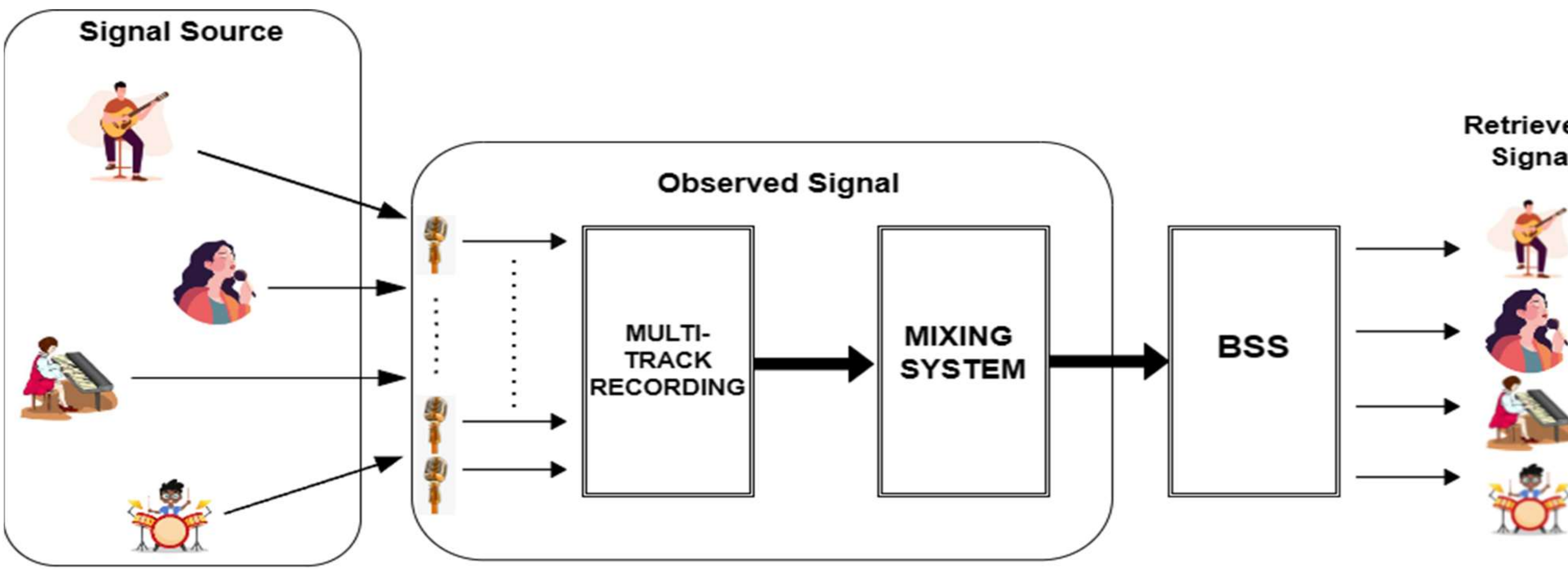
A Fixed-Point Pre-Processing Hardware Architecture Design for Complex Independent Component Analysis

Yashwant Moses & Madhav Rao

IIIT-Bangalore, India

BLIND SOURCE SEPERATION

- Blind Source Separation (BSS) is a signal processing technique used to segregate original source component signals from the given mixed signal without any prior knowledge of the sources or the mixing process.

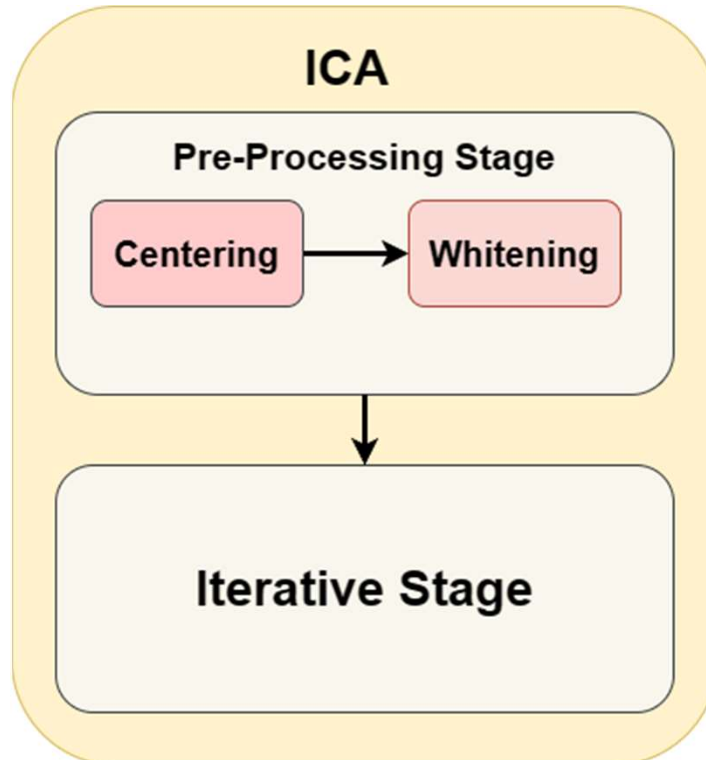


INDEPENDENT COMPONENT ANALYSIS (ICA)

- ICA runs by finding a transformation matrix that de-correlates the observed mixtures and maximizes the statistical independence of the resulting signals, thereby extracting the original components.

- The preprocessing step ensures that the input data has zero mean and no linear correlation.
  - Centering aids in turning the mixed data to zero mean.
  - The whitening process removes any correlations.

- The Iterative stage aims to find the de-mixing matrix.



BASIC CONCEPT

- Consider a source signal matrix s(t), which consist of 'n' source signals

$$s(t) = \begin{pmatrix} S_1 \\ \vdots \\ S_n \end{pmatrix} = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1k} \\ \vdots & & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nk} \end{pmatrix}$$

where, S_i, i = 1,2…n is the individual source signal with k time steps.

- Let x(t) ε $\mathrm{C}^{\mathrm{m \times k}}$, be the observed signal matrix which is obtained at the output of the mixing system which consist of 'm' recording elements. Then,

$$x(t) = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} . \begin{pmatrix} S_1 \\ \vdots \\ S_n \end{pmatrix}$$

x(t) = A . s(t) , where

A ε $\mathrm{R}^{\mathrm{m \times n}}$, is the mixing matrix.

- A is a regular matrix. Therefore, we can rewrite the model as

  s(t) = B.x(t) ,   where

  B = A^{-1}. It is only necessary to estimate B so that $\{S_{\mathrm{i}}\}$ are independent.

- Therefore the goal of ICA is to retrieve the original signal s(t) from the observable signal x(t), given the mixing matrix A is unknown.

