International Journal of Computer Communication and Information System(IJJCCIS)
Vol 7. No.1 2015 Pp. 8-14

available at : www.ijcns.com
ISSN: 0976–1349

Design of MIMO-OFDM Systems Using SVD Technique

[1]R.Loganathan, [2]R.Saravanan, [3]P.Balaji, [4]S.Vinoth Kumar
[1]PG Scholar, [2]Assistant Professor, [3]Assistant Professor, [4]Project Coordinator
Department of ECE, Aksheyaa College of Engineering, Maduranthagam, Tamil Nadu
logu.nathan07@gmail.com

Abstract - **An adaptive Singular Value Decomposition (SVD) algorithm is an optimal method to obtain spatial multiplexing gain MIMO-OFDM systems. We present an orthogonal reconstruction scheme to obtain more accurate SVD outputs and then the system performance will be greatly enhanced. Moreover this project reduces high cost of implementation and high decomposing latency. Finally adaptive SVD engine is simulated in Modelsim 6.3 and synthesized in Xilinx 9.1i and implemented in Spartan 3E FPGA. Hence, the proposed reconfigurable adaptive SVD engine design is very suitable for high-throughput wireless communication applications.**

Keywords: SVD, MIMO, OFDM, FPGA

.

I. INTRODUCTION

Due to the rapid evolution of wireless communication and the demand of high data rate for multi-media information access in recent years, single-input single-output transmission has become insufficient for use [1], [2]. However, in a MIMO system, one receiving antenna may suffer from the interference of other transmitting antennas [3]. This makes it hard for the receiver to obtain correct data. By applying the singular value decomposition (SVD), the interference can be totally eliminated. Hence, the throughput and coverage of a MIMO system can be greatly enhanced. From an information theoretical viewpoint, the use of SVD can be claimed as an optimal solution [5]. In recent years, [6] proposed one ASIC realization of the SVD without the need of CSI for WLAN applications. A matrix decomposition architecture was proposed in [7] according to the Golub-Kahan SVD (GK-SVD) algorithm. To increase the processing speed, it only computes V and U which are partial to SVD outputs. Nevertheless, the above-mentioned SVD designs only support 4 × 4 (four transmitter and four receiver antennas) antenna configuration which is not sufficient for dealing with different antenna configurations.

Some of its key features are listed as follows.

1. Adaptive step size scheme, partial update scheme, and subcarrier inherit scheme (SIS) to effectively reduce the decomposing latency and increase the processing throughput.
2. Reconfigurable architecture for all antenna configurations in an MIMO system.
3. Early termination scheme to improve hardware utilization without losing system performance.
4. Data interleaving scheme to deal with several channel matrices simultaneously.
5. Orthogonal reconstruction (OR) scheme to enhance the system performance.

As compared with other related works, this chip achieves the highest throughput and power efficiency in the 4 x 4 SVD operations.

II. INTRODUCTION TO SVD TECHNIQUE

Singular value decomposition (SVD)

Singular value decomposition takes a rectangular matrix of general expression data (defined as A, where A is n x p matrix) in which the n rows represents the genes, and the p columns represents the experimental conditions. The SVD theorem states:

$$\mathbf{H = U\,S\,V^T}$$

Where **H** is the given original matrix. **U** is the left singular vectors (gene coefficient vectors). **S** (the same dimensions as A) has singular values and is diagonal (mode amplitudes). $\mathbf{V^T}$ has rows that are the right singular vectors (expression level vectors).

The SVD represents an expansion of the original data in a coordinate system where the covariance matrix is diagonal. Calculating the SVD consists of finding the Eigen values and eigenvectors of $\mathbf{AA^T}$ and $\mathbf{A^TA}$. The eigenvectors of $\mathbf{A^TA}$ make up the columns of **V**; the eigenvectors of $\mathbf{AA^T}$ make up the columns of **U**. Also, the singular values in **S** are square roots of Eigen values from $\mathbf{AA^T}$ or $\mathbf{A^TA}$. The singular values are the diagonal entries of the **S** matrix and are arranged in descending order. The singular values are always real numbers. If the matrix **A** is a real matrix, then **U** and **V** are also real.

Singular Matrix

The given **H** matrix is said to be singular the determinant value should be Zero.

$$|\mathbf{H}| = \mathbf{0}$$

Algorithm

Step 1: Determine the given **H** matrix is Singular.

