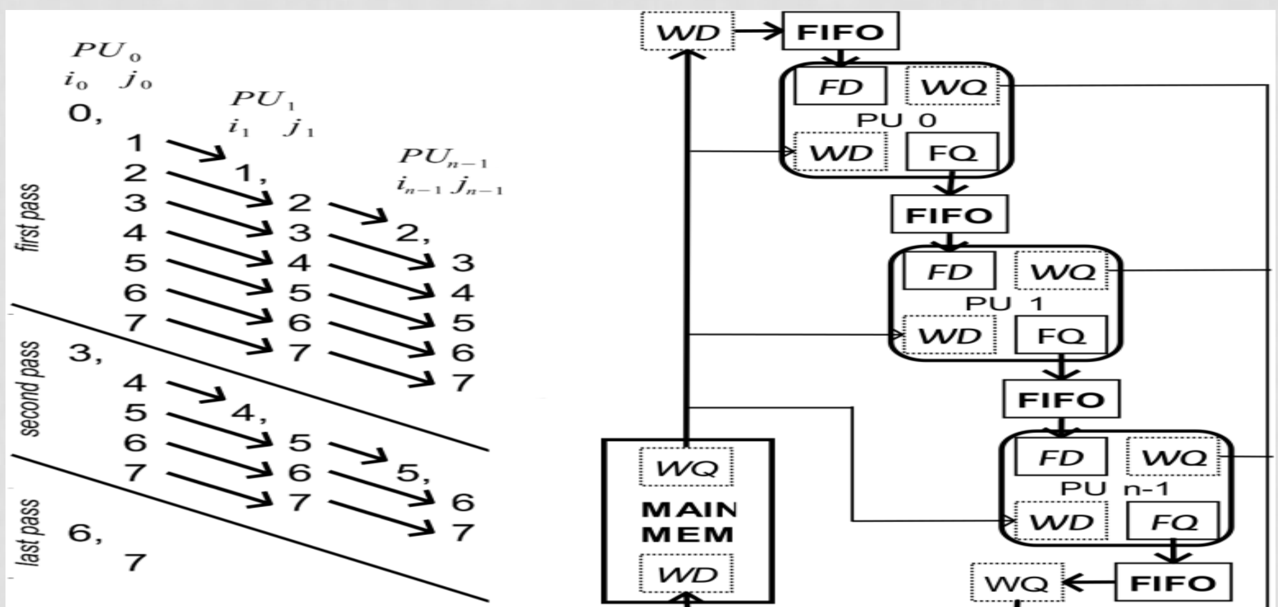
HW FPGA IMPLEMENTATION ARCHITECTURE

UPV EHU

-Motivation
-Selecting the Algorithm
-Selecting SVD Method
-Speaking About the Accuracy
-Improving Previous Work
-Results
- HW FPGA Implementation
-Why and How Scalable?
-Conclusion
-Future Work

Double Data-Flow:

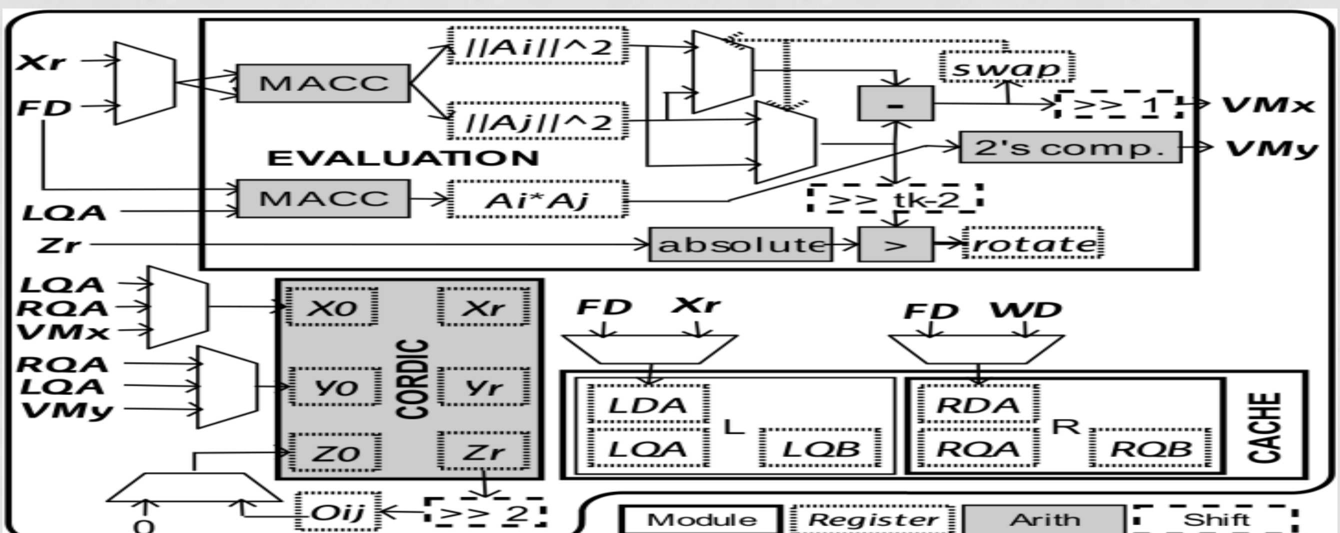
- Primary: Linear array to manage Ai/Aj
- Secondary: Asynchronous full-duplex shared bus to manage Vi/Vj
- FIFO between PUs