PRE-PROCESSING

- **Centering**:
  - $\hat{x}$ ε $C^{m \times k}$ is centered matrix with zero mean and E[x_i] is the mean of the input mixed signal of the i^{th} channel, where i ε 1,2…m.

$$\hat{x}_i = x_i - E[x_i]$$

- **Whitening**:
  - It involves the estimation of the covariance matrix $\hat{x}_c$ ε $C^{m \times m}$ which is Hermitian in nature.

$$\hat{x}_c = E[\hat{x}\,\hat{x}^H]$$

  - The EVD is performed on the $\hat{x}_c$, where, D is the eigenvalue of the $\hat{x}_c$ in the form of the diagonal matrix, and E is the eigenvector matrix.

$$\hat{x}_c = EDE^H$$

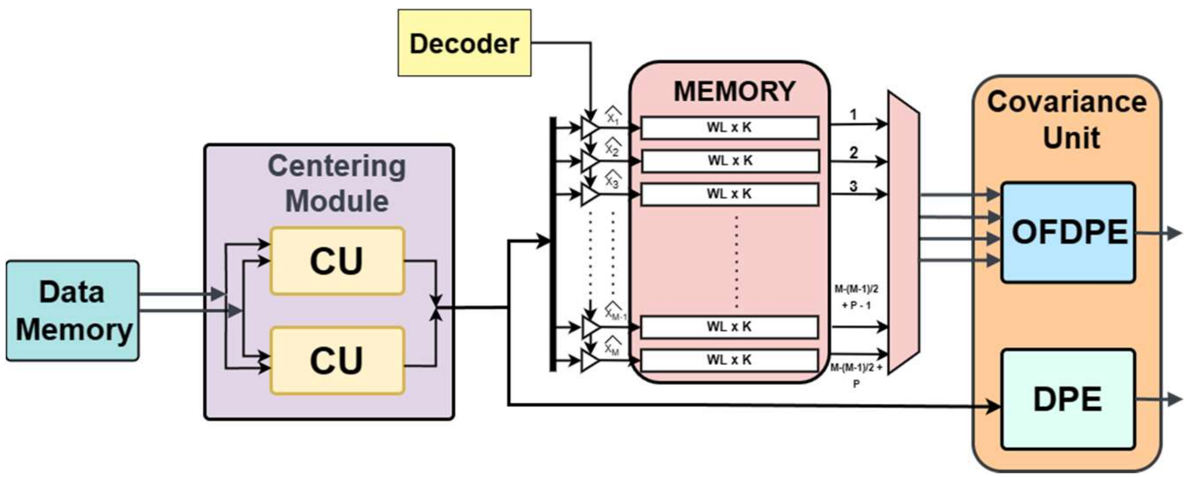
  - Eventually, the whitened matrix Y ε $C^{m \times k}$ is computed with unit variance.

$$Y = D^{1/2}\, E\, \hat{x}$$

PROPOSED ARCHITECTURE

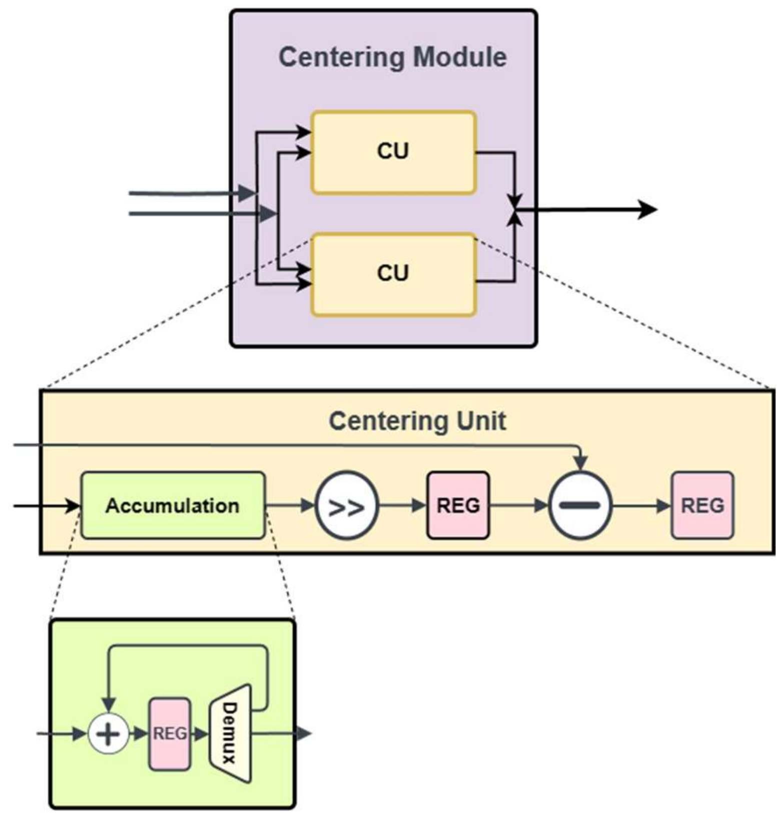
- The pre-processing is executed in three stages using the proposed architecture:
  - Centred matrix and Covariance matrix calculation.
  - Eigenvalue Decomposition of the Covariance matrix.
  - Whiten matrix calculation.

- Centred matrix and Covariance matrix calculation:
  - This is carried on by the Centering and Covariance Unit.



CENTERING MODULE

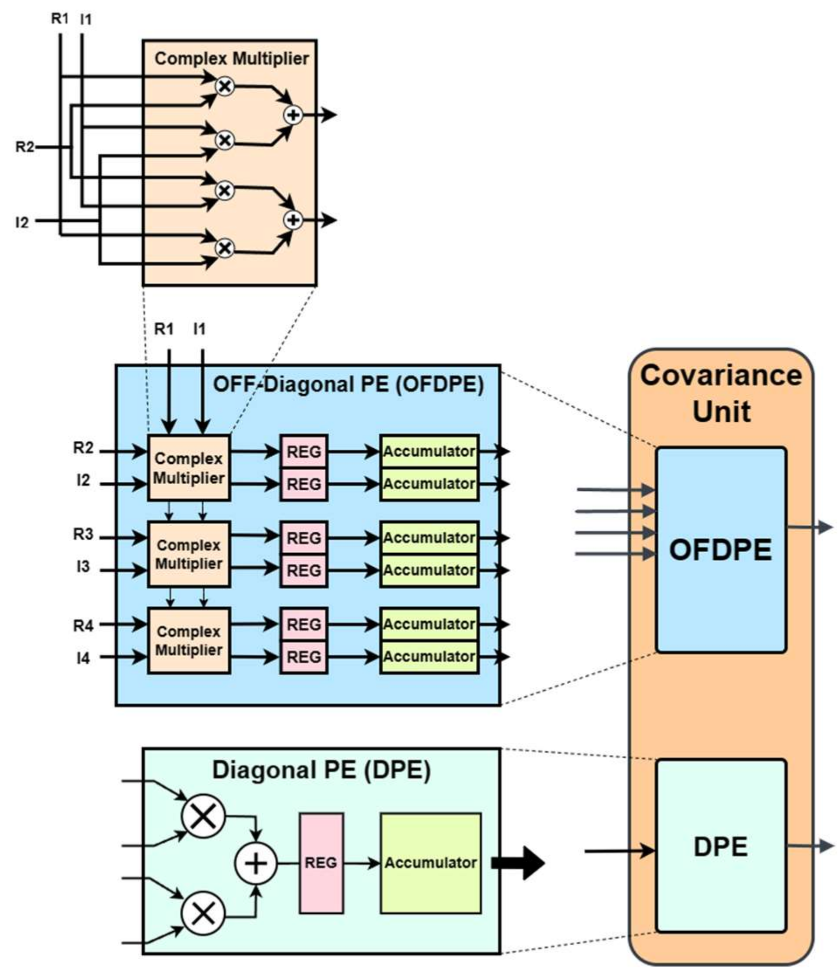
- The input data is fed channel wise (x_i) from the data memory to the centering unit, where the centering operations for real and imaginary components are performed in parallel.
- The accumulation and shifting operation is executed in parallel along with subtraction within the centering components maximizing the throughput. (Pipelined)
- The mean of K elements ($E[x_i]$) is calculated and subtracted from the same set of K elements to give $\hat{x}_i$
- This process is repeated M times, thus giving out the centered matrix $\hat{x}$



COVARIANCE UNIT

- The Covariance Unit consist of Off-Diagonal PE (OFDPE) and Diagonal PE (DPE).