Step 2: Find $\mathbf{H^T}$ from given **H** matrix.

Step 3: Multiply $\mathbf{H^T * H}$.

Step 4: To find Characteristic equation.

Step 5: To find the Eigen values and Eigen vectors.

Step 6: To find the **S** matrix.

Step 7: To find the **V** matrix.

Step 8: To find the **U** matrix.

Step 9: Multiply $\mathbf{U*S*V^T}$.

Step 10: Finally the original **H** matrix is obtained.

MIMO system model

Consider a MIMO system with N_T transmitter and N_R receiver antennas. The baseband, discrete-time equivalent model is written by

$$\mathbf{y = Hx + z}$$

Where $\mathbf{H} \in C^{NR\times NT}$ is the complex channel matrix. $\mathbf{z} \in C^{NR}$ is the additive white complex Gaussian noise vector.

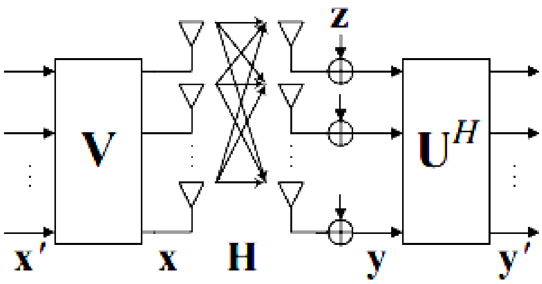

Fig.1: MIMO system with Adaptive SVD technique

$\mathbf{x} \in C^{NT}$ is the transmitted data vector. $\mathbf{y} \in C^{NR}$ is the received data vector. If we decompose the channel matrix **H** by SVD technique, we have

$$\mathbf{H = U\,\Sigma\,V^H}$$

Where **U** is $N_R\times N_R$ left singular matrix. **V** is $N_T \times N_T$ right singular matrix. **U** and **V** are unitary matrices (i.e., $\mathbf{UU^H = I}$ and $\mathbf{VV^H = I}$). $\mathbf{\Sigma}$ is an $N_R \times N_T$ matrix with only real and nonnegative main diagonal entries. The entry (i,i) of $\mathbf{\Sigma}$ denotes the i^{th} largest value σ_i , with $1 \le i \le \min(N_R,N_T)$. Let $\mathbf{x'} \in C^{NT}$ be the symbol vector.

$$\mathbf{x = Vx'}$$

The received signal **y** is multiplied by $\mathbf{U^H}$ as shown in Fig.1. The channel between **x'** and **y'** can be written as

$$\mathbf{y' = U^H\,y = U^H\,(Hx + z) = U^HHVx' + z' = \Sigma x' + z'}$$

Note that the distribution of **z'** is invariant since **U** is unitary. The MIMO channel can be treated as d = min (N_R, N_T) independent parallel Gaussian sub channels. The i^{th} sub channel has the gain being σ_i. Hence, the transmitter can send independent data streams across these parallel sub channels without any interference from an antenna. Note that the values $\sigma_1, \sigma_2, . . ., \sigma_d$ are called the singular values of **H**. The column vectors of **V** (i.e., $\mathbf{v}_1,\mathbf{v}_2, . . . ,\mathbf{v}_{NT}$) are the right singular vectors of **H.** The column vectors of **U** (i.e., $\mathbf{u}_1$, $\mathbf{u}_2$, . . . ,$\mathbf{u}_{NR}$) are the left singular vectors of **H**. n denotes the discrete time index. I denotes the identity matrix. (.) denotes the complex conjugate transpose of a matrix. The received signal y is used to estimate the autocorrelation matrix. The estimated autocorrelation matrix of $K_y = E[yy^H]$.

$\acute{k}_y$ is the estimated autocorrelation matrix of Ky. β is the forgetting factor and its choice depends on the stationary degree of the channel. d is the number of useful sub channels. The algorithm is meant to perform LMS-based estimation to find the pair (wi, λi). The step size μi controls the convergence speed and accuracy. The deflation process cancels the information of the pair (wi, λi) for the estimation of next pair (wi+1, λi+1). The deflation process continues until all pairs are estimated. The singular pairs (ui, σi) of channel matrix H can be derived by the use of the pairs (wi, λi) as follows:

$$\sigma i = \sqrt{\lambda i}\ ,\ \mu i = wi/\sqrt{\lambda i}\ ,\ i = 1, 2, \ldots, d.$$

Where σi - denotes the standard deviation. λi - denotes the Eigen values. μi - denotes the Step size. wi- denotes the sub carrier.

