University of Montenegro

Faculty of Electrical Engineering

SINGLE ITERATION ALGORITHM HARDWARE IMPLEMENTATION

Prof. dr Srdjan Stanković
Prof. dr Nedjeljko Lekić
Prof. dr Miloš Daković
Prof. dr Irena Orović
MScAndjela Draganić

ALGORITHM STEPS

$$x(n)=\sum_{i=1}^{K}A_i e^{j\frac{2\pi k_i n}{N}}$$

- Signal definition

$$\mathrm{var}=M\,\frac{N-M}{N-1}\,(A_1^2+A_2^2+\ldots+A_K^2)$$

- Missing samples noise variance
- N-signal length
- M-number of available samples
- K-number of signal components

$$T=\frac{1}{N}\left(-\mathrm{var}^2\log(1-\sqrt[N]{P})\right)^{1/2}$$

- Threshold

$$V(f)=\sum_{m=1}^{M}v(m)e^{-j\frac{2\pi fm}{N}},\ f=1,\ldots,N$$

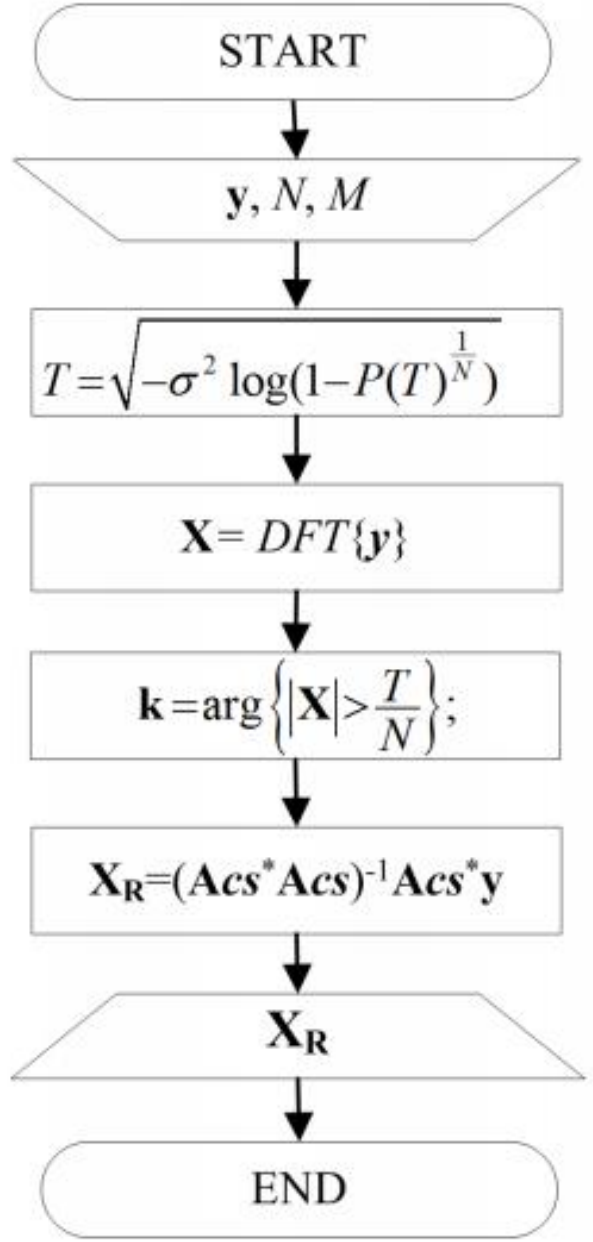
-Vector of initial DFT

$v(m)\rightarrow$ Vector of available signal samples

$$pos=\arg\left\{|V|>T\right\}$$

- Finding the positions above the threshold

ALGORITHM



START
y, N, M
$T = \sqrt{-\sigma^2 \log(1 - P(T)^{\frac{1}{N}})}$
$\mathbf{X} = DFT\{\mathbf{y}\}$
$\mathbf{k} = \arg\left\{|\mathbf{X}| > \frac{T}{N}\right\};$
$\mathbf{X_R} = (\mathbf{A}cs^*\mathbf{A}cs)^{-1}\mathbf{A}cs^*\mathbf{y}$
$\mathbf{X_R}$
END

ALGORITHM STEPS

Forming CS matrix :

$A_{N\times N}$ -DFT matrix

-**Rows**-positions of random measurements

$A_{CS_{M\times K}}$ -CS matrix

-**Columns**-positions of the V with amplitudes (absolute values) above the threshold

The optimization problem:

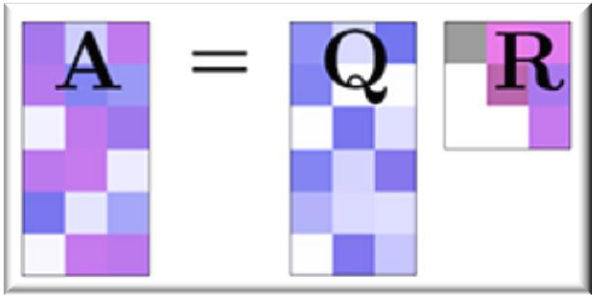
Hard to implement in hardware

$$X = (A_{CS}^{*} A_{CS})^{-1}(A_{CS}^{*} v)$$

QR decomposition → $A_{CS} = Q_{CS} R_{CS}$

QR Decomposition methods

- There are several methods for QR decomposition calculation:
  - Gram-Schmidt decomposition;
  - Householder transformation and
  - **Givens rotations**



- QR based on Givens rotations can be parallelized and has a low computational complexity

$$A = QR,$$

$$R = Q^T A,$$

$$Q^T Q = I.$$

- A-real or complex matrix; square or rectangular matrix
- Q-an orthogonal matrix; R - right triangular matrix

QR Decomposition

- Givens rotation matrix: $$G=\begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$
  - $c^2 + s^2 = 1$, $c = \cos(\theta)$ i $s = \sin(\theta)$