- Diagonal Processing Element (DPE)
  - As the covariance matrix is Hermitian in nature, the diagonal elements are real numbers, no complex multiplier required.
  - It takes K cycles by the DPE to calculate a single diagonal element, hence all diagonal element are generated within M × K cycles.



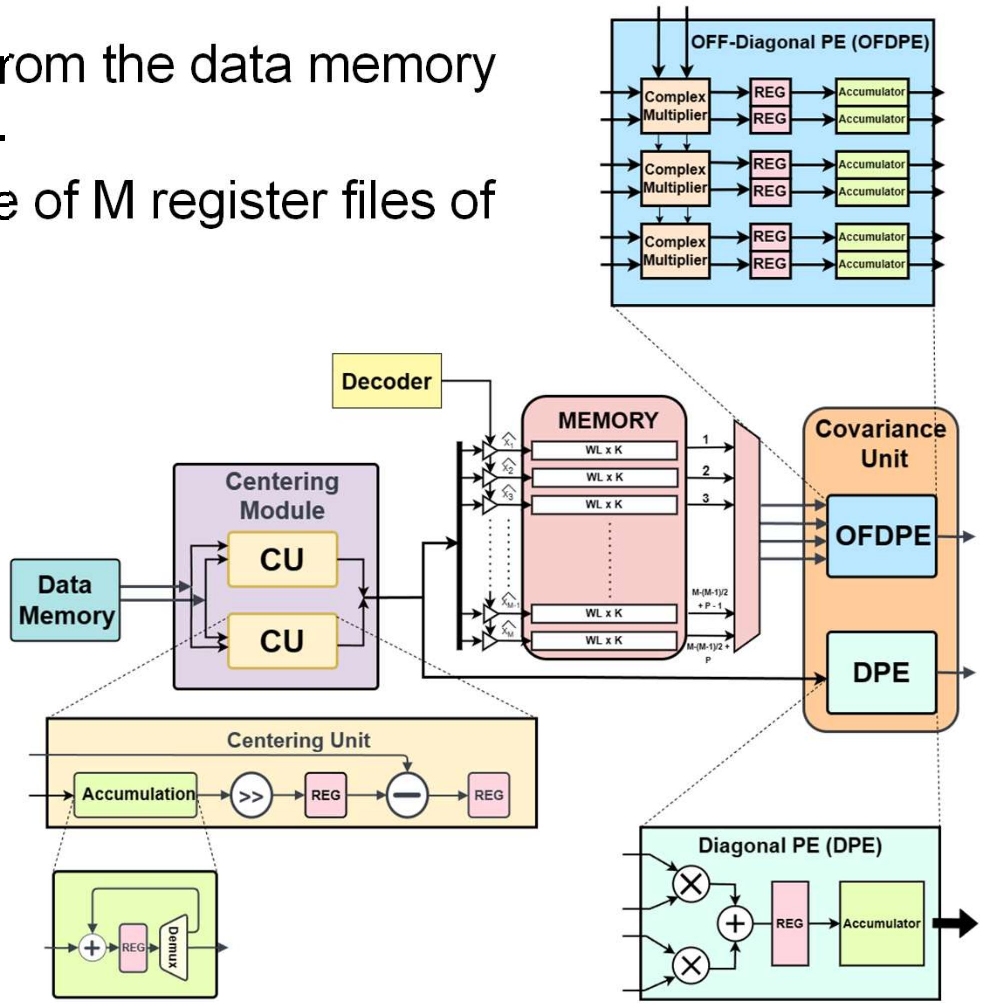
- Off-Diagonal Processing Element (OFDPE)
  - The OFDPE consist of complex multipliers.
  - The centering operation for at least four channels have to executed and stored in the memory in order to prevent data hazard.
  - The OFDPE is capable of taking four channel vectors as input simultaneously to generate three of the covariance matrix elements in M cycles.
  - Therefore, for computing the off-diagonal elements from M × K centered matrix,, consumes (M × (M - 1)/6 + P) × K cycles , where 0 ≤ P ≤ 2 depending on the dimension of the covariance matrix.

$$\hat{\mathbf{X}}_{\mathbf{c}} = \begin{bmatrix} \hat{X}_{c11} & \hat{X}_{c12} & \hat{X}_{c13} & \hat{X}_{c14} & \hat{X}_{c15} & \hat{X}_{c16} & \hat{X}_{c17} & \hat{X}_{c18} \\ \hat{X}_{c21} & \hat{X}_{c22} & \hat{X}_{c23} & \hat{X}_{c24} & \hat{X}_{c25} & \hat{X}_{c26} & \hat{X}_{c27} & \hat{X}_{c28} \\ \hat{X}_{c31} & \hat{X}_{c32} & \hat{X}_{c33} & \hat{X}_{c34} & \hat{X}_{c35} & \hat{X}_{c36} & \hat{X}_{c37} & \hat{X}_{c38} \\ \hat{X}_{c41} & \hat{X}_{c42} & \hat{X}_{c43} & \hat{X}_{c44} & \hat{X}_{c45} & \hat{X}_{c46} & \hat{X}_{c47} & \hat{X}_{c48} \\ \hat{X}_{c51} & \hat{X}_{c52} & \hat{X}_{c53} & \hat{X}_{c54} & \hat{X}_{c55} & \hat{X}_{c56} & \hat{X}_{c57} & \hat{X}_{c58} \\ \hat{X}_{c61} & \hat{X}_{c62} & \hat{X}_{c63} & \hat{X}_{c64} & \hat{X}_{c65} & \hat{X}_{c66} & \hat{X}_{c67} & \hat{X}_{c68} \\ \hat{X}_{c71} & \hat{X}_{c72} & \hat{X}_{c73} & \hat{X}_{c74} & \hat{X}_{c75} & \hat{X}_{c76} & \hat{X}_{c77} & \hat{X}_{c78} \\ \hat{X}_{c81} & \hat{X}_{c82} & \hat{X}_{c83} & \hat{X}_{c84} & \hat{X}_{c85} & \hat{X}_{c86} & \hat{X}_{c87} & \hat{X}_{c88} \end{bmatrix}$$

