High-Dynamic-Range, Real-Time Architecture for Robust Space-Time Synchronization Algorithm

Vitaliy Gleyzer*, Tom Baillio, Adam Margetts

{vgleyzer, tom.baillio, margetts} @ll.mit.edu

MIT Lincoln Laboratory

HPEC Workshop 2011

1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

This work is sponsored by the United States Air Force under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

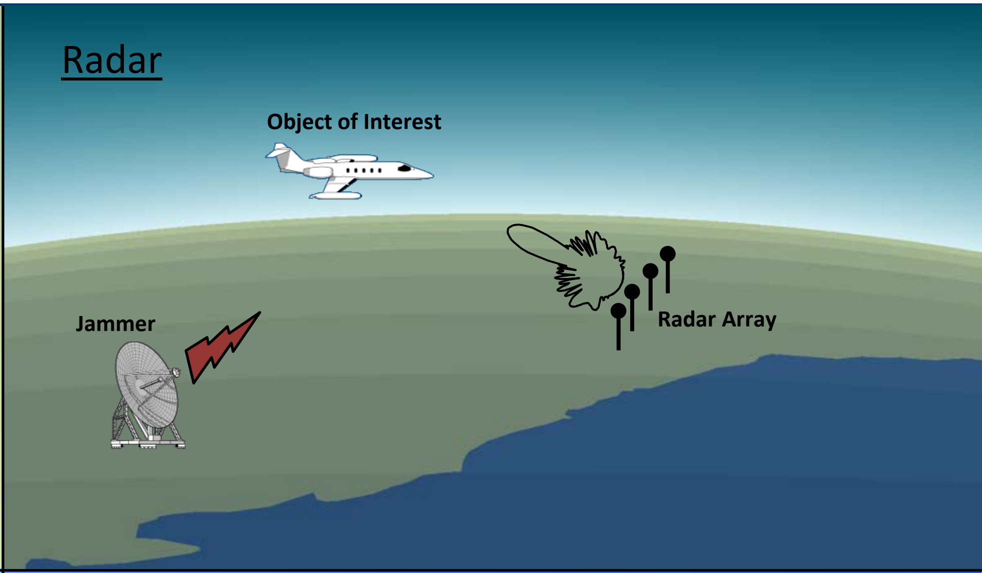
Adaptive System Building Blocks

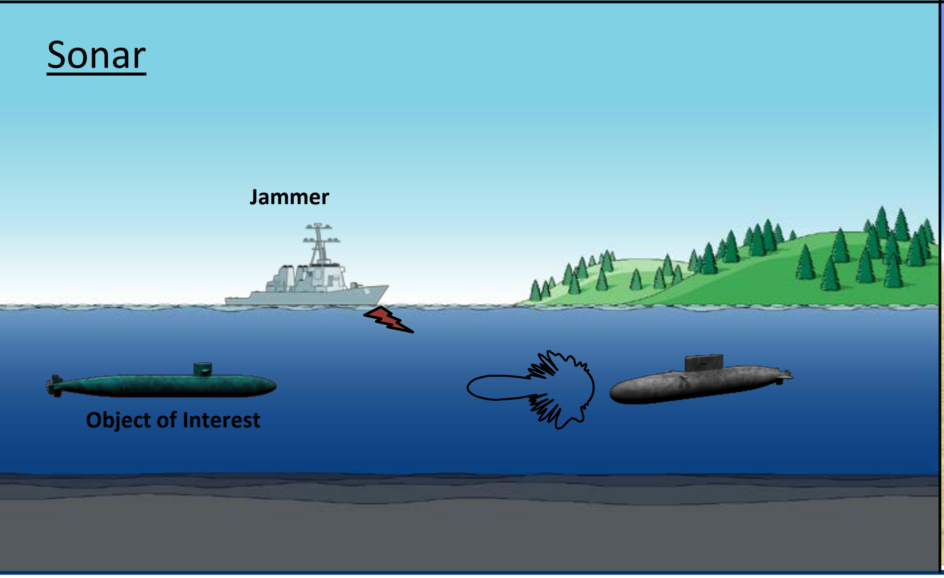
Space-Time adaptive signal processing

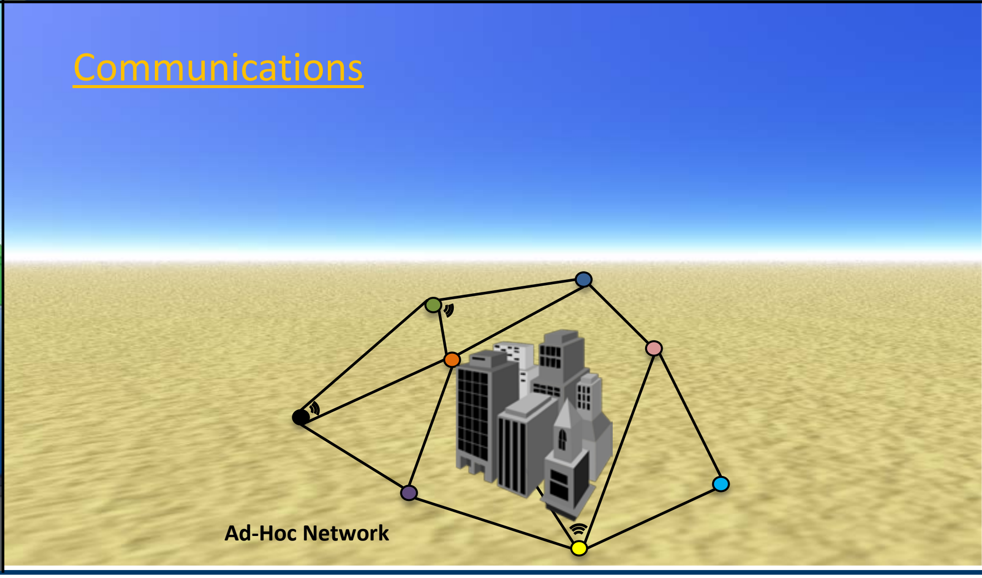
- Ideal for dynamic environments
- Numerous real-time applications

Common signal processing building blocks:

C = AB	Matrix Multiplication
Z = QR	QR Decomposition
C = A / B	Back/Forward Substitution
…	

