Adaptive Resource Scheduling for Energy Efficient QRD Processor with DVFS

Liu, Yangxurui; Prabhu, Hemanth; Liu, Liang; Öwall, Viktor

2015

Adaptive Resource Scheduling for Energy Efficient QRD Processor with DVFS

Yangxurui Liu, Hemanth Prabhu, Liang Liu, and Viktor Öwall
Department of Electrical and Information Technology
Lund University
{yangxurui.liu, hemanth.prabhu, liang.liu, viktor.owall}@eit.lth.se

***Abstract*—This paper presents an energy efficient adaptive QR decomposition scheme for Long Term Evolution Advance (LTE-A) downlink system. The proposed scheme provides a performance robustness to fluctuating wireless channels while maintaining lower workload on a reconfigurable hardware. A statistic based algorithm-switching strategy is employed in the scheme to achieve workload reduction and stable computing resource requirement for QR decomposition. With run time resource allocation, computing resources are assigned to highest performance gain segments to reduce performance loss. By utilizing the dynamic voltage and frequency scaling (DVFS) technique, we further exploit the potential of power saving in various workload situation while maintaining fixed throughput. The proposed technique brings power reduction upto 57.8% in EVA-5 scenario and 24.4% with a maximum SNR loss of 1 dB in EVA-70 scenario, when mapped on a coarse grain reconfigurable vector-based platform.**

***Index Terms*—QR decomposition, multiple-input multiple-output (MIMO), dynamic voltage and frequency scaling (DVFS)**

I. Introduction

Multiple-input multiple-output (MIMO) technique is adopted in LTE-A standard for its capability in increasing system data rate without extra bandwidth requirement. To implement practical MIMO system, pre-processing of the channel state information (CSI), e.g. QR decomposition (QRD) [1] is essential for various MIMO detection algorithms, like sphere decoder [2]. To tackle the high computing cost of QRD, several algorithms [3], [4] have been proposed. These low complexity implementations rely on approximations of QRD by leveraging wireless channel feature. Comparing with exact (brute force) QRD, these algorithms bear different degradation in system performance for different scenarios. Runtime adaptive processing that alternates current algorithm to fit changing external environment is a promising method to reduce computational burden on hardware, while still keeping robust performance in changing environment. From a hardware perspective, reduction in computation complexity directly leads to reduction in power consumption, since it decreases activity ratio of transistors. Benefiting from the diverse workload of adaptive processing, the potential of power saving could also be exploited by dynamic voltage and frequency scaling (DVFS) while still maintaining a constant throughput. DVFS is a well known low power technique [6], where both supply voltage and frequency can be decreased, when the workload of a processor is low. To meet the latency requirement during transmission, resource control schemes are necessary for realizing DVFS. [5].

This paper exploits the aforementioned low-power design techniques, by demonstrating energy efficient QRD for LTE-A downlink system. The adaptive scheme is proposed for adjusting algorithms to fit external environment and minimize the performance loss with reduced working frequency. To efficiently implement adaptive processing, we leverage the reconfigurability of a coarse grain vector processor for mapping adaptive algorithms. Using a 65 nm CMOS library set, we test achievable frequencies of the platform to set up work modes for several available supply voltages with post-layout results. Based on that, the performance simulation result shows that the scheme manages to control performance loss around 1 dB at 10^{-2} frame error rate (FER) in EVA-5 and EVA-70 scenarios with dedicated work mode. On the other side, the achieved power saving is 57.8% and 24.4% in EVA-5 and EVA-70 respectively.

II. Background

A. QR update scheme

There are several well-known QRD algorithms, Gram-Schmidt [7], Given rotation [8] and Householder [9]. Based on that, researches aimed to reduce the complexity of QRD by exploiting wireless channel characteristic, e.g. exploiting the correlation within time domain and frequency domain [3] [10] [11]. The method proposed in [3] takes advantage of the time correlation between two adjacent channel state information. This method is very effective in reducing complexity, however, its performance largely depends on time correlation which could be parametrized as Euclidean Distance (ED) between CSI matrices $\boldsymbol{H}$.