$$= \begin{bmatrix} \hat{X}_1\hat{X}_1^H & \hat{X}_1\hat{X}_2^H & \hat{X}_1\hat{X}_3^H & \hat{X}_1\hat{X}_4^H & \hat{X}_1\hat{X}_5^H & \hat{X}_1\hat{X}_6^H & \hat{X}_1\hat{X}_7^H & \hat{X}_1\hat{X}_8^H \\ \hat{X}_2\hat{X}_1^H & \hat{X}_2\hat{X}_2^H & \hat{X}_2\hat{X}_3^H & \hat{X}_2\hat{X}_4^H & \hat{X}_2\hat{X}_5^H & \hat{X}_2\hat{X}_6^H & \hat{X}_2\hat{X}_7^H & \hat{X}_2\hat{X}_8^H \\ \hat{X}_3\hat{X}_1^H & \hat{X}_3\hat{X}_2^H & \hat{X}_3\hat{X}_3^H & \hat{X}_3\hat{X}_4^H & \hat{X}_3\hat{X}_5^H & \hat{X}_3\hat{X}_6^H & \hat{X}_3\hat{X}_7^H & \hat{X}_3\hat{X}_8^H \\ \hat{X}_4\hat{X}_1^H & \hat{X}_4\hat{X}_2^H & \hat{X}_4\hat{X}_3^H & \hat{X}_4\hat{X}_4^H & \hat{X}_4\hat{X}_5^H & \hat{X}_4\hat{X}_6^H & \hat{X}_4\hat{X}_7^H & \hat{X}_4\hat{X}_8^H \\ \hat{X}_5\hat{X}_1^H & \hat{X}_5\hat{X}_2^H & \hat{X}_5\hat{X}_3^H & \hat{X}_5\hat{X}_4^H & \hat{X}_5\hat{X}_5^H & \hat{X}_5\hat{X}_6^H & \hat{X}_5\hat{X}_7^H & \hat{X}_5\hat{X}_8^H \\ \hat{X}_6\hat{X}_1^H & \hat{X}_6\hat{X}_2^H & \hat{X}_6\hat{X}_3^H & \hat{X}_6\hat{X}_4^H & \hat{X}_6\hat{X}_5^H & \hat{X}_6\hat{X}_6^H & \hat{X}_6\hat{X}_7^H & \hat{X}_6\hat{X}_8^H \\ \hat{X}_7\hat{X}_1^H & \hat{X}_7\hat{X}_2^H & \hat{X}_7\hat{X}_3^H & \hat{X}_7\hat{X}_4^H & \hat{X}_7\hat{X}_5^H & \hat{X}_7\hat{X}_6^H & \hat{X}_7\hat{X}_7^H & \hat{X}_7\hat{X}_8^H \\ \hat{X}_8\hat{X}_1^H & \hat{X}_8\hat{X}_2^H & \hat{X}_8\hat{X}_3^H & \hat{X}_8\hat{X}_4^H & \hat{X}_8\hat{X}_5^H & \hat{X}_8\hat{X}_6^H & \hat{X}_8\hat{X}_7^H & \hat{X}_8\hat{X}_8^H \end{bmatrix}$$

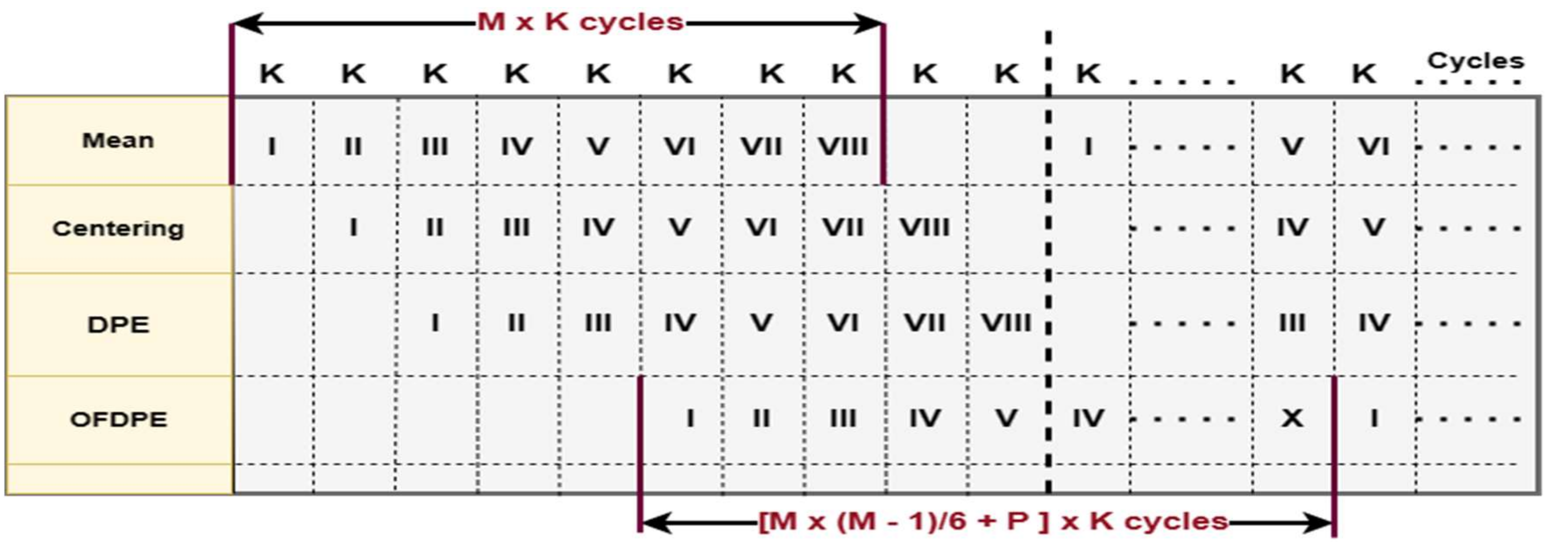
SUMMARY OF CENTERING AND COVARIANCE MATRIX

- The input data is fed to Centering Module from the data memory where the centered matrix ($\hat{x}$) is calculated.
- $\hat{x}$ is stored in the memory is which comprise of M register files of size WL x K where WL is the wordlength.
- Simultaneously the elemets of $\hat{x}$ are sent to the DPE row-wise ($\hat{x}_i$), to determine the diagonal elements of $\hat{x}_c$.
- Only after four channels of $\hat{x}$ are stored, the calculation of upper off-diagonal elements of the $\hat{x}_c$ are initiated by OFDPE.
- The data from register files are fed into OFDPE.



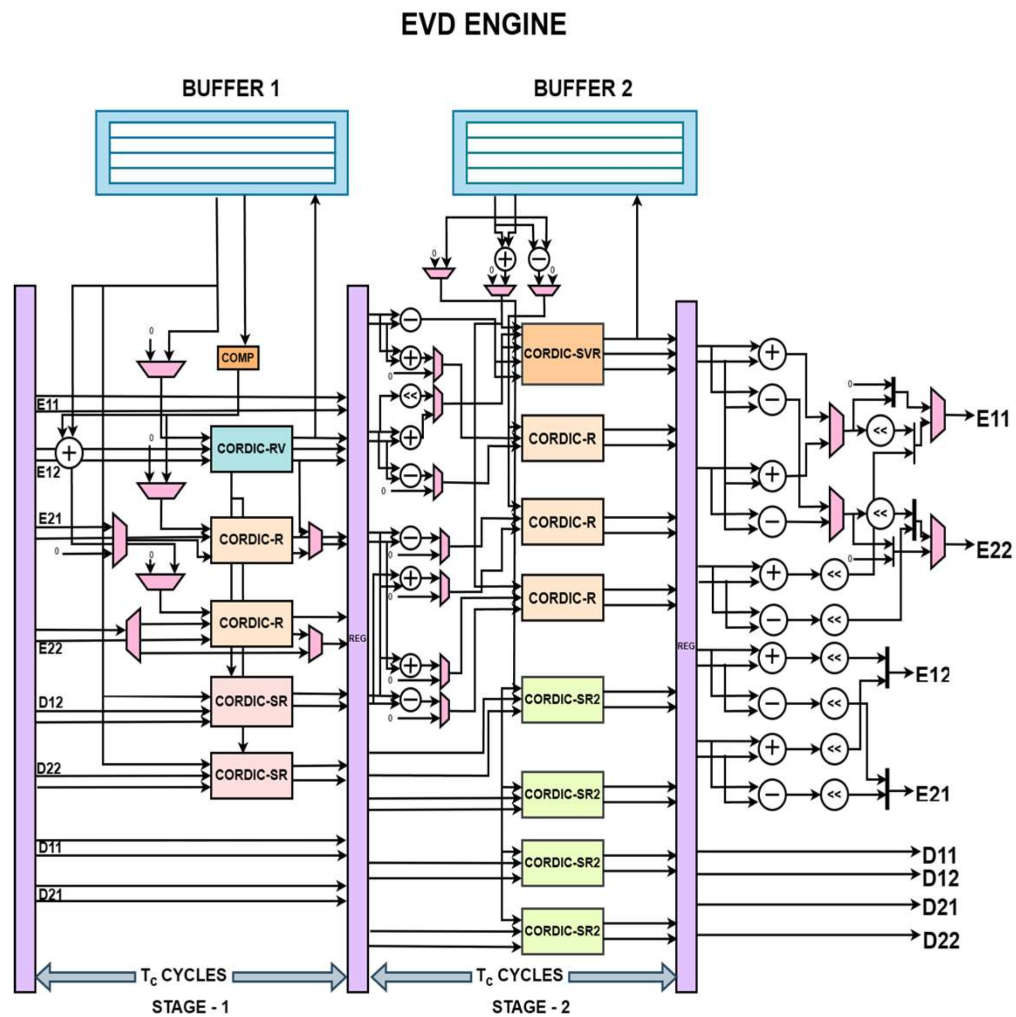
TIMING DIAGRAM

- For the centered matrix with M = 8 and K = 256, the proposed centering unit concedes around 2048 cycles and computation of the covariance matrix consumes 2560 cycles.
- An overall 3328 cycles of execution time is required for the entire process to complete for a single input mixed matrix X.