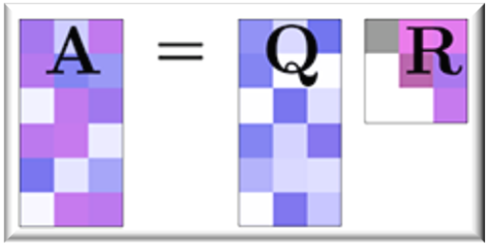
$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix}=\begin{bmatrix} \sqrt{a^2+b^2} \\ 0 \end{bmatrix}, \qquad c=\frac{a}{\sqrt{a^2+b^2}}, \qquad s=\frac{b}{\sqrt{a^2+b^2}}$$

  - General form of Givens rotation matrix:

$$G(i,j,\theta)=\begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 0 & \cdots & C & \cdots & S & \cdots & 0 \\ \vdots & \cdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ 0 & \cdots & -S & \cdots & C & \cdots & 0 \\ \vdots & \cdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}\begin{matrix} \\ \\ i \\ \\ j \\ \\ \\ \end{matrix}$$

$$\qquad\qquad\qquad i \qquad\qquad j$$

Algorithm steps

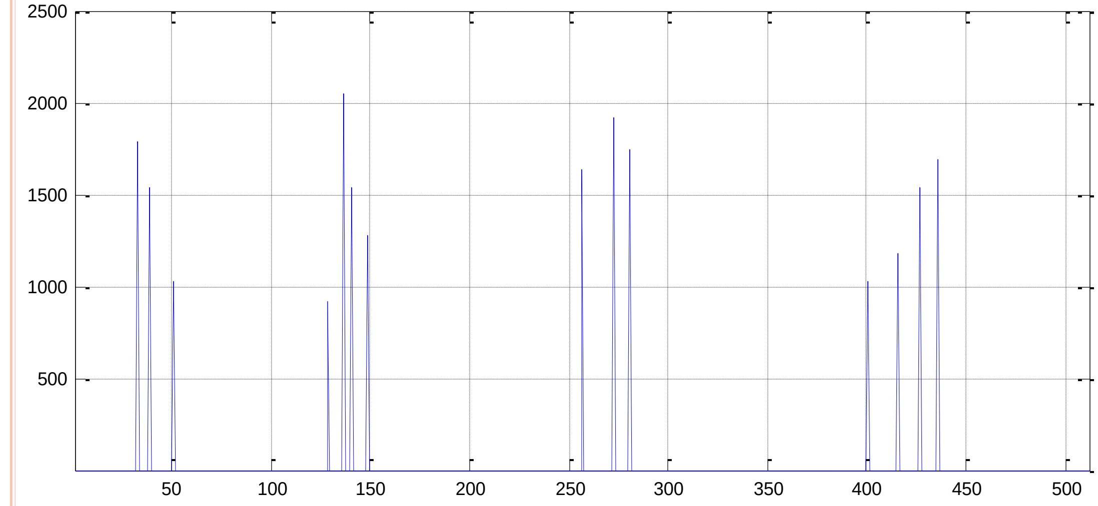


QR decomposition → $A_{CS} = Q_{CS} R_{CS}$

$$X = (A_{CS}^{*} A_{CS})^{-1} (A_{CS}^{*} v) \rightarrow X = \left((Q_{CS} R_{CS})^{*} (Q_{CS} R_{CS}) \right)^{-1} (A_{CS}^{*} v)$$

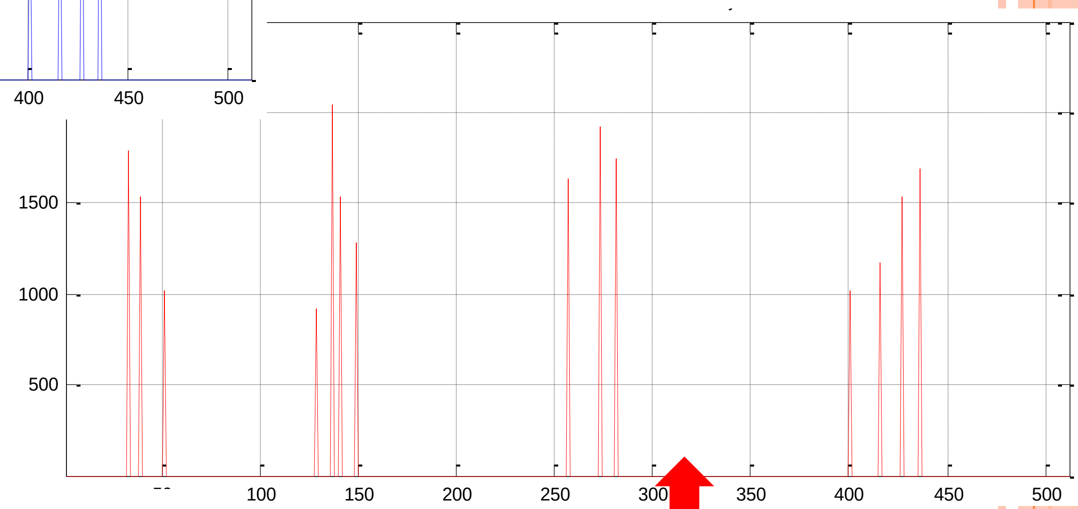
$$X = \left(R_{CS}{}^{*} Q_{CS}{}^{*} Q_{CS} R_{CS} \right)^{-1} (A_{CS}^{*} v) = (R_{CS}^{*} R_{CS})^{-1} (A_{CS}^{*} v)$$