The following equations are in-line with [3], wherein a MIMO system with N transmit antennas, N receive antennas is proposed. The QR update algorithm contains 2 steps, first an exact computation is performed as

$$\boldsymbol{H}_{\text{old}} = \boldsymbol{Q}_{\text{old}}\boldsymbol{R}_{\text{old}} \tag{1}$$

and then ultilizes approximate new CSI as

$$\boldsymbol{H}_{\text{new}} = \boldsymbol{Q}_{\text{new}}\boldsymbol{R}_{\text{new}} = \boldsymbol{Q}_{\text{old}}(\boldsymbol{Q}_{\text{old}}^{H}\boldsymbol{H}_{\text{new}}) = \boldsymbol{Q}_{\text{old}}\tilde{\boldsymbol{R}}_{\text{new}}. \tag{2}$$

(1) denotes a brute force QRD process which is referred to as Step-I and (2) denotes the next QR update process (Step-II) based on result obtained in (1). This tracking scheme

TABLE I: Complexity of candidate QR algorithms

	Multiplication*	Pure Brute Force QRD (case-I)	Hybrid QRD (case-II)	Pure QR update (case-III)
Brute Force ◇	$N^3 + N^2$	√	√	√
QR Update	$NN^{\dagger}_{update}$		√	√
ED Measuring	N_{update}		√	

* All multiplications are complex-valued.
◇ In this paper, we use Gram-Schmidt algorithm for brute force QRD.
† N_{update} denotes the number of updated elements in whole matrix.

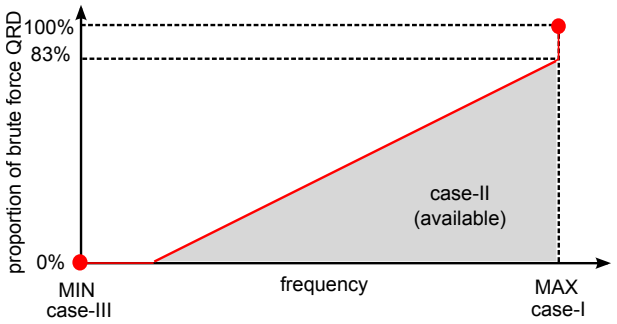


Fig. 1: Relationship between proportion of brute force QR on Step-II and frequency requirement ($N = 4$, $N_{update} = 8$).

introduces errors since there is no guarantee that $\tilde{\boldsymbol{R}}_{\text{new}}$ will be an upper-triangle matrix. The performance of this method is largely dependent on the changing rate of $\boldsymbol{H}$ which is positively correlative with Doppler shift of the terminal. Euclidean distance between these two matrices could be used as a parameter to arbitrate with a threshold for which algorithm will be used in the following QR decomposition to control error rate with an acceptable loss. Thus we obtain 3 candidate algorithms, pure brute force QRD, pure QR update, and hybrid QR, determined by the process on Step-II.

The complexity and constitution of these QR decomposition schemes are shown in Table. I. It is important to note that even in case-III, the brute force QR is an essential procedure since it requires an initial state. In hybrid QRD scheme, ED measuring is an essential step which determines which process will be chosen for the current QRD. Hence, the measuring and choosing strategy will influence the proportion of brute force QRD and QR update in large time scale and leads to variation on the average complexity of each QRD.