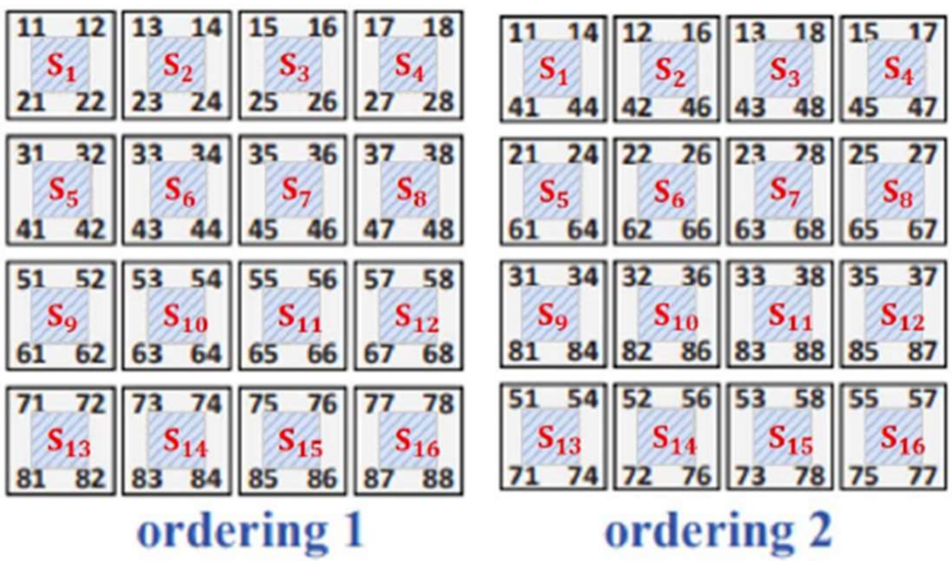
EIGENVALUE DECOMPOSITION

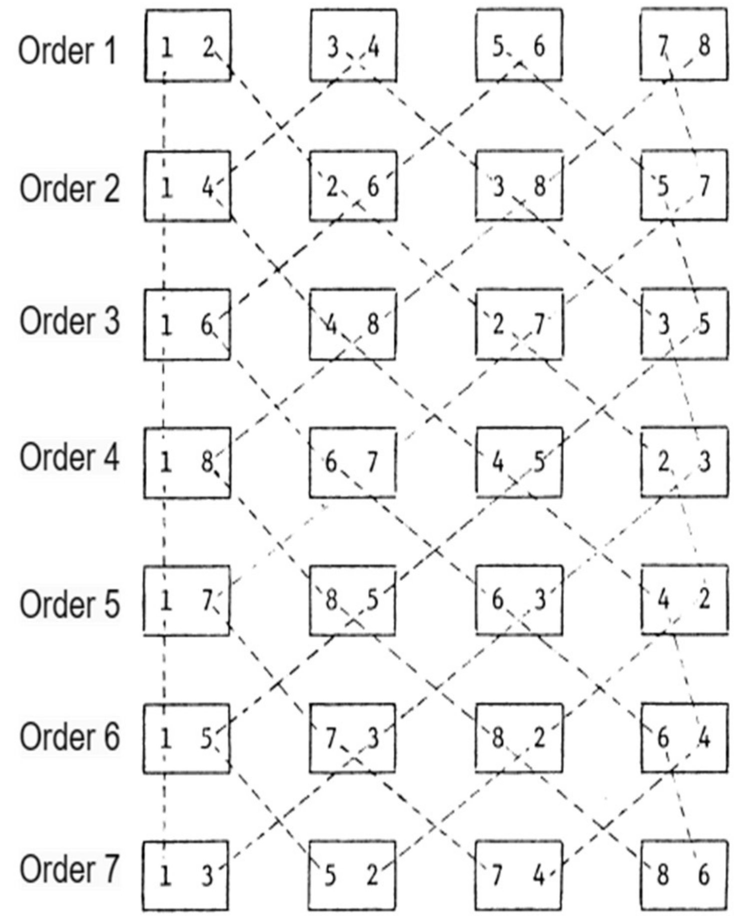
- Eigenvalue Decomposition of the Covariance matrix ($\hat{x}_c$) is performed by the EVD engine.
- The EVD engine works on the principle of parallel Jacobi method.
- It is a two staged pipeline design which is entirely multiplier-less and operates in two mode namely : diagonal and off-diagonal processing.
- It employs multiple specialized CORDIC units such as CORDIC-RV, CORDIC-SR, CORDIC-SVR CORDIC-SR2, and CORDIC-R.



PARALLEL JACOBI METHOD

- An M × M matrix is decomposed into 2 × 2 sub-matrices and process them based on a particular order.
- There are M - 1 order required for matrix of dimension M.



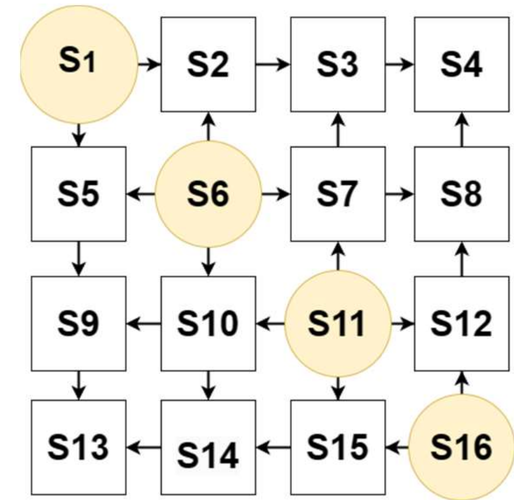


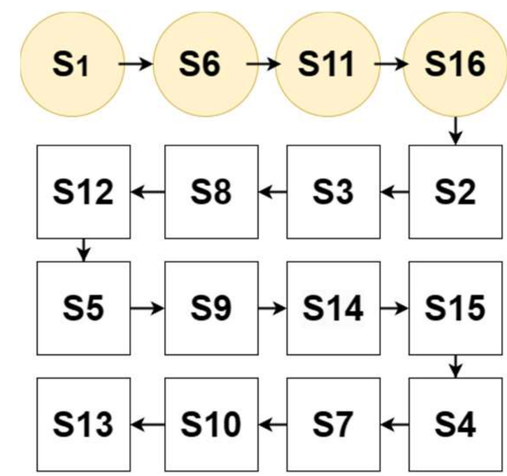
- Sub-matrices present at the diagonal undergo diagonalization iteratively by a series of rotations. (Diagonal processing)

- The off-diagonal sub-matrices are also rotated through the angles used by the corresponding diagonal sub-matrices in each order. (Off-Diagonal processing)

- Once the rotation required in all the order are completed, it is considered to be one sweep.