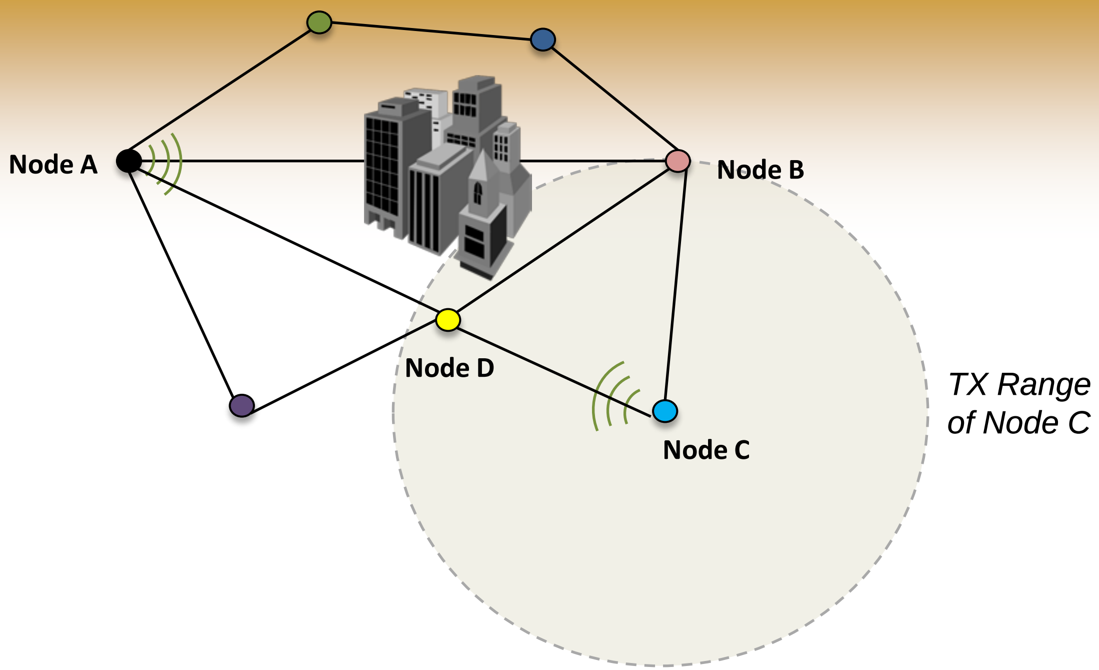






1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Interference in Ad-hoc Networks