$$X = R_{CS}^{-1} (R_{CS}{}^{-1})^{*} \cdot A_{CS}^{*} v$$

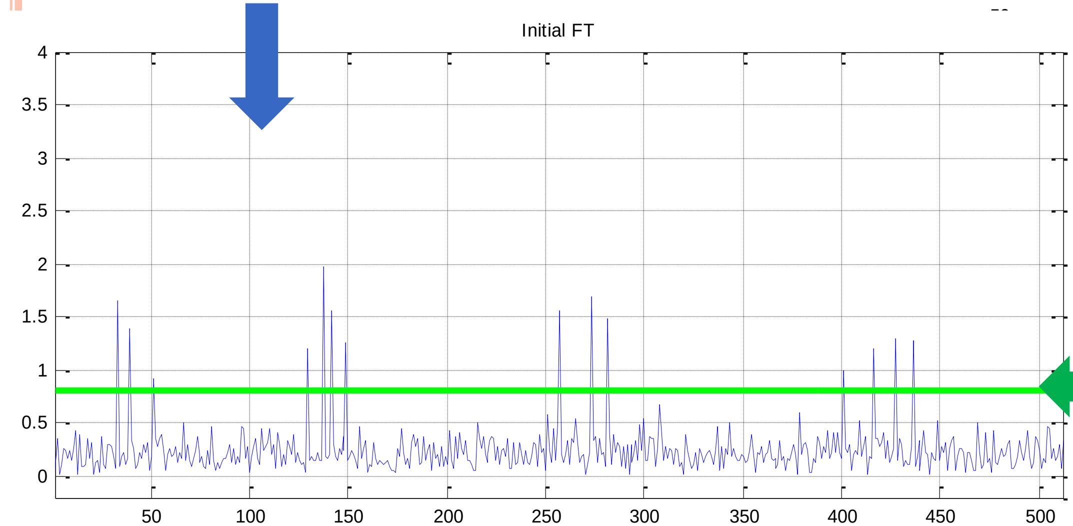


Original Fourier transform

220 out of 512 samples used



Initial Fourier transform



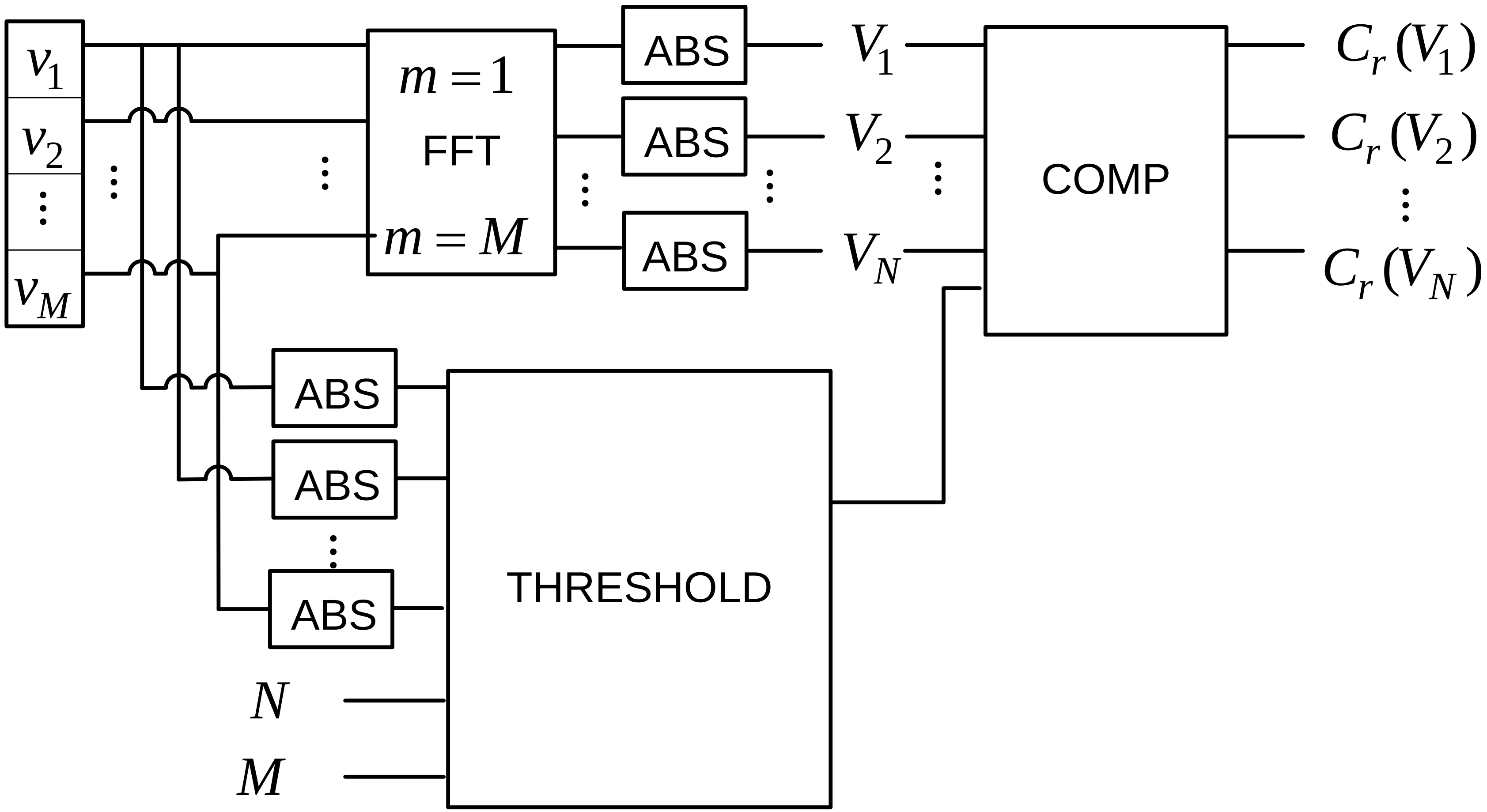
Reconstructed Fourier transform

Threshold

Block scheme

- *Part 1*: FFT calculation using the available signal samples and finding the positions of the FFT coefficients that are above the threshold;
- *Part 2*: Forming of the Compressive Sensing matrix;
- *Part 3*: QR decomposition and optimization problem solving;
- *Part 4*: Spectral positioning block