- Multiple sweeps are carried on till the off-diagonal elements are annihilated resulting in diagonal elements of M × M matrix remains that are the corresponding eigenvalues.





SUBMATRIX OPERATIONS

Step I - Calculate the rotational angles (α or β) from the diagonal submatrix.

Step II - Rotate the submatrix E by (α) and then (β) while D by only (β).

Step III - Calculate the rotational angle (θ) from the diagonal submatrix.

Step IV - Rotate the submatrix E by (θ_l) and then (θ_r) while D by only (θ_r) .

$$\mathbf{D^k\hat{U}_r} = \begin{bmatrix} D^k_{pp} & D_{pq} \\ D^k_{qp} & D_{qq} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{i\beta} \end{bmatrix} \begin{bmatrix} c_r & s_r \\ -s_r & c_r \end{bmatrix}$$

$$\mathbf{\hat{D}^kU_r} = \begin{bmatrix} D^k_{pp} & D_{pq}e^{i\beta} \\ D^k_{qp} & D_{qq}e^{i\beta} \end{bmatrix} \begin{bmatrix} c_r & s_r \\ -s_r & c_r \end{bmatrix}$$

$$\mathbf{D^{k+1}} = \begin{bmatrix} D^{k+1}_{pp} & D^{k+1}_{pq} \\ D^{k+1}_{qp} & D^{k+1}_{qq} \end{bmatrix}$$

$$\theta = tan^{-1}\left(\frac{2*E^k_{pq}}{(E^k_{qq} + E^k_{pp})}\right)$$

$$c_l = \cos(\theta_l) \quad s_l = \sin(\theta_l)$$
$$c_r = \cos(\theta_r) \quad s_r = \sin(\theta_r)$$

$$\alpha = \beta = tan^{-1}\left(\frac{imag(E^k_{pq})}{real(E^k_{pq})}\right)$$

$$\mathrm{\hat{U}_l E^k \hat{U}_r} = \begin{bmatrix} c_l & -s_l \\ s_l & c_l \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix} \begin{bmatrix} E^k_{pp} & E^k_{pq} \\ E^k_{qp} & E^k_{qq} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{i\beta} \end{bmatrix} \begin{bmatrix} c_r & s_r \\ -s_r & c_r \end{bmatrix}$$

$$\mathrm{U_l \hat{E}^k U_r} = \begin{bmatrix} c_l & -s_l \\ s_l & c_l \end{bmatrix} \begin{bmatrix} E^k_{pp} & E^k_{pq}e^{i\beta} \\ E^k_{qp}e^{i\alpha} & E^k_{qq}e^{i(\alpha+\beta)} \end{bmatrix} \begin{bmatrix} c_r & s_r \\ -s_r & c_r \end{bmatrix}$$

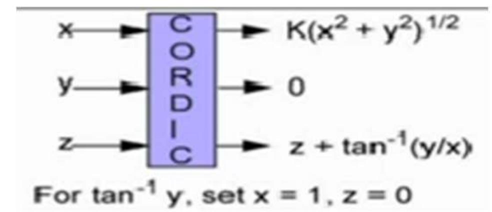
$$\mathrm{E^{k+1}} = \begin{bmatrix} E^{k+1}_{pp} & E^{k+1}_{pq} \\ E^{k+1}_{qp} & E^{k+1}_{qq} \end{bmatrix}$$

CORDIC UNITS

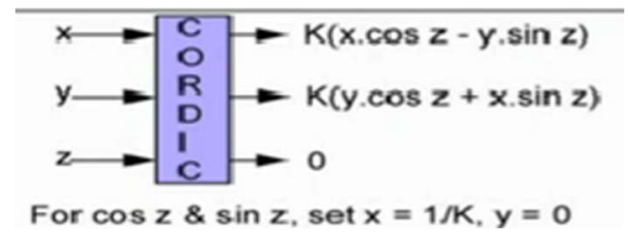
- **CORDIC-RV:**
  - It operates in vectoring mode for diagonal processing and rotational mode for off-diagonal processing.
  - In vectoring mode, it is used for computing the rotation angle (α or β) and transforming complex input to real.
  - In rotation mode, it is utilized for rotating the off-diagonal sub-matrix elements E_{12} through the rotation angle (β). $(\mathbf{E_{12}}.\, e^{i\boldsymbol{\beta}})$.

- **CORDIC-SR:**
  - It is used for stage-1 rotation of the eigenvector. $(\mathbf{D_{12}}.\, e^{i\boldsymbol{\beta}})$ & $(\mathbf{D_{22}}.\, e^{i\boldsymbol{\beta}})$
  - Operates only in rotation mode.
  - It intakes the sign set from the CORDIC-RV for diagonal submatrices, and for off-diagonal sub-matrices it intakes the angle from the Buffer-1

Vectoring Mode



Rotation Mode



- **CORDIC-SVR**
  - It consists of two sets of CORDIC engine.
  - It operates in both vectoring as well as rotation mode simultaneously.
  - For diagonal processing, one vectoring mode configured CORDIC engine generates sign-set which is used by the CORDIC-SR2 for computing eigenvector sub-matrices (D).
  - For off-diagonal processing, other CORDIC engine operating on rotational mode computes $\bar{y}_{SVR}$.

Diagonal Processing

$$\bar{y}_{SVR} = 2E^k_{pq} sin(2\theta) - (E^k_{pp} - E^k_{qp}) cos(2\theta)$$

Off-Diagonal Processing

$$\begin{aligned}\bar{y}_{SVR} =& (E^k_{p_l p_r} - E^k_{q_l p_r}) cos(\theta_l + \theta_r) + \\ & (E^k_{p_l q_r} + E^k_{q_l p_r}) sin(\theta_l + \theta_r)\end{aligned}$$

- **CORDIC-R**
  - It only operates in rotation mode in Stage-1, and is used only for off-diagonal elements such as $(\boldsymbol{E_{22}. e^{i(\alpha+\beta)}})$ and $(\boldsymbol{E_{21}. e^{i(\alpha)}})$.
  - All CORDIC-R engines in the second stage remain non-functional for diagonal processing.
  - For the non-diagonal sub-matrices, CORDIC-R produces $\bar{x}_{SVR}$,$\bar{x}_R$ and $\bar{y}_R$.

Off-Diagonal Processing

$$\begin{aligned}\bar{x}_{SVR} =& (E^k_{p_l q_r} + E^k_{q_l p_r}) cos(\theta_l + \theta_r) - \\ & (E^k_{p_l p_r} - E^k_{q_l q_r}) sin(\theta_l + \theta_r)\end{aligned}$$

$$\begin{aligned}\bar{x}_R =& (E^k_{p_l p_r} + E^k_{q_l q_r}) cos(\theta_l - \theta_r) - \\ & (E^k_{p_l q_r} - E^k_{q_l p_r}) sin(\theta_l - \theta_r)\end{aligned}$$

$$\begin{aligned}\bar{y}_R =& (E^k_{p_l q_r} - E^k_{q_l p_r}) cos(\theta_l - \theta_r) + \\ & (E^k_{p_l p_r} + E^k_{q_l q_r}) sin(\theta_l - \theta_r)\end{aligned}$$

- **CORDIC-SR2**
  - It is used for stage-2 rotation of the eigenvector.
  - It is similar to the CORDIC-SR unit.
  - For operating on off-diagonal sub-matrices these operators behave as traditional rotational CORDIC operators.
  - **NOTE** : The sign set of 2θ is equivalent to θ which is used for rotation operations for diagonal processing, with little modification in z and scaling factor K of CORDIC-SR2.