III. PROPOSED ADAPTIVE SVD ALGORITHM

In many MIMO OFDM-based communication standards, the channel matrix H can be obtained through channel estimation. With this additional information, we propose a complete adaptive SVD algorithm for high-throughput MIMO OFDM-based applications. The BER performance may be affected by imperfect channel estimation, H, and the degradation discussed in referenced works about the channel estimation.

Derivation of matrix R1

In Algorithm 1, the positive semi definite matrix R1 is estimated by a moving average of the recent received signal vectors. In many MIMO OFDM-based standards, the channel matrix H is already known by channel estimation. Therefore, we can utilize the information to evaluate accurate R1

$$R1 = \begin{cases} H^H H, N_R \geq N_T \\ HH^H, N_R < N_T \end{cases}$$

With this definition of R1, we can still use the same update and deflation process to find the pairs (wi, λi) sequentially. In the i^th update process, we have

$$wi(n+1) = wi(n) + \mu i(Ri - \lambda i(n)I)wi(n)$$

$$\lambda i(n+1) = wi(n+1)^H wi(n+1),\ i = 1, 2, \ldots, (d-1)$$

Where d is min (N_R,N_T), and the i^th deflation process is given by

$$Ri+1 = Ri - wi(n+1)wi(n+1)^H,\ i = 1, 2, \ldots, (d-1)$$

After convergence, we have wi and λi with 1 ≤ i ≤ (d−1).We can derive the singular values and the corresponding singular vectors of H by using the pairs (wi, λi). Since it is possible to have $N_R \geq N_T$ or $N_R < N_T$, there are two cases to be considered. For the case when $N_R \geq N_T$, we have

$$\sigma i = \sqrt{\lambda i},\quad vi = wi/\sqrt{\lambda i},\quad ui = \frac{Hvi}{\sigma i}, i = 1, 2, \ldots, (d-1).$$

On the other hand, when $N_R < N_T$, we only need to interchange vi with ui, and H is changed to H^H.

Partial update scheme

w_d and λ_d are derived by applying the update operation. From our observation, after the (d−1) -time deflation, the positive semi-definite matrix R_d can be expressed as

$$R_d = w_d w_d^H$$

Hence, the update operation for w_d and λ_d is unnecessary. We can directly find the dth singular value and the corresponding singular vectors by some simple operations. For the case of $N_R \geq N_T$, we get

$$\sigma_d = \sqrt{tr}\ (Rd\ )$$

$$v_d = R_d\ (:, 1)/||Rd\ (:, 1)||$$

$$u_d = Hv_d/\sigma_d$$

On the other hand, when $N_R < N_T$, we only need to interchange v_d with u_d, and H is changed to H^H. The advantage of applying partial update is to effectively reduce the decomposing latency.

Adaptive step size scheme

The step size μi is an important parameter for the convergence speed and stability of the algorithms. The objective function is a quadratic function which is complicated to derive the exact bound of the step size. We derive a loose bound by approximating the objective function from a quadratic function to quadratic function in Appendix. We have derived a convergence region and a near-optimal step size as follows:

$$0 < \mu i < \frac{1}{\lambda i}$$

and

$$\mu i = \frac{2}{3\lambda i - \lambda i + 1} > \frac{2}{3\lambda i}$$

Hence, fixed step size is inefficient and not robust for all kinds of channel matrices. The step size is adaptively adjusted as 0.05/λi(n) which is too small for fast convergence purposes. Therefore, for the goal of fast and stable convergence, the proposed adaptive step size is given by

$$\mu i(n) = \frac{a}{\lambda i\ (n)}$$

Where a is a scaling factor. We suggest that the value of a could be 0.75 or 0.5 for hardware friendly implementation.

SIS

We have to give the initial values of $\{w\ (n)\}_{d-1}^{i=1}$ for each update process. Although the update equation surely converges with arbitrary initial values, choosing good initial values can help to speed up the update processes. We denote wi(0) and wi(∞) as the initial and converged values of wi(n). In a wireless MIMO-OFDM system, since two adjacent subcarriers often have similar channel matrices, one subcarrier's converged information is useful to its adjacent subcarrier. Therefore, if one subcarrier's $\{w\ (\infty)\}_{d-1}^{i=1}$ is obtained, we can take the converged values as its adjacent subcarrier's initial values of $\{w\ (n)\}_{d-1}^{i=1}$. It should be noted that pilot and null subcarriers will be skipped since they do not need SVD operations.