B. Hardware Computing Resource Model

To assist the requirement of our resource scheduling scheme, we briefly introduce a computing resource model. In a certain period of time $\boldsymbol{T}_{total}$ processing $\boldsymbol{n}$ QRD, the amount of available computing resource $\boldsymbol{S}$ is defined by the capacity of average operation per clock cycle $\boldsymbol{C}$ and the clock cycle number $\boldsymbol{n}_{clk}$. $\boldsymbol{C}$ is a constant characteristic of the target hardware that denotes the fitness between hardware and algorithm. Clock cycle period $\boldsymbol{T}_{clk}$ is reciprocal to frequency $\boldsymbol{f}$. Thus we determined that the amount of operation resource we could obtain in a certain period of time is proportional to the current operational frequency $\boldsymbol{f}$. The relationship between frequency requirement and proportion $\boldsymbol{p}$ of brute force in all QRD is

$$\begin{aligned}\boldsymbol{S} = \boldsymbol{C} \times \boldsymbol{n}_{clk} = \boldsymbol{C} \times \frac{\boldsymbol{T}_{total}}{\boldsymbol{T}_{clk}} = \boldsymbol{C}\boldsymbol{T}_{total} \times \boldsymbol{f} \\ \geq \begin{cases} \boldsymbol{n}R_b & \text{case-I} \\ \boldsymbol{p}\boldsymbol{n}R_b + (1-\boldsymbol{p})\boldsymbol{n}R_u + \boldsymbol{n}R_t & \text{case-II} \\ \boldsymbol{n}R_u & \text{case-III.} \end{cases}\end{aligned} \tag{3}$$

The complexities of QR update, brute force QRD, and algorithm decision-making are represented as R_u, R_b, and R_t, respectively. The amount of operation is determined by the selected strategy, and in hybrid schemes influenced by the proportion of brute force algorithm $\boldsymbol{p}$. Conversely, the upper bound of proportion of brute force algorithms is determined by current frequency in advance, as shown in Fig. 1.

III. Real Time Self-adaptive QRD scheme

To fully benefit from the potential workload reduction in the aforementioned hybrid QR scheme, we propose a real time self-adaptive QRD scheme according to current external environment and the supply voltage and frequency. The performance loss between hybrid algorithms and pure brute force varies in different scenario, e.g. QR update scheme suffer lower performance loss in low speed (mobility) scenario than in high speed scenario. Thus the workload requirement or complexity of update scheme is largely relevant to real time circumstance and scheme threshold we chosen, while maintaining the throughput to be a constant is also an essential requirement at system level. To avoid conflicts occurring between the upcoming QRD frames, throughput of QRD in a certain number of frames must be fixed. This is an important prerequisite for this scheduling scheme. In order to solve the fixed throughput problem, gap between computing resource consumption and resource budget must be tackled.

A. Real Time Computing Resource Allocation Scheme

The number of QRD in a certain time interval, e.g. within one frame, is determined by LTE-A standard and the number of sub-carriers in assigned bandwidth. The initial computing resource we have is defined as clock cycle we have in a frame. Using this definition, we can simplify our problem to allocation limited computer resource to certain number of processes. The remaining resource for k-th QRD operation is formulated as $R_{k+1} = R_k - C_k$, where C_k determined the complexity consumption in current QRD. Each time node that we have to decide which algorithm in the following QRD is to be used are regarded as *decision point*, as shown in Fig. 2,. The performance obtained in k-th QRD is determined by the chosen algorithm and external environment e_k, regarded as $P_k(C_k, e_k)$. The scheme problem is simplified as maximizing the overall performance