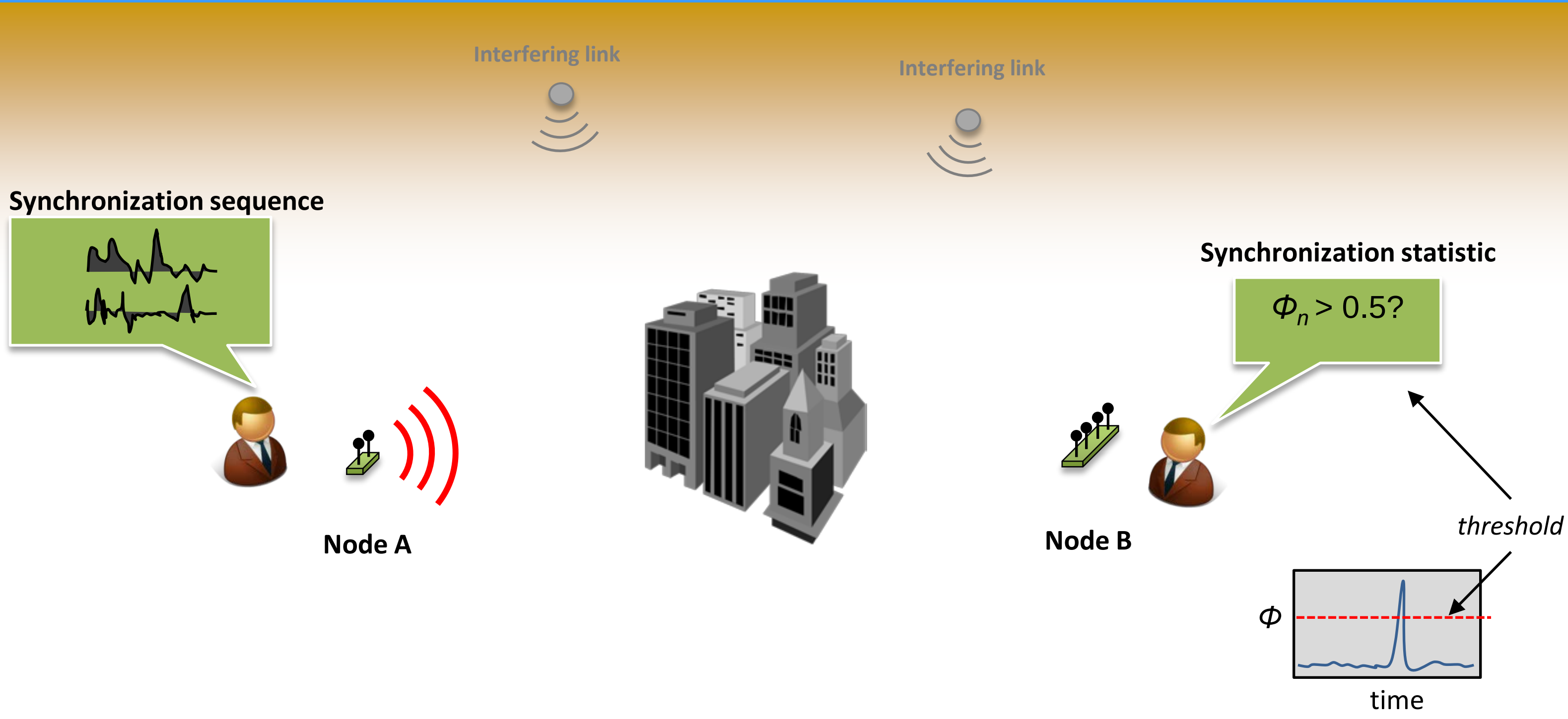


Wireless nodes compete for channel resources and interfere with each other

Outline

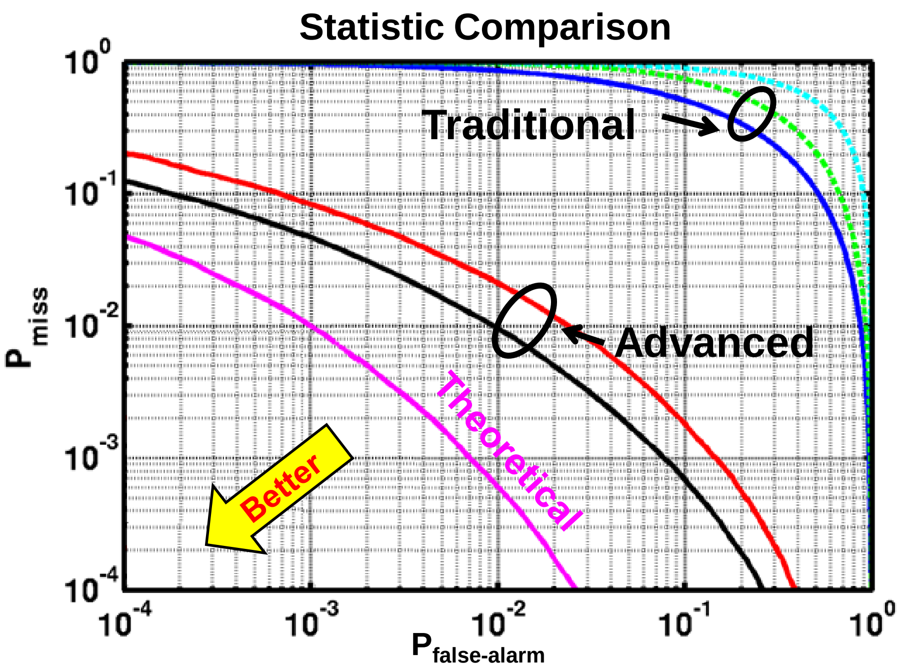
- **Time Synchronization**
  - Overview
  - Algorithm comparison
- **Implementation Details**
  - Performance results
  - Architecture overview

Synchronization in Time



Time synchronization is a fundamental requirement for communication

Sync Statistic Performance in Presence of Interference

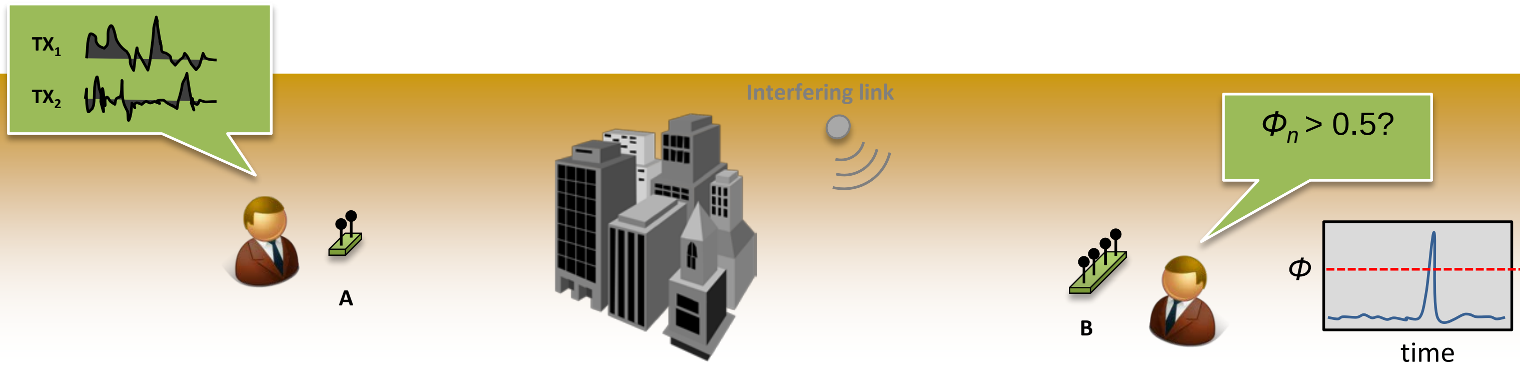


Performance Bounds
- Rand – Random guess
- NP – Theoretical bound

Traditional Techniques
- Cor – Correlator
- LSE – Least Squared Channel Estimator

Advanced Techniques
- MMSE – Minimum Mean Squared Error
- GLRT – Generalized Likelihood Ratio Test

System Parameter	Value
Interference-To-Noise Ratio	10 dB
Signal-To-Noise Ratio	2 dB
# of Interferers	1



GLRT Synchronization Metric

Synchronization Metric

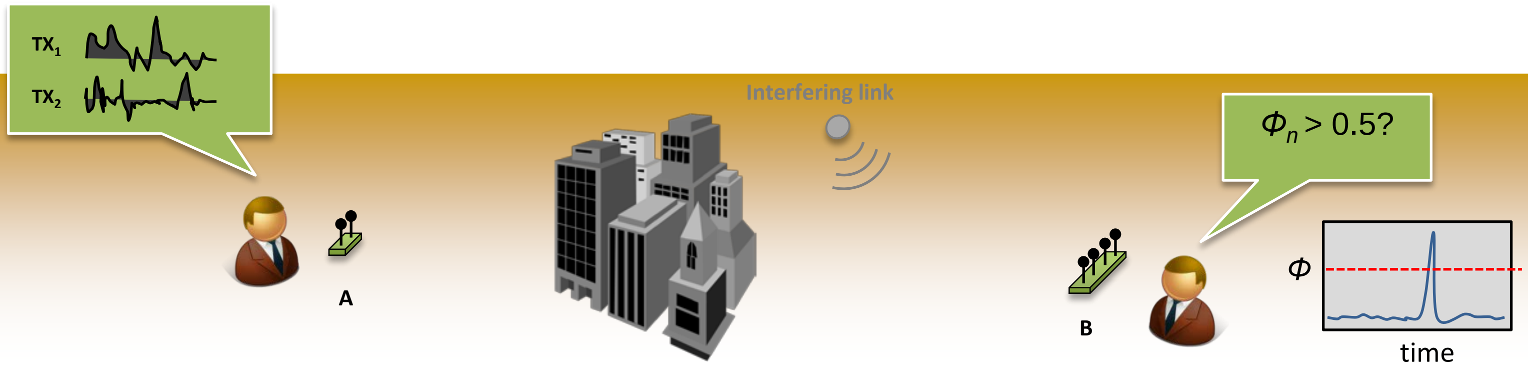
$$\phi_n = \left| I - S Z_n^{\mathrm{H}} \left(Z_n Z_n^{\mathrm{H}} \right)^{-1} Z_n S^{\mathrm{H}} \right|$$

Expected Sequence (Normalized)

Sample Matrix

Reference

D. W. Bliss and P. A. Parker, "Temporal synchronization of MIMO wireless communication in the presence of interference," *IEEE Trans. on Signal Processing* Mar. 2010