To observe the effect of the subcarrier inherit scheme, we consider the channel model in a 128-subcarrier 4×4 system. Assume that the division-free adaptive step size with the early termination scheme is applied. When the subcarrier inherit scheme (SIS) has been included and excluded, Table III and Table IV show the averaged required iteration number in updating each pair (wi, λi) for μi(n) = 0.5/λi(n) and μi(n) = 0.75/λi(n), respectively. Note that with the partial update scheme, updating the last pair (w4, λ4) is unnecessary. As this shows, utilizing the subcarrier inherit scheme has the significant effect of reducing the total iterations by 19.1% and 21.5% for μi(n) = 0.5/λi(n) and μi(n) = 0.75/λi(n), respectively.

IV. ARCHITECTURE DESIGN OF ADAPTIVE SVD TECHNIQUE

The block diagram of the proposed reconfigurable adaptive SVD engine is depicted in Fig. 2. There are two single port SRAM banks in the memory module, and four 16 entries × 80 bits memory banks in the H buffers. It consists of six functional units which are zero padding unit, deflation unit, update unit, singular calculation unit, partial update unit, and simplified Gram–Schmidt unit. We could implement deflation unit directly and the block diagram of deflation unit is shown in Fig. 4. The register REG is used to store all entries of the positive semi-definite matrix. In the first update process, $R_i = R_1$. After the first update process, R_{i+1} is derived from R_i.

Reconfigurable design for different size of channel matrix

In a MIMO system, assume that the maximum number of transmitter and receiver antennas is M_R and M_T, respectively. This means that we have possibly $M_R x M_T$ different sizes of channel matrices (i.e., 1×1, 1×2, . . ., $M_R \times M_T$). Therefore, we propose a reconfigurable scheme to support all antenna configurations.

Zero padding scheme for square and non-square channel matrix

The maximum size of channel matrix is $M_R \times M_T$ in a MIMO system. Hence, it is sensitive to design an SVD engine to support the maximum channel size. For the smaller channel matrix, we can extend it to the maximum-size channel matrix by inserting zeros. If the size of a given matrix is $N_R \times N_T$, the extended channel matrix is

$$\mathbf{H}_{\text{extended}} = \begin{bmatrix} \mathbf{H}_{N_R \times N_T} & \mathbf{0}_{N_R \times (M_T - N_T)} \\ \mathbf{0}_{(M_R - N_R) \times N_T} & \mathbf{0}_{(M_R - N_R) \times (M_T - N_T)} \end{bmatrix}_{M_R \times M_T}$$

After extending the original channel matrix by inserting zeros, the SVD operation of the original channel is exactly the same as that of the maximum-size channel matrix. Support the antenna configurations after some modifications based on their own SVD algorithms.