$$\Psi_{opt} = \max_{\sum C_k = S} \sum_{k=1}^{n_{QR}} P_k(C_k, e_k), \tag{4}$$

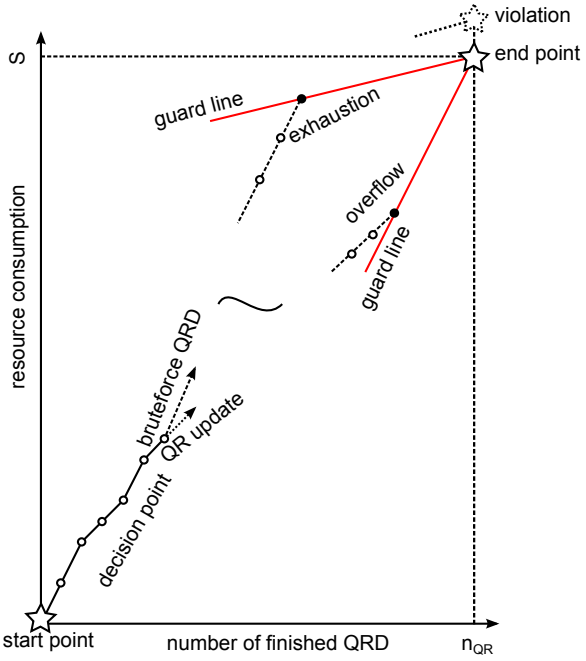


Fig. 2: Illustration of resource allocation of n_{QR} QRD with $\boldsymbol{S}$.

TABLE II: Pseudo-Code of the Proposed Greedy Scheme.

$R = S; state = normal;$	*% initial statement*
for $k = n_{QR} : 1$ **do**	*% index of unfinished QR*
case $state$	
$exhaustion:$	
$\boldsymbol{QRupdate}$	*% keep QR update from now on*
$overflow:$	
$\boldsymbol{QRbruteforce}$	*% keep brute-force QR from now on*
$normal:$	
$\boldsymbol{EUtest}$	
if $EUtestresult > \theta$	*% decision point*
$\boldsymbol{QRbruteforce}$	
$R = R - R_t - R_b$	*% decrease remaining resource*
else	
$\boldsymbol{QRupdate}$	
$R = R - R_t - R_u$	*% decrease remaining resource*
endif	
if $R \geq kR_b$	*% guard line*
$state = overflow$	
if $R \leq kR_u$	*% guard line*
$state = exhaustion$	
endcase	
endfor	

which is also known as the $0-1$ $Knapsack\ Problem$ [12]. In following sections, different schemes are described and later we evaluate the performance of these schemes.

1) Greedy Resource Allocation Scheme: External environment can't be obtained until the next time unit, thus no optimal resource allocation scheme could be obtained in real life system. A simple greedy algorithm is proposed in Table. II that allocates computing resources for n_{QR} QRD. It is greedy and a low-complex algorithm requiring addition and subtraction. It starts with initial remaining resource $R = S$ and the starting state as $normal$. In each check point, we identify the next algorithm chosen by testing Euclidean distance between previous CSI matrix H_{k-1} and current H_k, where the subscript k is the time index. It must be pointed out that when the remaining resource is more than using brute force algorithm in the following QRD, simply running brute force QRD will avoid extra performance loss in accurate without risking in time-out. This phenomenon is named as resource $overflow$ in this paper. $Overflow$ will cause loss in performance for those previous QRD which could have been solved using an accurate algorithm. On the contrary, the phenomenon that remaining resource is less than using QR update in the following QRD is named as resource $exhaustion$. $Exhaustion$ will also cause loss in performance due to all further QRD must be solved using an inaccurate algorithm. $Overflow$ and $exhaustion$ are collectively referred to as $abnormal$ state and should be avoid. The current state will be transferred to $overflow$ state or $exhaustion$ state while the remain computing resource approach the resource guard line.

2) Threshold Adjust Scheme: Greedy scheme introduces a low complexity method for a specific external environment. However, due to the varied external environment, fixed threshold could introduce a mass of $overflow$ or $exhaustion$ state, which leads to performance loss. In this case, runtime adapting threshold to environment to avoid $exhaustion$ or $overflow$ state is a more efficient method to avoid unnecessary performance loss. Since exact estimation of future ED is very difficult to obtain, real time update based on current resource and environment situation is a sub-optimal method.