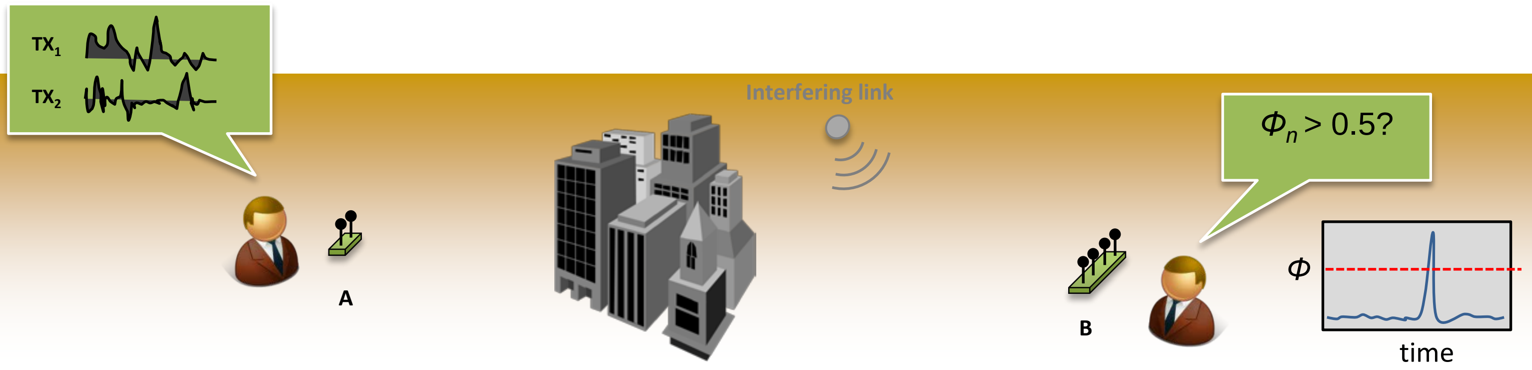


1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Sample Matrix Formation

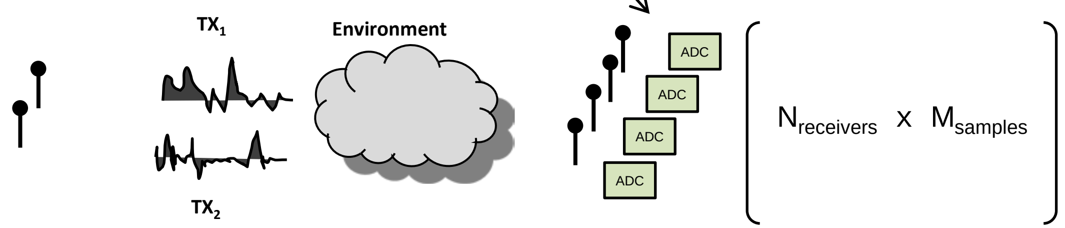
$$\phi_n = \left| I - S Z_n{}^{\mathrm{H}} \left(Z_n Z_n{}^{\mathrm{H}} \right)^{-1} Z_n S^{\mathrm{H}} \right|$$

Spatial diversity is captured via multiple receive antennas

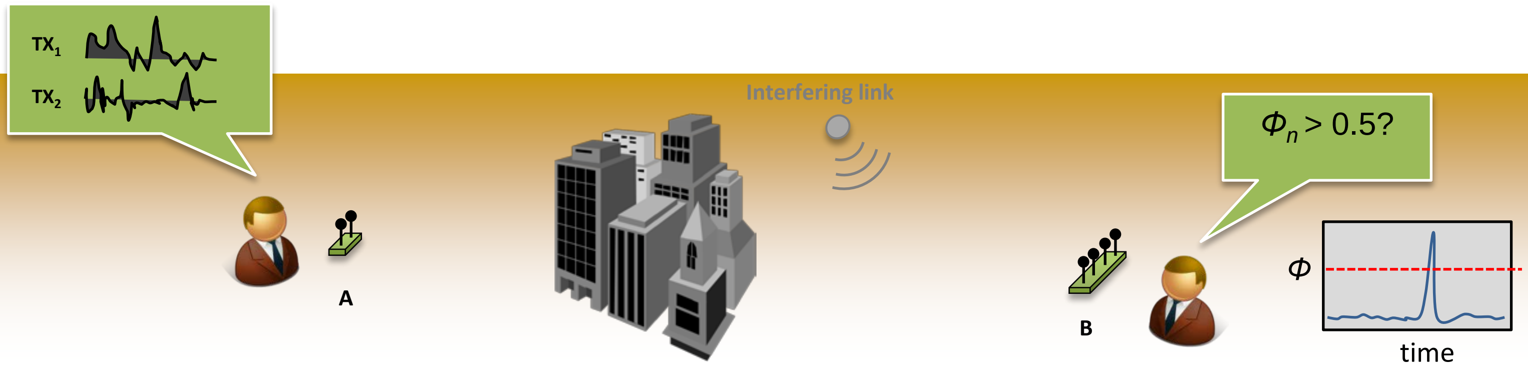


Sample Matrix Formation

$$\phi_n = \left| I - S Z_n^{H} \left(Z_n Z_n^{H} \right)^{-1} Z_n S^{H} \right|$$



Spatial diversity is captured via multiple receive antennas



Outline

- Time Synchronization Overview
- **Implementation Details**
  - Performance results
  - Architecture overview

Implementation Challenges and Strategy

Challenge	Approach
Dynamic Environment	Single-Precision Floating Point
Real-Time Processing	Highly Parallel Computational Platform
Mobile Profile	Embedded Form Factor

Xilinx Virtex 6 FPGA

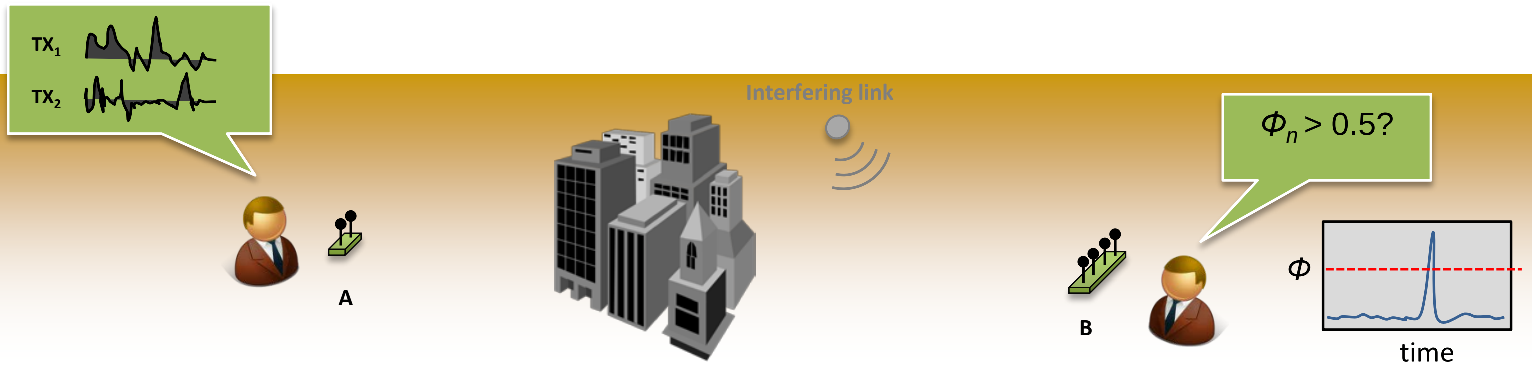
Virtex6 LX550 Resources	Quantity
6-input LUTs	343,680
DSP Blocks	864
BRAM	22 Mbits
Registers	687,360