Scaling Factor Adjustment

$$z_{i+1} = \sum d_i \cdot 2^{-(i+1)}$$
$$K = \prod cos(2^{-(i+1)})$$

$$x_{i+1} = x_i - d_i \cdot y_i \cdot 2^{-i},$$
$$y_{i+1} = y_i + d_i \cdot x_i \cdot 2^{-i},$$
$$z_{i+1} = z_i - d_i \cdot \beta_i$$

- The final result for diagonal processing and off-diagonal processing are shown.

Diagonal Processing

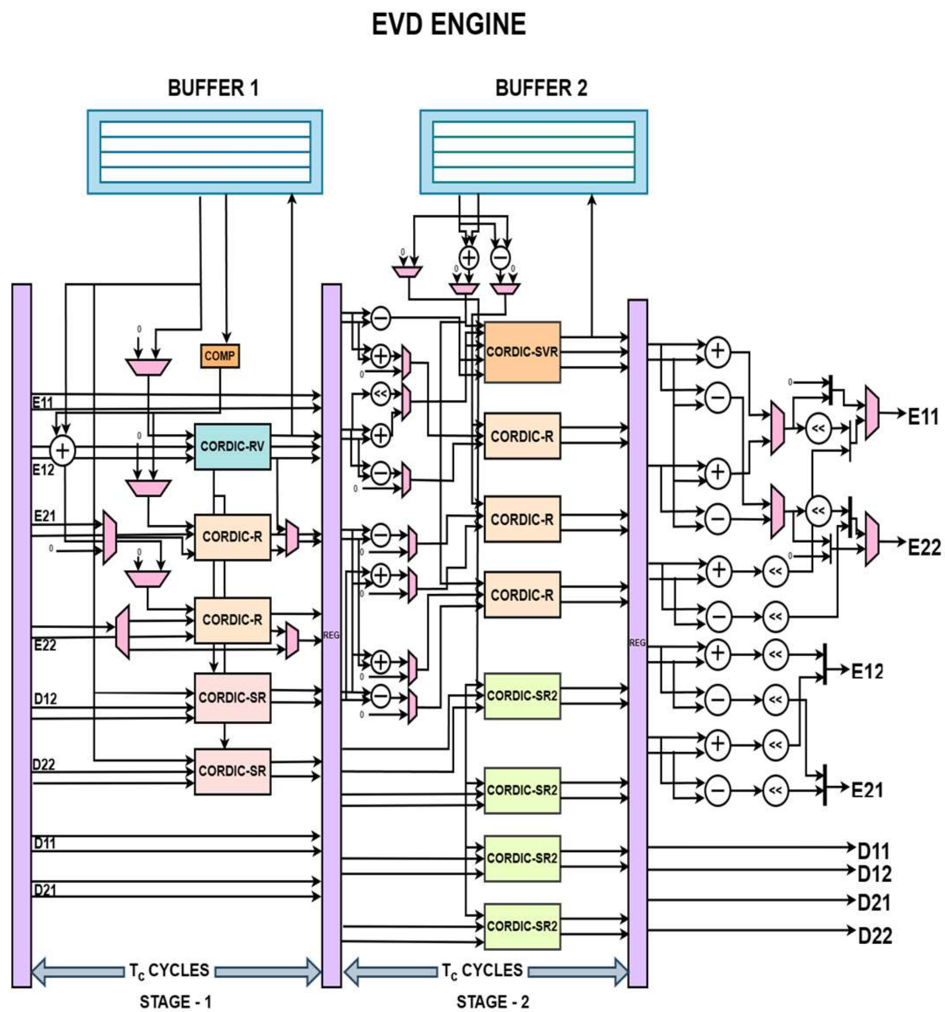
$$E_{pp}^{k+1} = (E_{qq}^k + E_{pp}^k + \bar{y}_{SVR})/2$$
$$E_{qq}^{k+1} = (E_{qq}^k + E_{pp}^k - \bar{y}_{SVR})/2$$

Off-Diagonal Processing

$$E_{pp}^{k+1} = (\bar{x_R} + \bar{y}_{SVR})/2$$
$$E_{pq}^{k+1} = (\bar{x}_{SVR} + \bar{y_R})/2$$
$$E_{qp}^{k+1} = (\bar{x}_{SVR} - \bar{y_R})/2$$
$$E_{qq}^{k+1} = (\bar{x_R} - \bar{y}_{SVR})/2$$

SUMMARY OF EIGENVALUE DECOMPOSITION

- The submatrices will be entered into the engine one by one in a pipelined fashion, first the diagonal and then the non-diagonal submatrices, in each ordering.
- This process repeats for ‘Ns’ sweeps to adequately generate the precise results.
- EVD engine requires [Tc × (M - 1) × Ns × (M × M / 4) + 20] cycles.
- For a covariance matrix of size M = 8, Tc = 10, it consumes a total of 4500 cycles.



WHITEN MATRIX CALCULATION

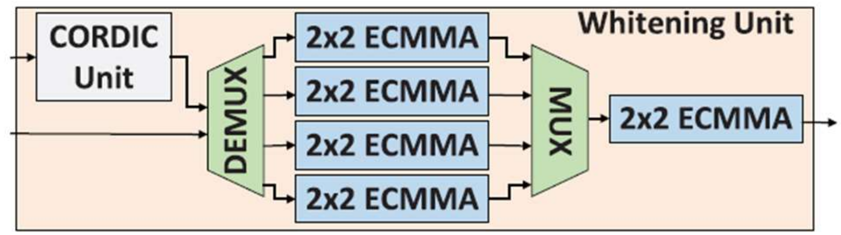
- The Whiten matrix is calculated using Whitening unit.
- The Whitening Unit used here is taken from the reference paper [19].

- The CORDIC in the whitening unit calculates the value of the diagonal item in the eigenvalue matrix in M × 32 cycles.

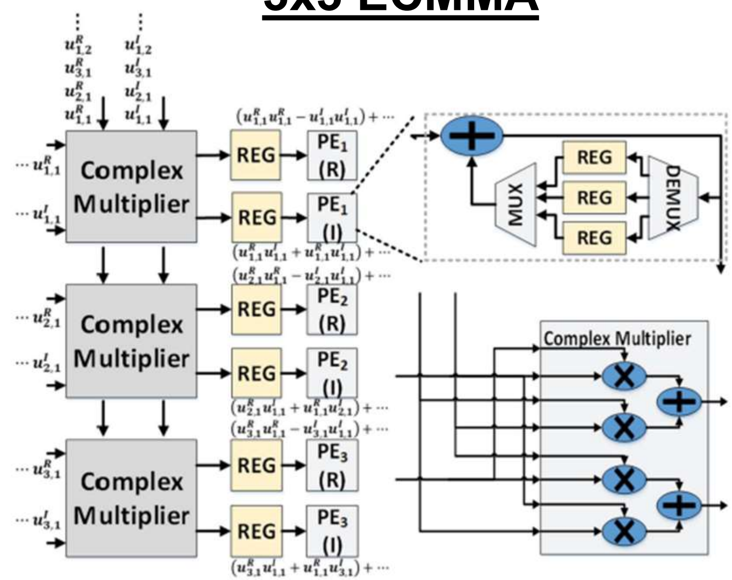
- The four 2 × 2 ECMMA compute the multiplications of the matrices by approximately (M/8) × M × K cycles.

- Hence the Whitening Unit consumes 2304 cycles for the M = 8 and k = 256.