We propose a scheme that divide a large time interval into m sub time units and update threshold in each sub time unit to obtain a revised threshold. The main idea of the scheme is based on the portion p' of QR update in the remaining computing resource, updating threshold through the distribution of ED. The process of obtaining real time portion is shown in (5) where $\boldsymbol{R'}$ and n' denotes remaining computing resource and the number of QRD to proceed, respectively. In each sub time unit, the threshold is updated based on

$$p' = \frac{n'\boldsymbol{R_t} + n'\boldsymbol{R_u} - \boldsymbol{R'}}{\boldsymbol{R_b} - \boldsymbol{R_u}} \tag{5}$$

One straight-forward way to find the ideal threshold for current sub time unit is to sort all ED and choose the $\frac{np'}{m}$-th largest ED as the revised threshold. However, the complexity of sorting algorithms, $O(\frac{n}{m}log\frac{n}{m}) \sim O((\frac{n}{m})^2)$ [13] [14], is unacceptable for threshold decision. More importantly, it requires the buffering of the data of a whole frame, resulting in high latency.

Near-ideal threshold $\boldsymbol{\theta}$ could be calculated with p' and inverse cumulative distribution function (CDF). According to the experimental results of ED distribution as shown in Fig. 3, the distribution of ED in a certain frame is irregular. However, to simplify the problem, we assume that the distribution of

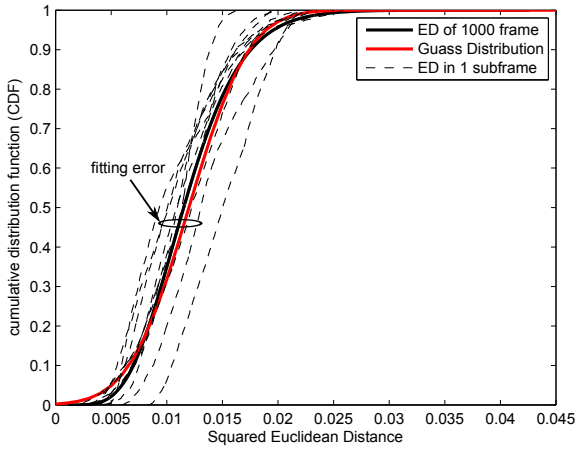


Fig. 3: Distribution of squared Euclidean distance between full-H pilot and half-H pilot in 3GPP Extended Vehicular A with a maximum Doppler shift of 70 Hz (EVA-70).

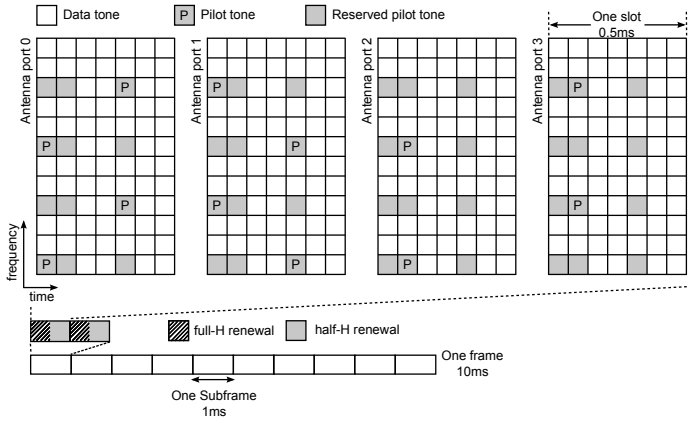


Fig. 4: LTE-A time-frequency resource structure of 4×4 MIMO system.