Algorithm Reformulation

$$\Phi_n = | I - S Z_n^{H} (Z_n Z_n^{H})^{-1} Z_n S^{H} |$$

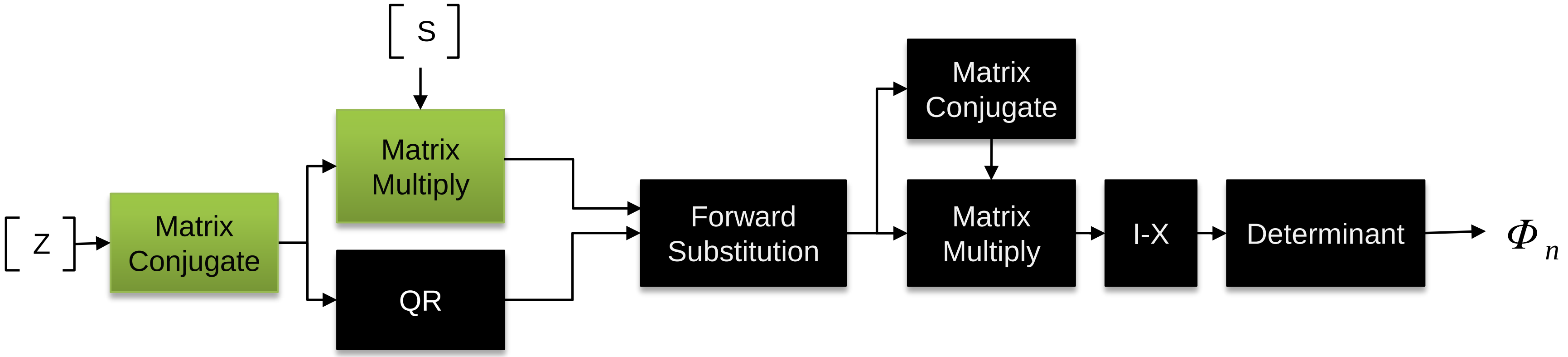
Z^H = QR, where Q is a unitary matrix, and R is upper triangular

$$\Phi_n = | I - (S Z_n^{H} R^{-1})(R^{-H} Z_n S^{H}) |$$



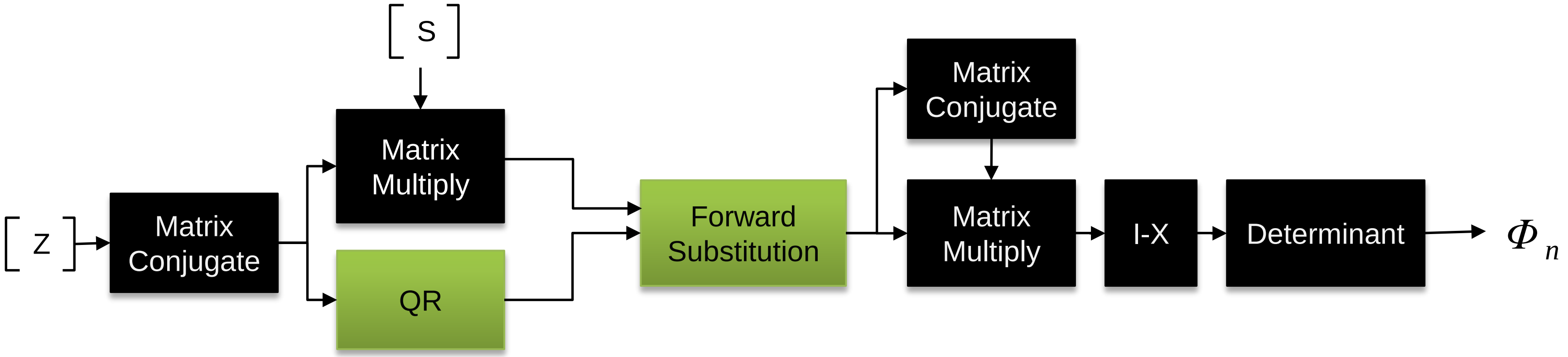
Data Flow Overview

$$\Phi_n = | I - (SZ_n^{\mathrm{H}} R^{-1})(R^{-\mathrm{H}} Z_n S^{\mathrm{H}}) |$$