HW FPGA IMPLEMENTATION PROCESSING UNIT

PU Design:

- Evaluation: Computing square Euclidean norms and vector multiplication, swaping and deciding
- Cordic: Theta calculation and rotations' performing
- Cache: L(m+n) and R(max(m,2n)

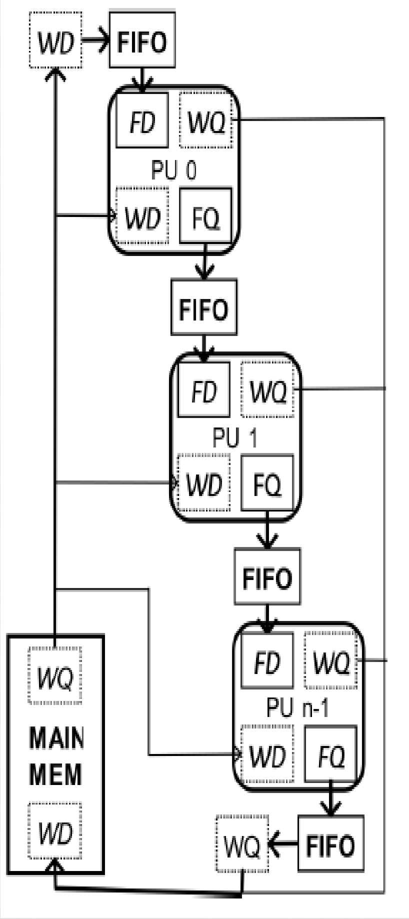


WHY AND HOW SCALABLE?

UPV EHU

-Motivation
-Selecting the Algorithm
-Selecting SVD Method
-Speaking About the Accuracy
-Improving Previous Work
-Results
- HW FPGA Implementation
-Why and How Scalable?
-Conclusion
-Future Work

- No limited to specific matrices and HW: Generic Solution
  - Different sizes
  - Different Shapes
  - Different budgets
- Architecture
  - Based on basics processing units PUs
  - PUs variable on quantity
  - From 2 to n/2



IMPLEMENTATION RESULTS

	xc6slx45-3fgg484	xc7k160t-3fbg484
Area	56 %	86 %
DSPs	93 %	60 %
RAM	62 %	44 %
Matrix Size ; K(A)	300x100 ; 10^2	750 x 250
PUs	9	60
Frequency	55 MHz	90 MHz
Processing Time	60 ms (5-6 sweeps, 8-16 ms/sweep)	

Word-Length :18 bits

CONCLUSION

- AAMN and AARH proposed outperforming previous proposals.
  - Small Columns Important Columns
  - Same Accuracy Less Rotations
  - User-defined Accuracy -> Threshold
  - HW Friendly
- An implemented parallell processing scheme proposed:
  - Linear Array of PUs
  - Scalable
  - Double Data-Flow

FUTURE WORK

- Online reduction of problem size
- Improve sorting
- Optimize PU design
  - Improved CORDIC realization (Redundant arithmetic (Ercegovac) or Square root and division free (Gotze))
  - Ad-hoc online estimators
    - Atan
    - Square norm

THANK YOU VERY MUCH

- Questions or Details ?