Whitening Unit



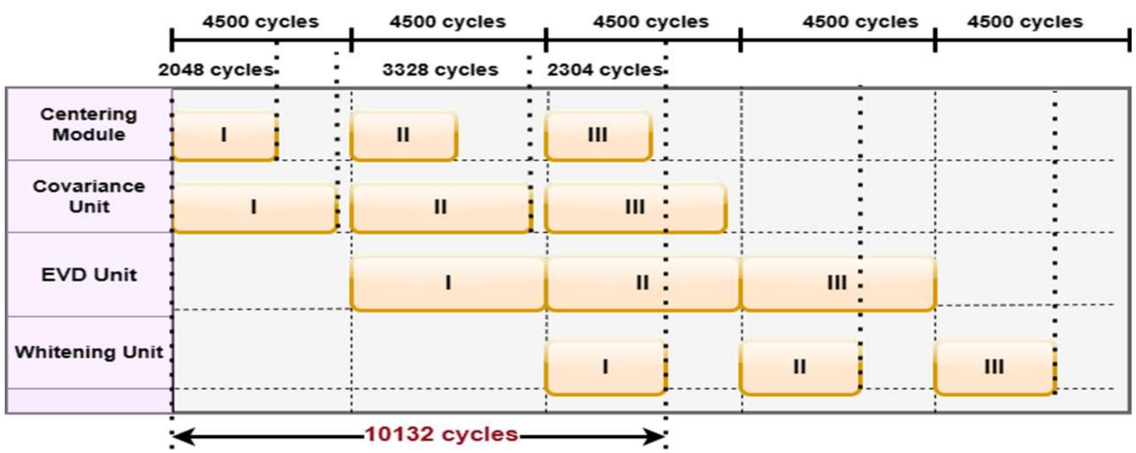
3x3 ECMMA



*Images obtained from [19]

TIMING DIAGRAM FOR PREPROCESSOR

- For pre-processing the signal meant for c-ICA of channels (M = 8) and sample length (K = 256).
- The centering module consumes 2048 cycles and the overall process concedes 3328 cycles to execute.
- Given (Ns = 4), for determining the eigenvalue and eigenvector, EVD engine requires 4500 cycles.
- The Whitening Unit consumes 2304 cycles.
- Overall, the proposed accelerator design requires a total of 10132 cycles.



EXPERIMENTAL RESULTS

- The proposed fixed-point c-ICA pre-processing accelerator was synthesized using GPDK 45 nm process node with a nominial voltage of 1.0 V.

Comparison of Proposed Centering and Covariance unit with SOTA Designs.

Parameters	[19]		[18]		[15]		Proposed
	Replicated	**Original**	**Replicated**	**Original**	**Replicated**	**Original**	**Original**
Year of Publication	2022	2022	2022	2022	2019	2019	-
Arithmetic Type	Fixed-Point	Fixed-Point	Fixed-point	Fixed-point	Floating-Point Single Precision	Floating-Point Single Precision	Fixed-Point
Word Length	10-bits	10-bits	26-bit	26-bit	32-bits	32-bits	10-bits
Architecture	Complex	Complex	Real	Real	Real	Real	Complex
Matrix Size	8x256	8x256	8x256	8x256	8x256	8x256	8x256
Node Size	45nm	90nm	45nm	90nm	45nm	90nm	45nm
Technology	GPDK45	TSMC	GPDK45	-	GPDK45	TSMC	GPDK45
Area (um^2)	212121.698	-	1024463.625	-	204856.411	-	164951.006
Total Complexity (kGE)	828.6	11.8	4001.811	840*	800.177	840*	644.34
Memory	Not included	-	included	included	included	included	Not included
Power (mW)	11.444	50.2	50.73	760*	5.591	65*	7.238
Frequency (MHz)	472	250	140	555	136	100	571
Critical Path (ns)	2.118	-	7.136	-	7.314	-	1.749
Clock Cycle (Single Matrix)	5120	5120	512	512	-	-	3328
SA Throughput	92.217	48.82	273.74	1079	-	-	98.232
SA Frequency	472	-	140	-	136	-	571

Comparison of Proposed EVD unit with SOTA Designs.

Parameters	[19]		[18]		[15]		[16]		Proposed
	Replicated	**Original**	**Replicated**	**Original**	**Replicated**	**Original**	**Replicated**	**Original**	**Original**
Year of Publication	2022	2022	2022	2022	2019	2019	2015	2015	-
Arithmetic Type	Fixed-Point	Fixed-Point	Fixed-point	Fixed-point	Floating-Point Single Precision	Floating-Point Single Precision	Fixed-Point	Fixed-Point	Fixed-Point
Word Length	10-bits	10-bits	26-bit	26-bit	32-bits	32-bits	16-bits	16-bits	10-bits
Architecture	Complex	Complex	Real	Real	Real	Real	Complex (Scaled)	Real	Complex
Matrix Size	8x8	8x8	8x8	8x8	8x8	8x8	8x8	8x8	8x8
Node Size	45nm	90nm	45nm	90nm	45nm	90nm	45nm	90nm	45nm
Technology	GPDK45	TSMC	GPDK45	-	GPDK45	TSMC	GPDK45	TSMC	GPDK45
Area (um^2)	8068.261	-	64242.754	-	30166.114	-	26400.6	54116*	6484.900
Total Complexity (kGE)	31.455	61.5	250.459	1808*	117.836	840*	102.926	131.89*	25.278
Memory	Not included	-	included	included	included	included	Not included	Not included	Not included
Power (mW)	0.870	50.2	4.502	760*	3.568	65.0*	2.568	0.0816	0.655
Frequency (MHz)	794	250	158	555	147	100	423	11	573
Critical Path (ns)	0.669	-	6.295	-	6.792	-	2.364	-	0.933
Clock Cycle (Single Matrix)	4540	4540	2688	2688	43008	-	-	-	4500
SA Frequency	1494	-	158	-	147	-	423	-	1071
SA Throughput	437.25	72.15	59.10	204.41	3.02*	-	-	-	238.20

CONCLUSION

- The proposed EVD design presents a maximum throughput of 238.20 kMatrices/s, and standalone (SA) operating frequency of 1.071 GHz, which is only second to **[19]** design.
- The **[19]** design suffers from footprint and power costs.
- The overall throughput of the proposed design for the entire pre-processing is around 56.43 kMatrices/s which is greater than 37.63 kMatrices/s [19] and (14057×140) = 17.6 kMatrices/s **[18]**
- The overall throughput has increased by 49.96%
- The complexity have decreased by 22.23% and 19.63% for centering cum covariance unit and EVD unit respectively.
- The proposed architecture provides an optimal balance between throughput, complexity and power.

IMPORTANT REFERENCES

1. L.-D. Van, T.-C. Lu, T.-P. Jung, and J.-F. Wang, “Hardware-oriented memory-limited online fastica algorithm and hardware architecture for signal separation,” in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 1438–1442. **[15].**

2. C.-H. Yang, Y.-H. Shih, and H. Chiueh, “An 81.6 μw fast ICA processor for epileptic seizure detection,” IEEE Transactions on Biomedical Circuits and Systems, vol. 9, no. 1, pp. 60–71, 2015 **[16].**

3. S. M. R. Shahshahani and H. R. Mahdiani, “FICA: A fixed-point custom architecture fast ICA for real-time and latency-sensitive applications,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 30, pp. 2896–2905, 2022. **[18]**

4. H.-H. Lu, C.-A. Shen, M. E. Fouda, and A. M. Eltawil, “Configurable independent component analysis preprocessing accelerator,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 30, no. 12, pp. 1840–1852, 2022. **[19]**

THANK YOU