BLOCK SCHEME – PART 1

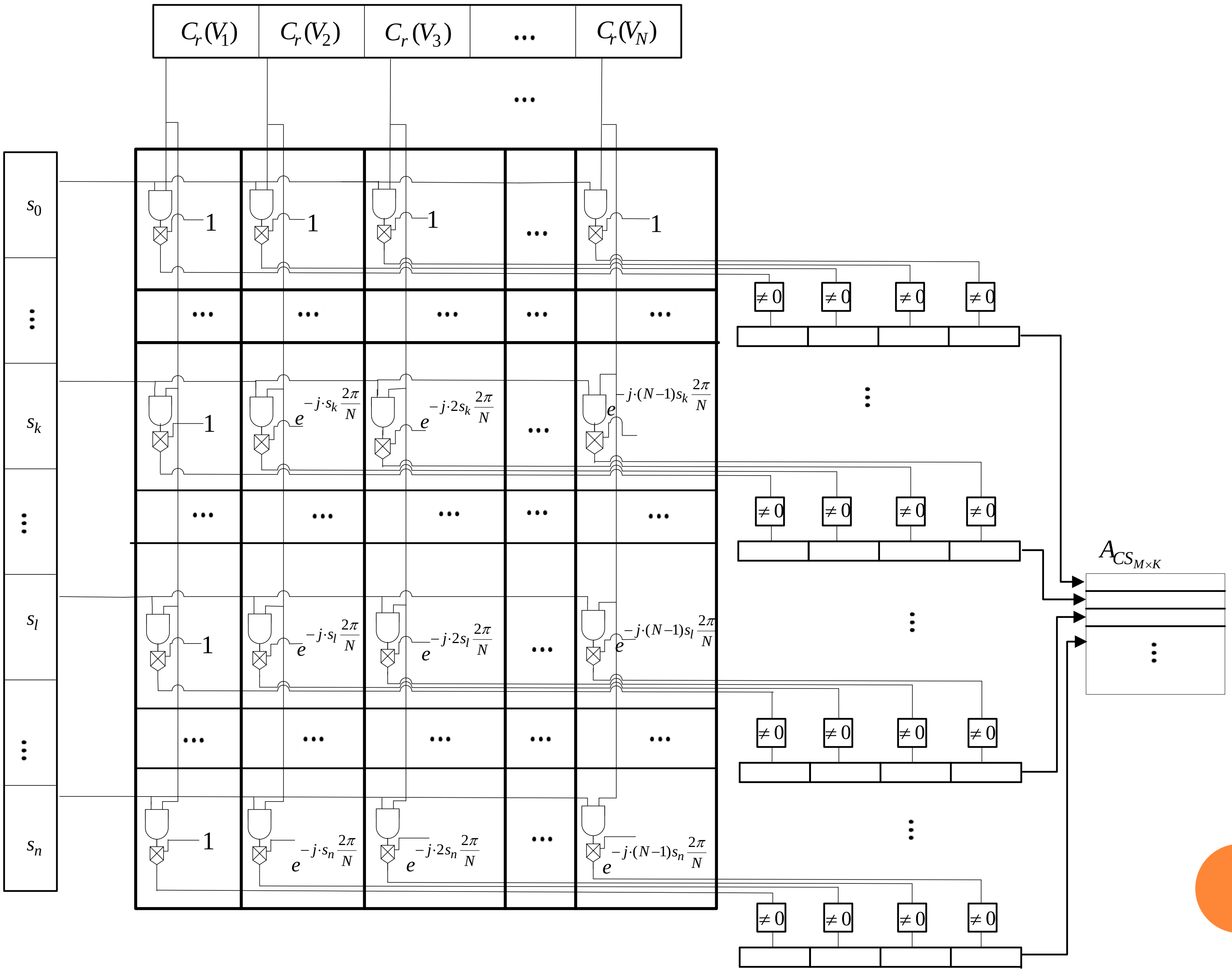


$v_1, \ldots, v_M$ ⇨ Available signal samples

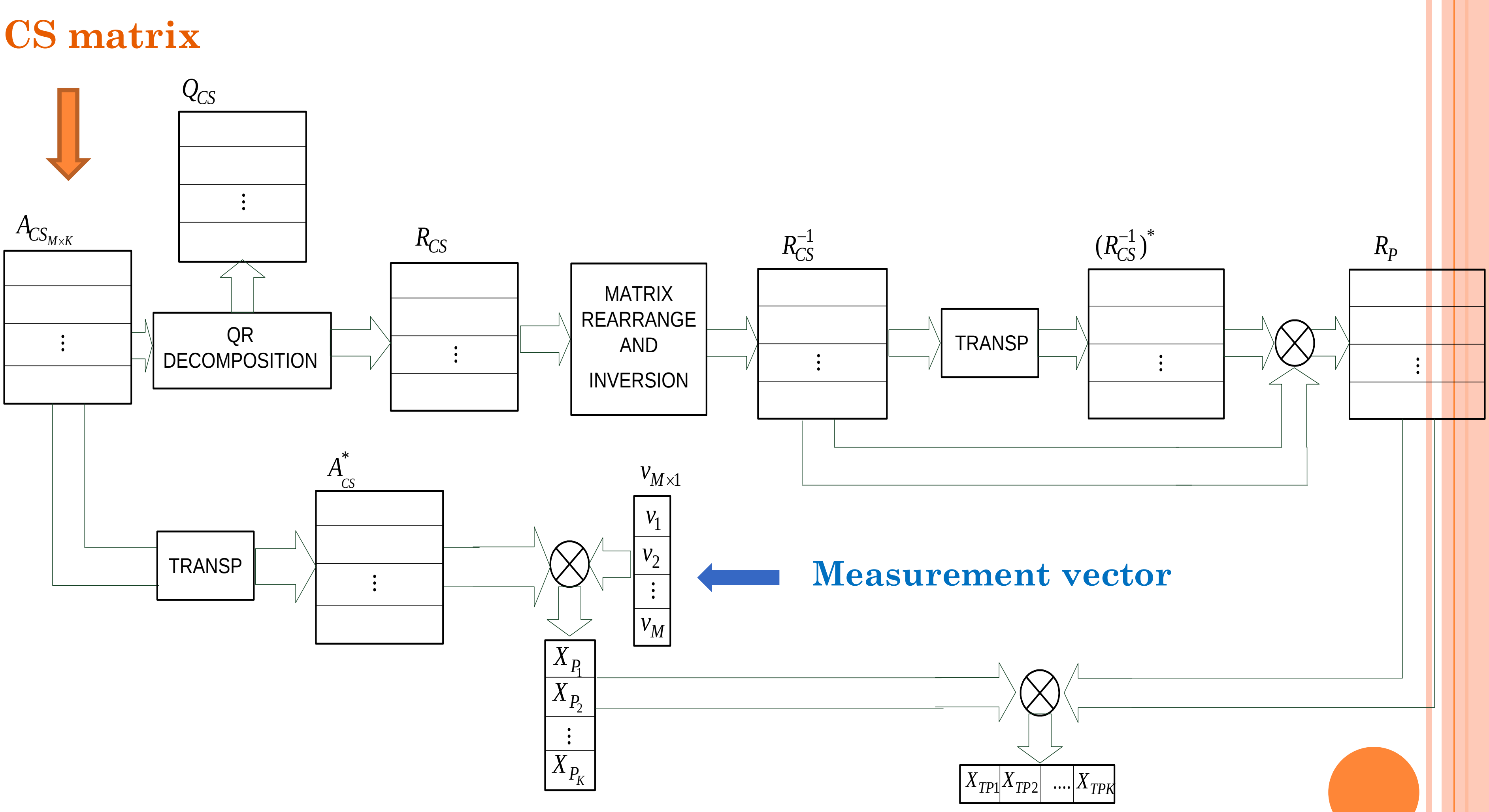
N-signal length

M-number of available samples

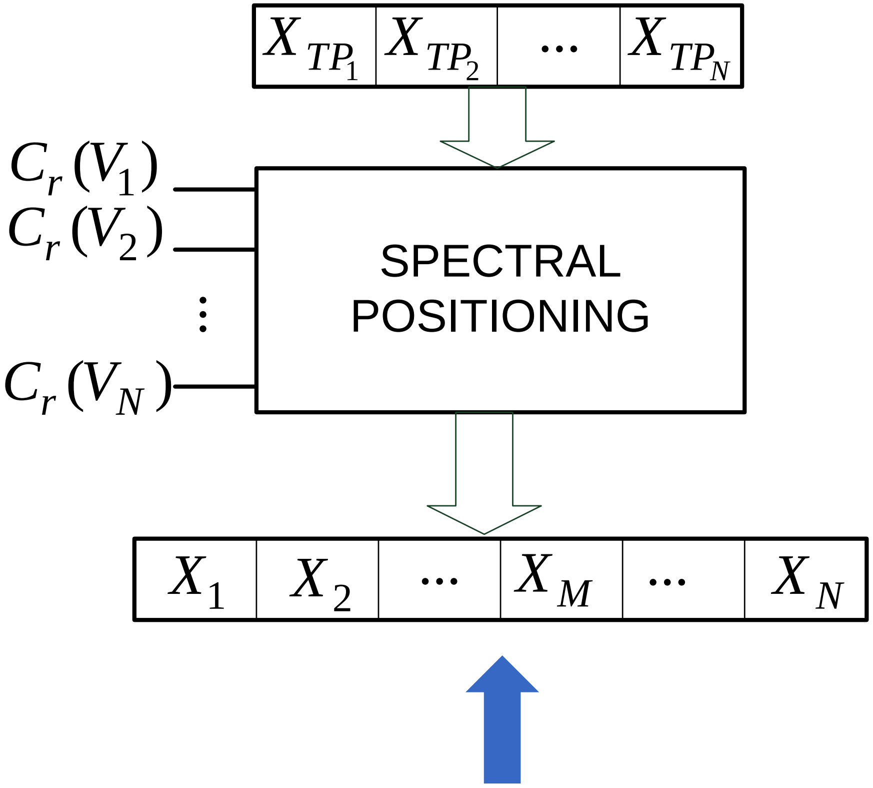
Block scheme – Part 2



Block scheme – Part 3

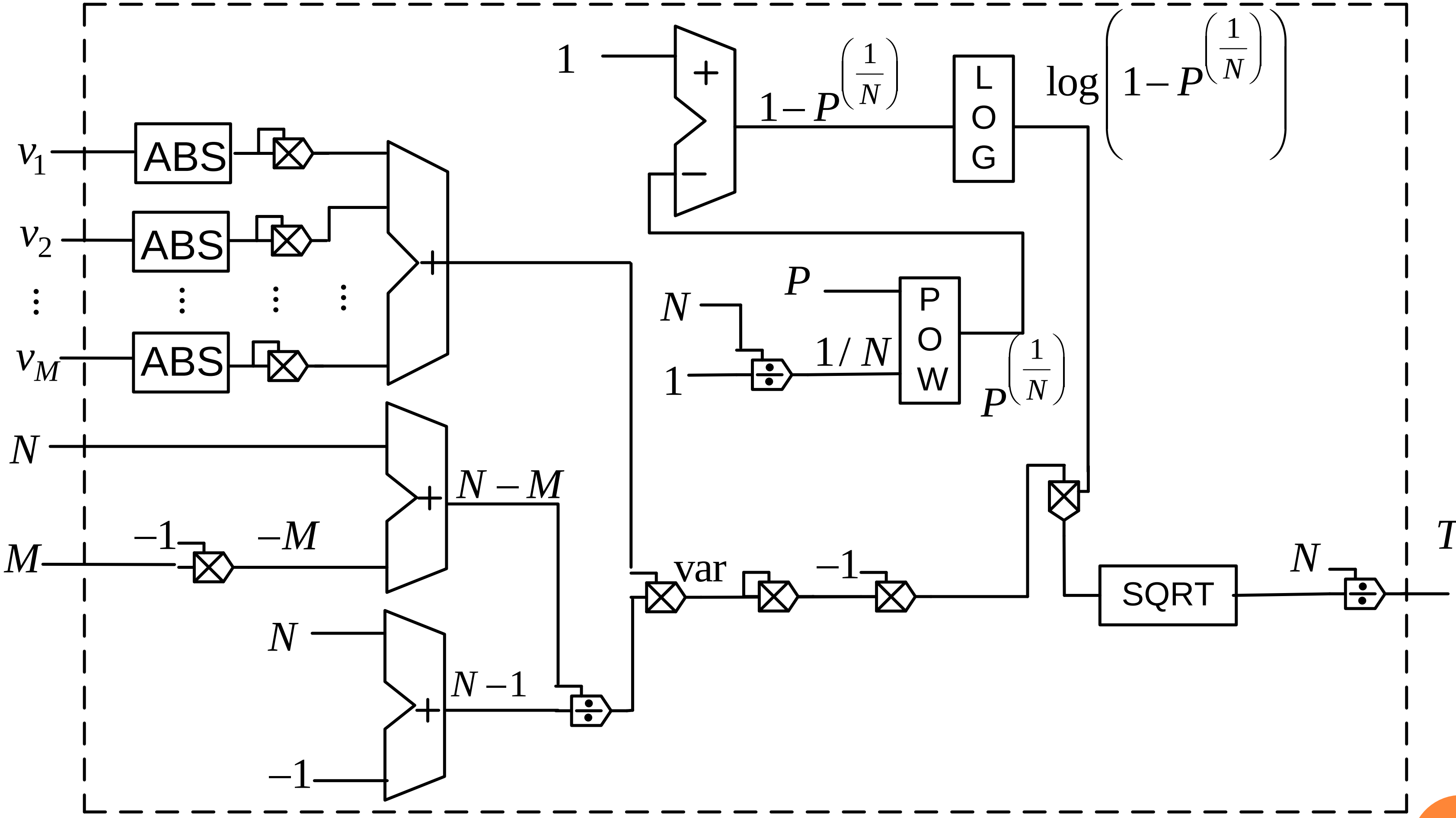


Block for spectral positioning



Reconstructed DFT

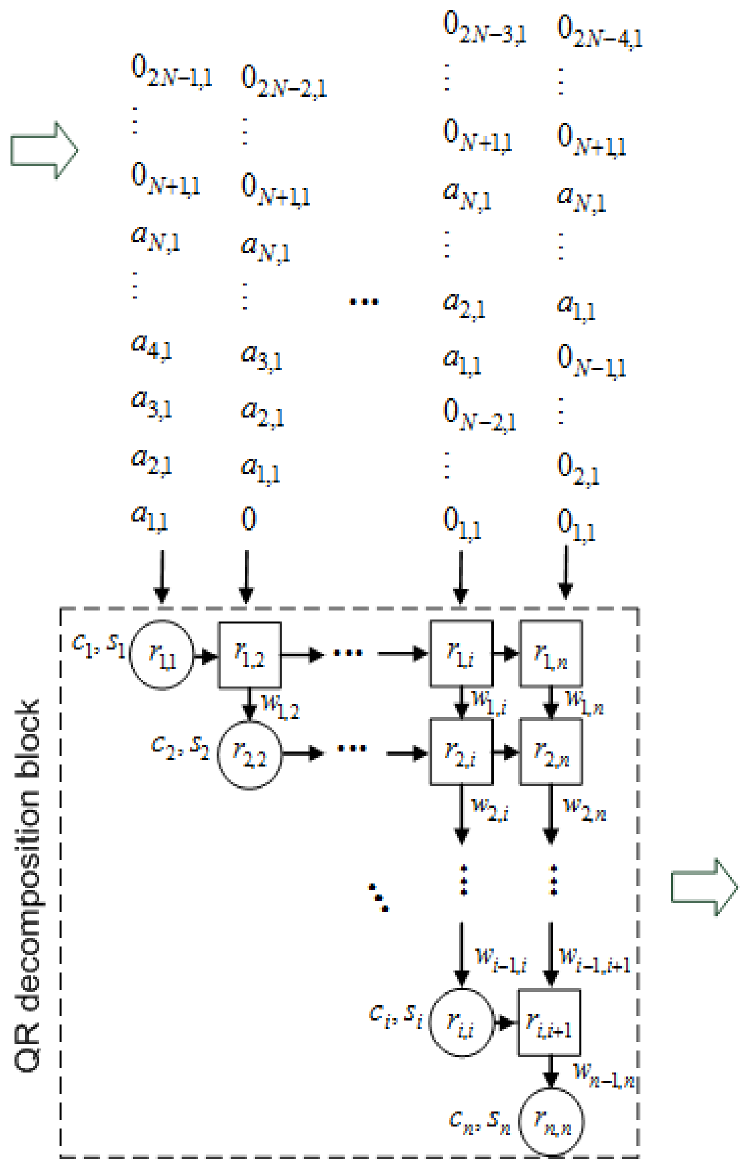
Block for THRESHOLD calculation



QR Decomposition

$$A = QR,$$
$$Q^T Q = I.$$
$$\Downarrow$$
$$A^{-1} = R^{-1} Q^T$$

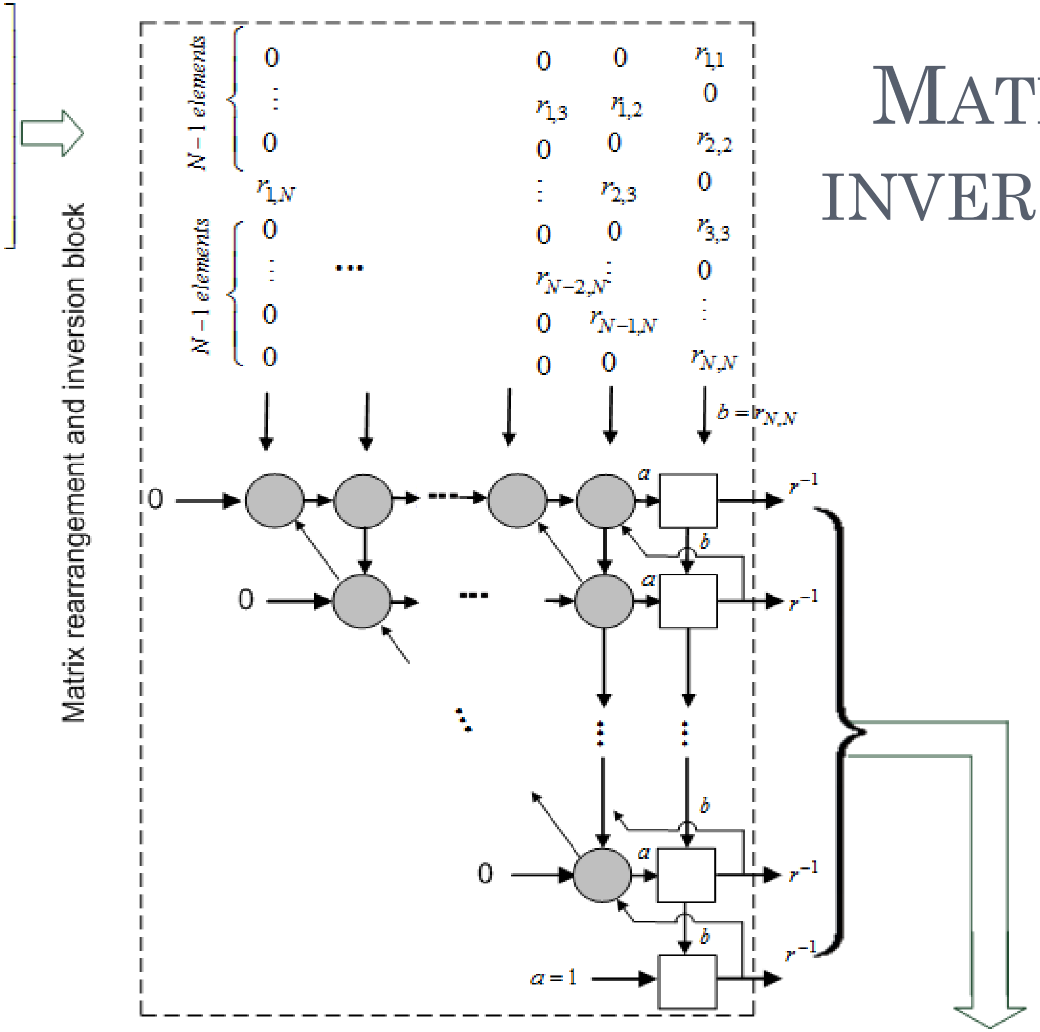