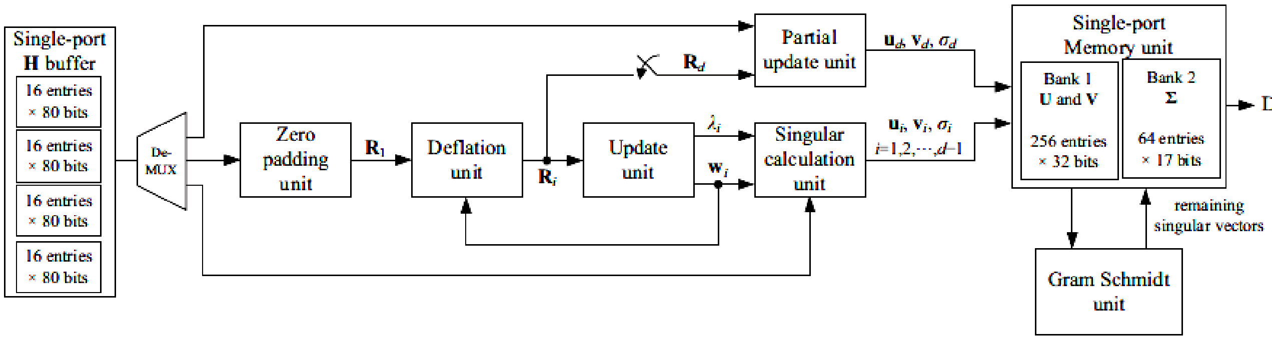

Fig.2: Block diagram of adaptive SVD technique

Therefore, d is still equal to min (N_R, N_T). Fig.3 shows the block diagram of zero padding unit. A given channel matrix $H_{NR\times NT}$ is extended to $H_{MR\times MT}$ by inserting zeros, and the multiplexer is used to construct the positive semi-definite matrix R_1. We also apply the zero padding Scheme to singular calculation unit and partial update unit.

Zero padding is used to refill the maximum information in place of null information position in H buffers. The maximum size of channel matrix is $M_R \times M_T$ in a MIMO system. Hence, it is intuitive to design an SVD engine to support the maximum channel size. For the smaller channel matrix, we can extend it to the maximum-size channel matrix by inserting zeros. If the size of a given matrix is after extending the original channel matrix by inserting zeros, the SVD operation of the original channel is exactly the same as that of the maximum-size channel matrix.

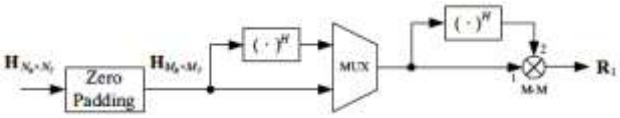

Fig.3: Block diagram of zero padding unit

Configurations after some modifications based on their own SVD algorithms. Let number of transmitters T=2; Let number of receivers R=2; Then, channel matrix=2*2=4. Hence, we should form the input blocks in to 4*4 matrix format. If suppose, the input is not available in 4*4 matrix format, then we should equalize this matrix size using zero padding unit.

Architectural design of update unit

The main computational time of our SVD architecture is in the update unit. Fig. 5 shows the block diagram of the original update unit. For the architectural design of the update unit, we propose three schemes to reduce the decomposing latency and enhance the hardware utilization.

Derivation of output Parameter Ui

Step:1

Select the first column of the 3*3 matrix R_d.

First column elements= [1 1 1];

Find the sum of first column elements=3

Step:2

Find normalization= (first column elements)/(sum of first column elements)

Normalization= [1/3 1/3 1/3]=[0.3 0.3 0.3]

Step:3

Determine M and N;

M=normalization output;

N=ouput from zero padding unit;

Step:4

$Ui = (M*N)/\sigma_d$;

Data interleaving scheme to deal with several channel matrices simultaneously.

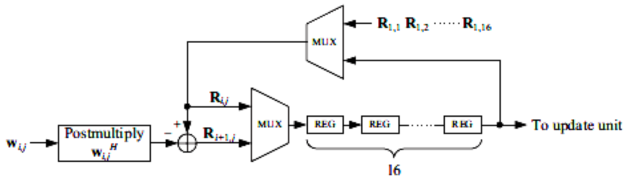

Fig.4: Block diagram of deflation unit

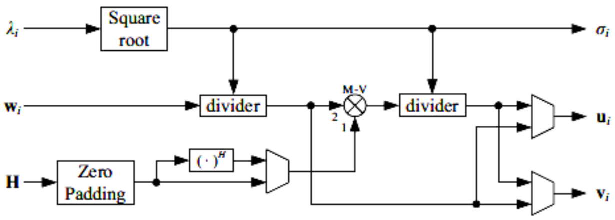

Fig.6: Block diagram of singular calculation unit

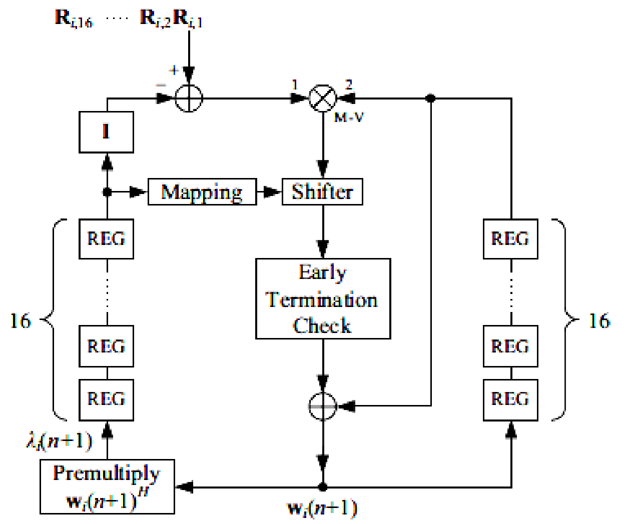

Fig.5: Block diagram of original update unit

Architectural design of partial update unit

In Algorithm 1, $\mathbf{w}_d$ and λ_d are derived by applying the update operation. From our observation, after the (d−1)- time deflation, the positive semi-definite matrix $\mathbf{R}_d$ can be expressed as