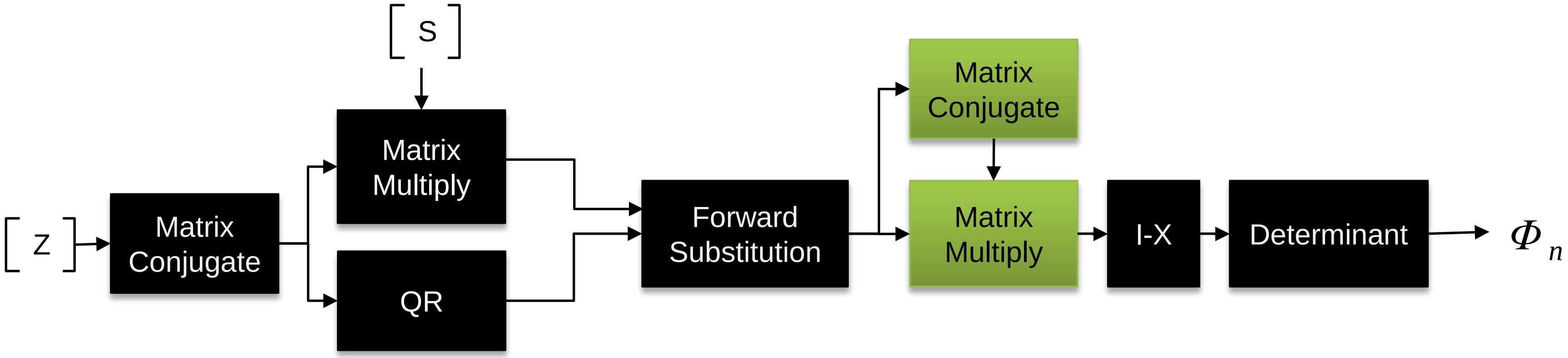
Data Flow Overview

$$\Phi_n = |\, I - (S Z_n{}^{H} R^{-1})(R^{-H} Z_n S^{H}) |$$



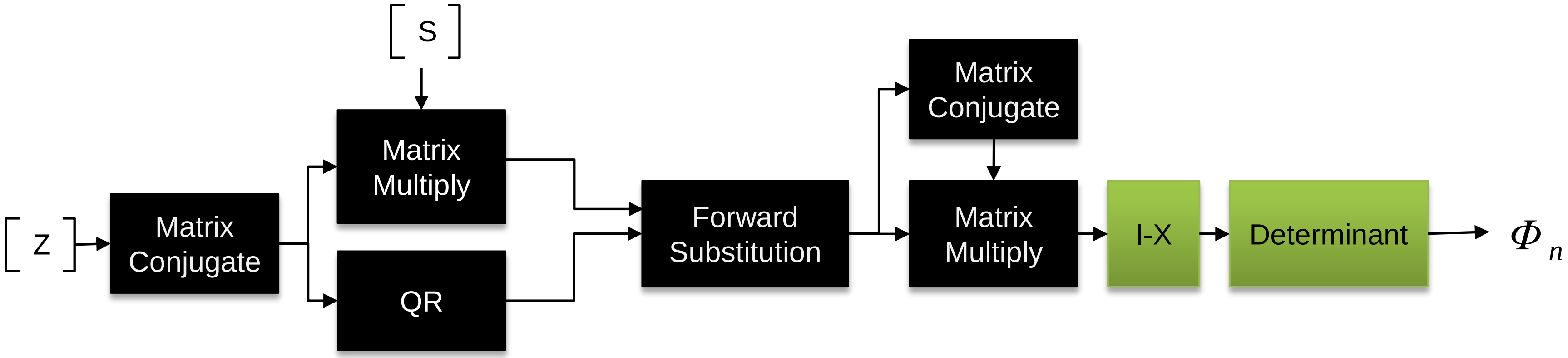
Data Flow Overview

$$\Phi_n = |\, I - (SZ_n{}^{H}R^{-1})(R^{-H}Z_nS^{H})|$$



Data Flow Overview

$$\Phi_n = | I - (SZ_n^{\mathrm{H}}R^{-1})(R^{-\mathrm{H}}Z_nS^{\mathrm{H}})|$$



Operation Complexity

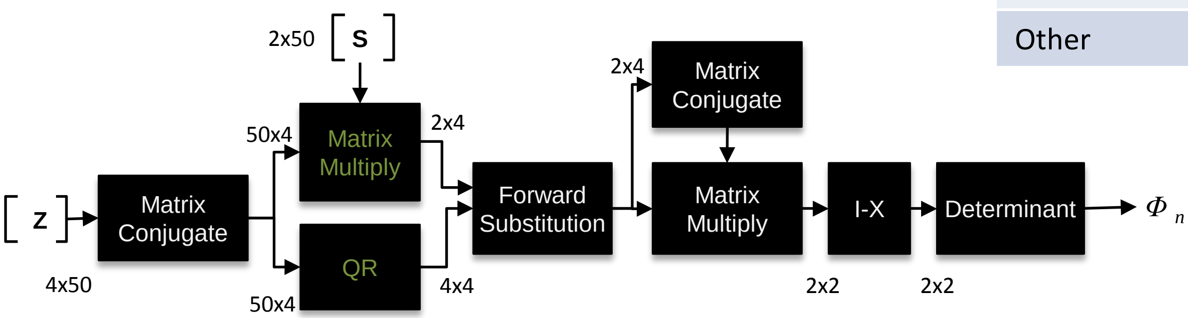
$$\Phi_n = | I - (SZ_n^H R^{-1})(R^{-H} Z_n S^H) |$$

Realistic Configuration:

Parameter	Value
Observation Size ($N_{samples}$)	50 samples
Number of Receivers($N_{receivers}$)	4
Number of Transmitters($N_{transmitters}$)	2

Operation Breakdown:

Computation	Operations (MUL/ADD)
QR Decomposition (50x4)	9200
Matrix Multiply (2x50 * 50x4)	2800
Matrix Multiply (2x4 * 4x2)	120
Forward Substitution (4x4)	34
Other	32



~12,000 arithmetic OPs

Complexity dominated by QR Decomposition

1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Implementation Performance

Realistic Configuration:

Parameter	Value
Observation Size ($N_{samples}$)	50 samples
Number of Receivers($N_{receivers}$)	4
Number of Transmitters($N_{transmitters}$)	2

Performance of Virtex6 LX550 Implementation:

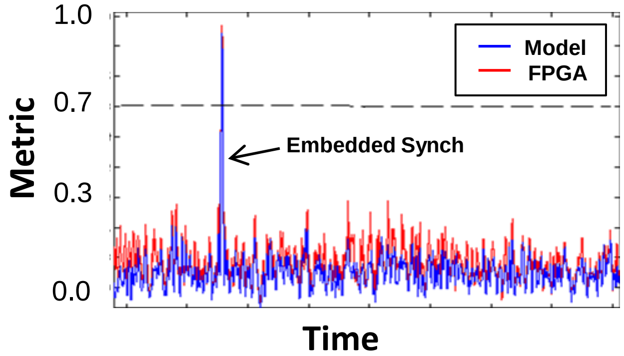
Computation Throughput	36 GFLOPS
Computation Throughput / Watt	2.4 GFLOPS/W

Synchronization Rate	2.5 MHz

Resource Utilization:

Resource Name	Utilization (%)
LUTs	45%
DSP Blocks	6%
BRAM	3%
Registers	6%

Example Output



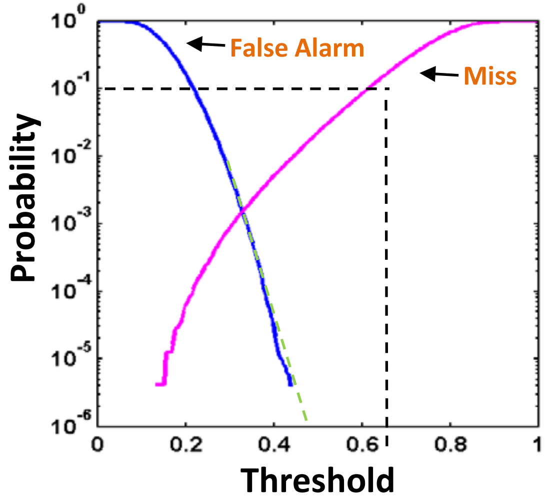
Feasible rates for networks that can stream multimedia content