$$A_{CS} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,N} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{N,1} & a_{N,2} & a_{N,3} & \cdots & a_{N,N} \end{bmatrix}$$



$$R_{CS} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \cdots & r_{1,N-1} & r_{1,N} \\ 0 & r_{2,2} & r_{2,3} & \cdots & r_{2,N-1} & r_{2,N} \\ 0 & 0 & r_{3,3} & \cdots & r_{3,N-1} & r_{3,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & r_{N-1,N-1} & r_{N-1,N} \\ 0 & 0 & 0 & \cdots & 0 & r_{N,N} \end{bmatrix}$$

MATRIX INVERSION

$$R_{CS}=\begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \cdots & r_{1,N-1} & r_{1,N} \\ 0 & r_{2,2} & r_{2,3} & \cdots & r_{2,N-1} & r_{2,N} \\ 0 & 0 & r_{3,3} & \cdots & r_{3,N-1} & r_{3,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & r_{N-1,N-1} & r_{N-1,N} \\ 0 & 0 & 0 & \cdots & 0 & r_{N,N} \end{bmatrix}$$



$$A = QR,$$

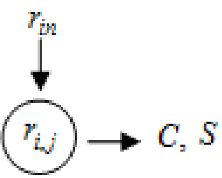
$$Q^T Q = I.$$

$$A^{-1} = R^{-1} Q^T$$

$$R_{CS}^{-1}=\begin{bmatrix} r_{1,1}^{-1} & r_{1,2}^{-1} & r_{1,3}^{-1} & \cdots & r_{1,N-1}^{-1} & r_{1,N}^{-1} \\ 0 & r_{2,2}^{-1} & r_{2,3}^{-1} & \cdots & r_{2,N-1}^{-1} & r_{2,N}^{-1} \\ 0 & 0 & r_{3,3}^{-1} & \cdots & r_{3,N-1}^{-1} & r_{3,N}^{-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & r_{N-1,N-1}^{-1} & r_{N-1,N}^{-1} \\ 0 & 0 & 0 & \cdots & 0 & r_{N,N}^{-1} \end{bmatrix}$$