$$\mathbf{R}_d = \mathbf{w}_d\mathbf{w}_d^H$$

Hence, the update operation for $\mathbf{w}_d$ and λ_d is unnecessary. We can directly find the dth singular value and the corresponding singular vectors by some simple operations. For the case of $N_R \geq N_T$, we get

$$\sigma_d = \sqrt{\text{tr}}\,(\mathbf{R}_d\,),\ \mathbf{v}d = \mathbf{R}_d\,(:,1) \,/\, \|\mathbf{R}_d\,(:,1)\|,\ \mathbf{u}_d = \mathbf{H}\mathbf{v}_d \,/\, \sigma_d$$

On the other hand, when $N_R<N_T$, we only need to interchange $\mathbf{v}_d$ with $\mathbf{u}_d$, and $\mathbf{H}$ is changed to $\mathbf{H}_H$. The advantage of applying partial update is to effectively reduce the decomposing latency.

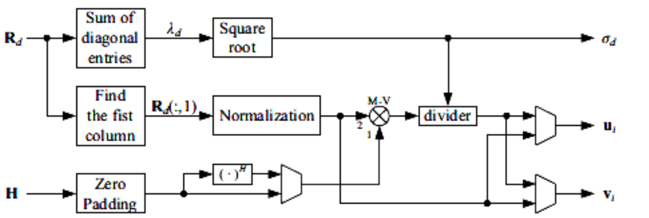

Fig.7: Block diagram of partial update unit

Architectural design of Gram-Schmidt unit

For an $N_R \times N_T$ channel matrix, we need to find d singular values, N_R left singular vectors, and N_T right singular vectors, where $d=\min(N_R,N_T)$. After applying the above schemes, we can find d singular values, d left singular vectors, and d right singular vectors. If the channel matrix is square, it means that $d=N_R=N_T$. Therefore, we can find all singular values and singular vectors. But for the case of non-square channel matrix, assume that $N_R>N_T$, we have $d=N_T$, there are still (N_R-N_T) unsolved left singular vectors (i.e., $\mathbf{u}N_T+1$, $\mathbf{u}N_T+2, \ldots, \mathbf{u}N_R$) after applying the above schemes. On the other hand, when $N_R<N_T$, there are (N_T-N_R) unsolved right singular vectors (i.e., $\mathbf{v}N_R+1$, $\mathbf{v}N_R+2, \ldots, \mathbf{v}N_T$). Note that both the cases are similar. To find these remaining vectors, recall that $\mathbf{U}$ and $\mathbf{V}$ are the unitary matrices, the column vectors in $\mathbf{U}$ or $\mathbf{V}$ are orthonormal to each other. That is

$$[\mathbf{u}i\,,\mathbf{u}j] = 0, \quad \forall i \neq j$$

and

$$[\mathbf{v}i\,,\mathbf{v}\,j] = 0, \quad \forall i \neq j$$

Therefore, the remaining vectors can be obtained by applying the Gram–Schmidt technique. First, we consider the case of $N_R>N_T$. After applying the above schemes, we already have $\mathbf{u}1, \mathbf{u}2, \ldots$, and $\mathbf{u}N_T$.