Performance Plots

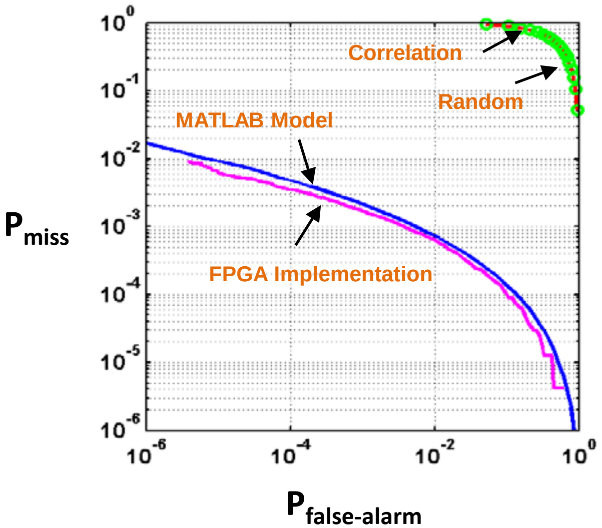
Environment:

Parameter	Value
Interference-To-Noise Ratio	80 dB
Signal-To-Noise Ration	0 dB
Number of Interferers	2

Threshold Trade-Off

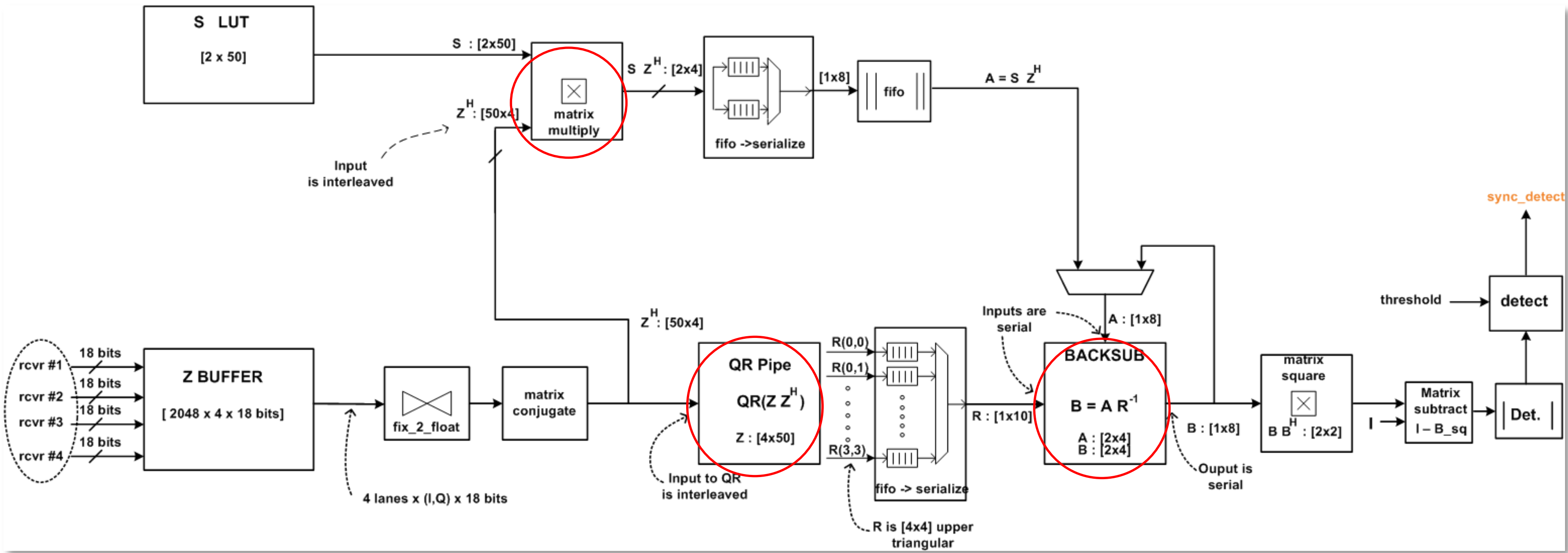


Algorithm Comparison



Vastly outperforms traditional techniques in presence of interference

Initial Sync Architecture

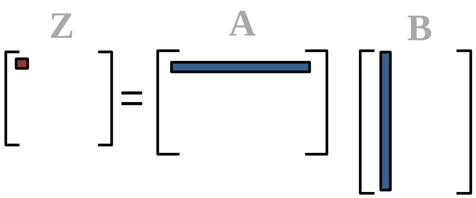


Three main component: QR, Matrix Multiply, and Forward Substitution

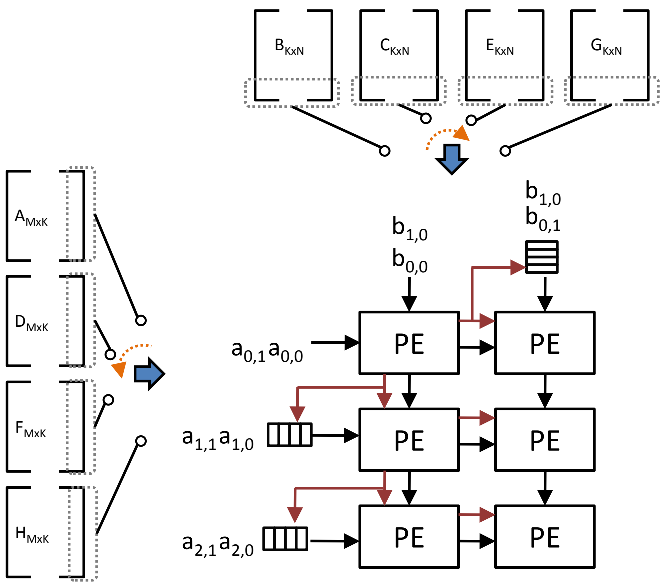
Systolic Matrix Multiplier

Operation

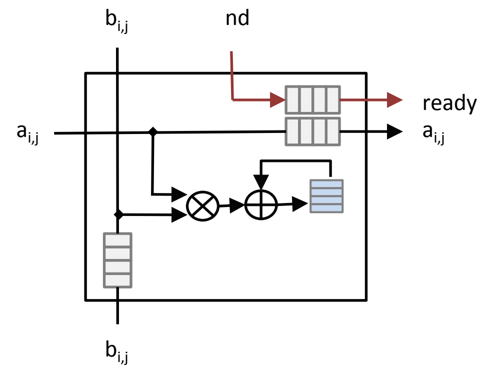
$$\mathbf{Z}_{MxN} = \mathbf{A}_{MxK} \cdot \mathbf{B}_{KxN}$$



Architecture



Processing Element(PE)