$r_{N,N}^{-1}$	0	$r_{N-1,N-1}^{-1}$	···	$r_{3,3}^{-1}$	0	$r_{2,2}^{-1}$	0	$r_{1,1}^{-1}$
0	0	$r_{N-1,N}^{-1}$	···	$r_{3,N}^{-1}$	0	$r_{2,3}^{-1}$	0	$r_{1,2}^{-1}$
0	0	0	···	0	0	$r_{2,N}^{-1}$	0	$r_{1,3}^{-1}$
⋮		⋮	···				⋮	⋮
0	0	0		0	0	0	0	$r_{1,N}^{-1}$

Cell architectures for QR decomposition and matrix inversion:

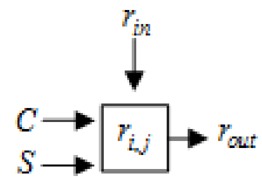


$$r_{in} = 0 \rightarrow \quad C \leftarrow 1, \quad S \leftarrow 0$$

$$r_{in} \neq 0 \rightarrow \quad C \leftarrow \frac{r_{i,j}}{\sqrt{r_{i,j}^2 + r_{in}^2}}$$

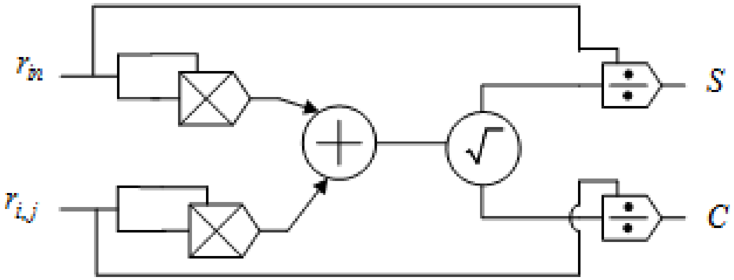
$$S \leftarrow \frac{r_{in}}{\sqrt{r_{i,j}^2 + r_{in}^2}}$$