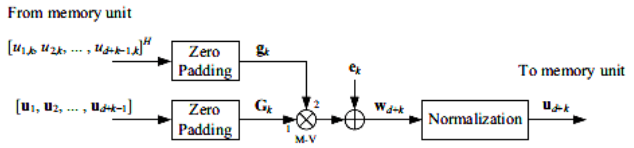

Fig.8: Block diagram of Gram-Schmidt unit

V.RESULT ANALYSIS

Device utilization summary

Selected Device: 3s250epq208-4

Number of Slices : 149 out of 2448 6%

Number of 4 input LUTs: 279 out of 4896 5%

Number of IOs : 291

Number of bonded IOBs : 291 out of 158 184%

IOB Flip Flops : 128

Number of MULT18X18SIOs: 4 out of 12 33%

Number of GCLKs : 1 out of 24 4%

Timing summary

Speed Grade : -4

Minimum period : No path found

Minimum input arrival time before clock : 16.452ns

Maximum output required time after clock: 4.283ns

Maximum combinational path delay: No path found

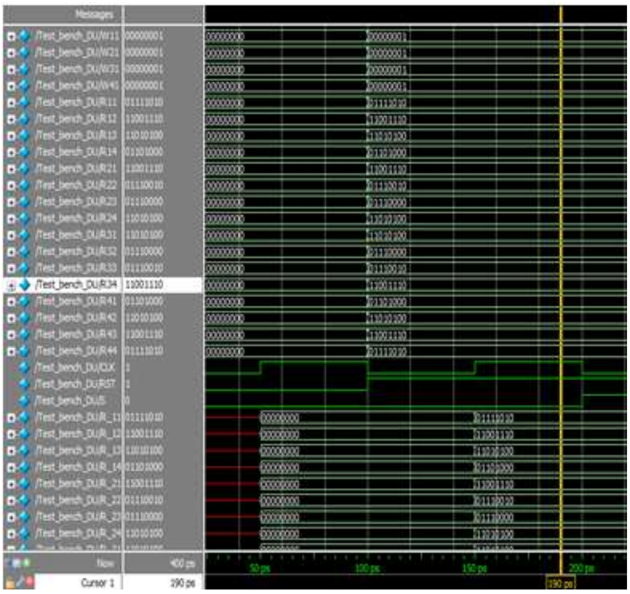

Fig.9: Simulation Result

VI.CONCLUSION

This paper presented a reconfigurable adaptive SVD engine design for MIMO-OFDM systems. The proposed architectural design techniques can lower the computational complexity, effectively reduce the decomposing latency, and support all antenna configurations in a MIMO system. These design strategies enable the use of SVD to be effectively applied to the high-throughput wireless communication applications.

VII.REFERENCES

[1] N. Seshadri and J. H. Winters, “Two signaling schemes for improving the error performance of frequency-division-duplex (FDD) transmission systems using transmitter antenna diversity,” in Proc. IEEE 43rd Veh. Technol. Conf., May 1993, pp. 508–511.

[2] S. M. Alamouti, “A simple transmit diversity technique for wireless communications,” IEEE J. Sel. Areas Commun., vol. 16, no. 8, pp. 1451– 1458, Oct. 1998.

[3] I.E. Telatar, “Capacity of multi-antenna Gaussian channels,” Eur. Trans.Telecommun., vol. 10, no. 6, pp. 585–595, 1999.

[4] D. J. Love and R. W. Heath, Jr., “Equal gain transmission in multiple input multiple-output wireless systems,” IEEE Trans. Commun., vol. 51, no. 7, pp. 102–1110, Jul. 2003.

[5] J. Ha, A. N. Mody, J. H. Sung, J. R. Barry, S. W. Mclaughlin, and G. L.Stüber, “LDPC coded OFDM with alamouti/SVD diversity technique,” Wireless Personal Commun., vol. 23, no. 1, pp. 183–194, Oct. 2002.

[6] D. Markovic, B. Nikolic, and R. W. Brodersen, “Power and area minimization for multidimensional signal processing,” IEEE J. Solid-State Circuits, vol. 42, no. 4, pp. 922–934, Apr. 2007.

[7] Y. G. Li, J. H. Winters, and N. R. Sollenberger, “MIMO-OFDM for wireless communications: Signal detection with enhanced channel estimation,” IEEE Trans. Commun., vol. 50, no. 9, pp. 1471–1477, Sep.2002.

[8] C. Studer, P. Blösch, P. Friendli, and A. Burg, “Matrix decomposition architecture for MIMO systems: Design and implementation trade-offs,” in Proc. 41st Asilomar Conf. Signals, Syst., Comput., Nov. 2007, pp.1986–1990.

[9] C. Senning, C. Studer, P. Luethi, and W. Fichtner, “Hardware-efficient steering matrix computation architecture for MIMO communication system,” in Proc. IEEE Int. Symp. Circuits Syst., May 2008, pp. 304–307.