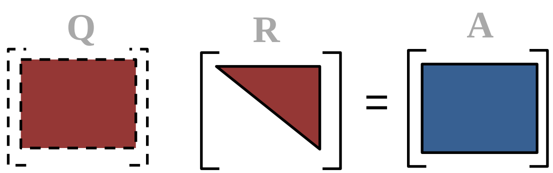


- **Single precision floating-point**
- **Parameterized result matrix size**
- **Fully pipelined**
  - *Accepts new data row every clock cycle*
- **Multiple interleave windows**
  - *Allows overlapping multiple, simultaneous matrix multiplication*

Systolic QR Decomposition

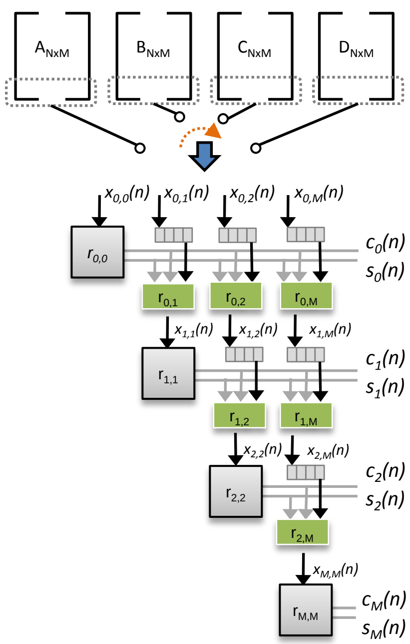
Operation

$$\mathbf{R}_{NxN} = qr(\mathbf{A}_{MxN})$$



(Unitary matrix Q is not explicitly computed)

Architecture



Processing Elements(PE)

$$r_{ij}(0) = \begin{cases} \sqrt{\alpha} & i = j \\ 0 & i \neq j \end{cases}$$

Boundary cell (grey):

$$\begin{cases} r_{ii}(n) = \sqrt{r_{ii}^2(n-1) + |x_{i,i}(n)|^2} \\ c_i(n) = \dfrac{r_{ii}(n-1)}{r_{ii}(n)} \\ s_i(n) = \dfrac{x_{i,i}(n)}{r_{i,i}(n)} \end{cases}$$

Internal cell (green):

$$\begin{cases} r_{ij}(n) = c_i r(n-1) + s_i^* x_{i,j}(n) \\ x_{i+1,j}(n) = -s_i r_{ij}(n-1) + c_i x_{i,j}(n) \end{cases}$$

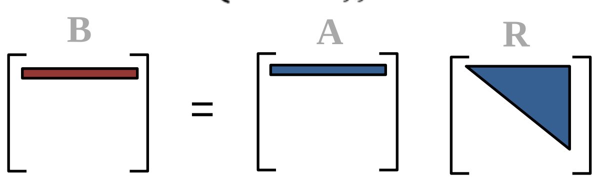
- **McWhirter/Givens array**
- **Single precision floating-point**
- **Parameterized result matrix size**
- **Fully pipelined**
  - *Accepts new data row every clock cycle*
- **Multiple interleave windows**
  - *Allows overlapping multiple, simultaneous QR computations*

1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

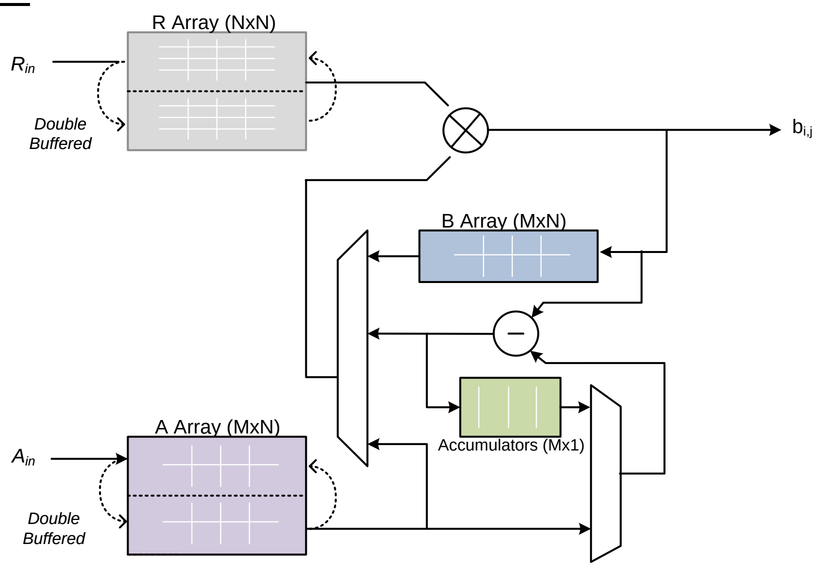
Forward Substitution

Operation

$$b_{i,j} = \begin{cases} \frac{a_{i,j}}{r_{j,j}}, & j = 0 \\ \frac{a_{i,j} - \sum_{k=0}^{j-1} b_{i,k} r_{k,j}}{r_{j,j}}, & j > 0 \end{cases}$$



Architecture

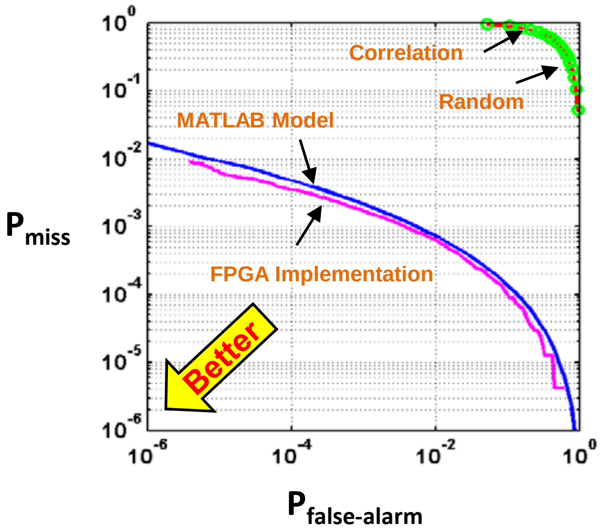


- Single precision floating-point
- Parameterized size of input matrices
- Interleaves multiple rows for greater throughput

Summary and Future Work

- Successfully implemented real-time space-time adaptive synchronization algorithm in FPGA
- Performance verified to be vastly superior to less sophisticated techniques
- Implemented high-performance blocks to be utilized for other integral communication components
- Future Work:
  - Implementing other components of the communication sub-system:
    - Channel capacity estimation
    - Frequency offset estimation
    - Beamforming
  - Constructing complete ad-hoc network node environment

Algorithm Comparison



Acknowledgements

- Dan Bliss
- Huy Nguyen