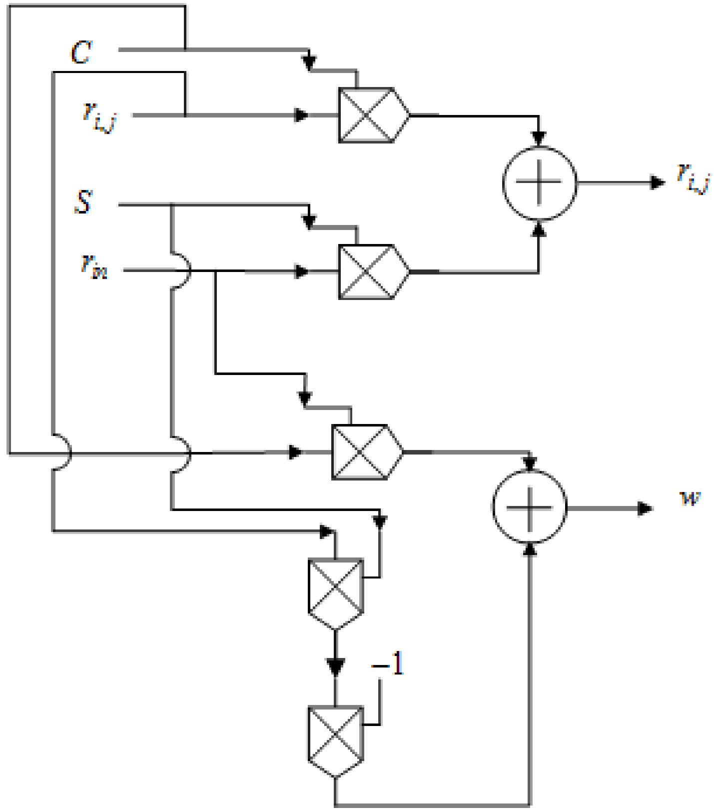
$$r_{i,j} \leftarrow \sqrt{r_{i,j}^2 + r_{in}^2}$$



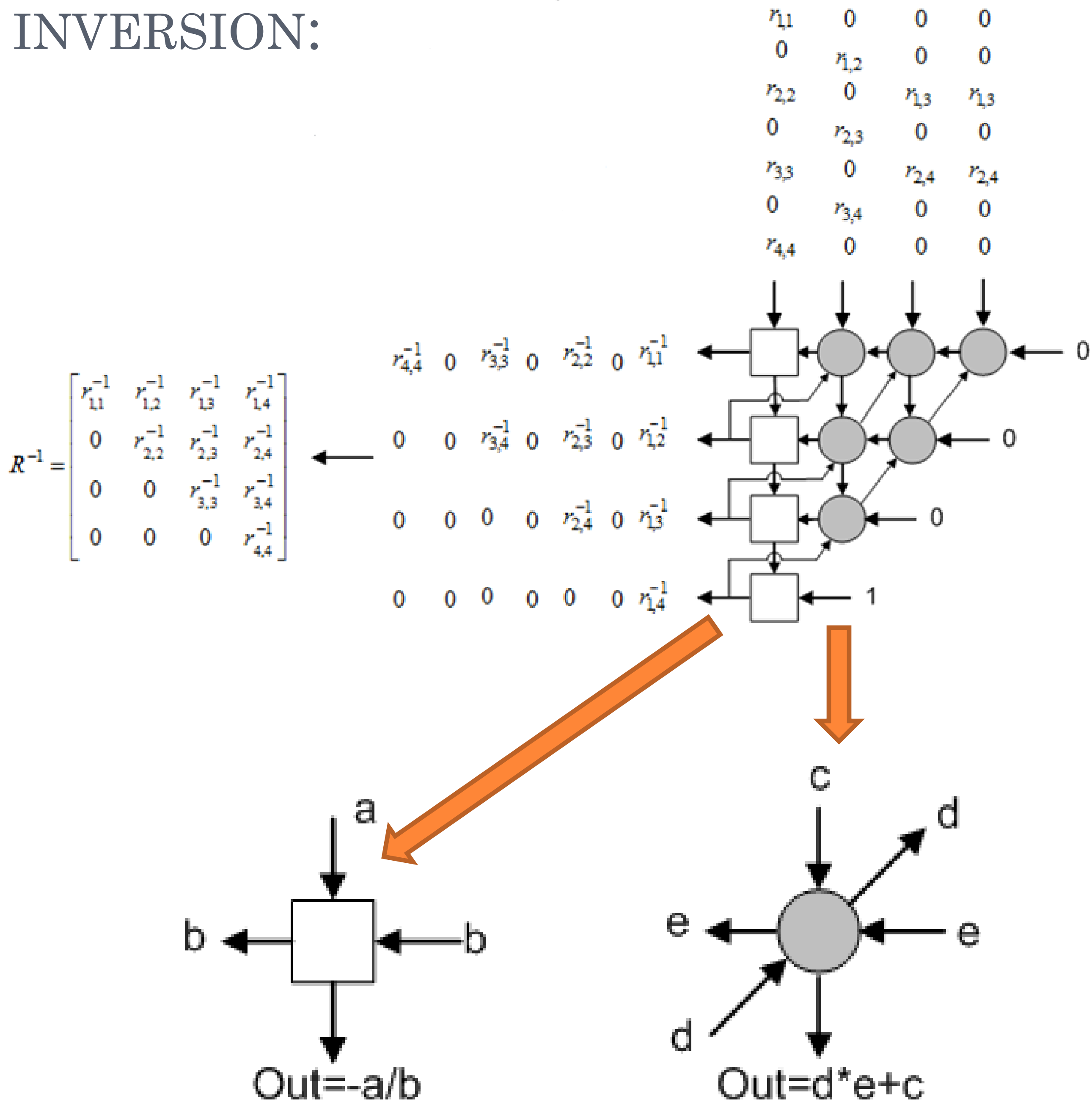
$$w \leftarrow -S \cdot r_{i,j} + C \cdot r_{in}$$

$$r_{i,j} \leftarrow C \cdot r_{i,j} + S \cdot r_{in}$$





Cell architectures for Triangular matrix inversion:



Thank you