ED in a subframe approximated as a Gaussian distribution. Potential error will be introduced due to this approximation. Nevertheless, its low complexity makes it a promising method in practical implementation. (6) denotes quantile function for threshold estimation [15] where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ refers to the mean value and variance of ED distribution in the previous time unit. erf^{-1} is an inverse Gauss error function, which can be defined and expanded in terms of Maclaurin series as

$$\begin{aligned} \mathrm{F}^{-1}(\boldsymbol{p};\boldsymbol{\mu},\boldsymbol{\sigma}) &= \boldsymbol{\sigma}\boldsymbol{\Phi}^{-1}(\boldsymbol{p}) + \boldsymbol{\mu} \\ &= \sqrt{2}\boldsymbol{\sigma}\,\mathrm{erf}^{-1}(2\boldsymbol{p}-1) + \boldsymbol{\mu} \\ &\approx \sigma\frac{\sqrt{2\pi}}{2}((2\boldsymbol{p}-1) + \frac{\pi}{12}(2\boldsymbol{p}-1)^3) + \boldsymbol{\mu} \\ &= \boldsymbol{\theta}. \end{aligned} \tag{6}$$

Considering the range of the input value $\boldsymbol{p}$ is [0,1] and the accuracy we get, only first two Maclaurin expansions is chosen to estimate near-ideal threshold.

To avoid unnecessary variation during the tracking, We also consider a forgetting factor β to track ED distribution between the past 2 subframes

$$\begin{aligned} \hat{\boldsymbol{\mu}}_i &= \beta\boldsymbol{\mu}_i + (1-\beta)\hat{\boldsymbol{\mu}}_{i-1} \\ \hat{\boldsymbol{\sigma}}_i &= \beta\boldsymbol{\sigma}_i + (1-\beta)\hat{\boldsymbol{\sigma}}_{i-1}. \end{aligned} \tag{7}$$

The current treated characteristics, denoted as $\hat{\boldsymbol{\mu}}_i$ and $\hat{\boldsymbol{\sigma}}_i$, can be expressed using current raw characteristics and previous treated characteristics. When $\beta = 1$, then the estimated characteristic of current frame is equivalent to the statistical result of last frame. By introducing the forgetting factor β, a more stable threshold can be achieved, which potentially is favourable for performance of entire system.

B. Performance Evaluation

The performance is evaluated under a framework using LTE-A standard. A MIMO system of 4 transmitting antennas and 4 receiving antennas is simulated with K-best MIMO detector using $K = 10$. The bandwidth is 5 MHz and the number of used sub-carriers is 300. 64-QAM modulation is used and coded with a rate 1/2 parallel concatenated turbo code. Specification of resource in frequency and time domain on LTE-A standard shown in Fig. 4, ED are determined as the Euclidean distance between full-H renewal and half-H renewal. Resulting from only half channel information is updated in half-H renewal, algorithms selection is focused on half-H renewal for limited computing resource. Thus in full-H renewal, the brute force QR procedure is always applied. The resource allocation will be restricted within one frame, thus the amount of resource is a constant within a certain frame. To specify proposed strategy, a 65 nm ST CMOS standard cell library set with a nominal supply voltage of 0.9 V, 1.0 V, and 1.1 V is respectively assigned to aforementioned 3 schemes, pure update, hybrid algorithms scheme, and pure brute force. Using pure brute force running with supply voltage 1.1 V as reference, post-layout simulation result shows that 60% brute force is achievable in hybrid algorithms with a supply voltage of 1.0 V. Fig. 5 compares the proportion of $overflow$ state and $exhaustion$ state with different adjustment scheme in an initial brute force QR proportion of 60%. Comparing with the length of $overflow$ state and $exhaustion$ state between fixed threshold strategy and the dynamic threshold strategy, the proposed threshold adjustment achieves an improvement in decreasing the proportion of $abnormal$ state.

To further verify our scheme, we investigate the frame error rate (FER) as performance indicator for strategies with different resource supply and Doppler shift scenarios as shown in Fig. 6. For completeness, we includes the pure QR update schemes and the pure QR brute force schemes which are in different computing resource level. Moreover, we also includes the fixed threshold greedy resource allocation in which the threshold is kept constant fix at 60% of cumulative percentage as contrast. Fig. 6(a) and Fig. 6(b) show respectively the performance results with EVA-5 and EVA-70, using the same constellations in the downlink phases. Comparing to the one with fixed threshold greedy resource allocation, it clearly

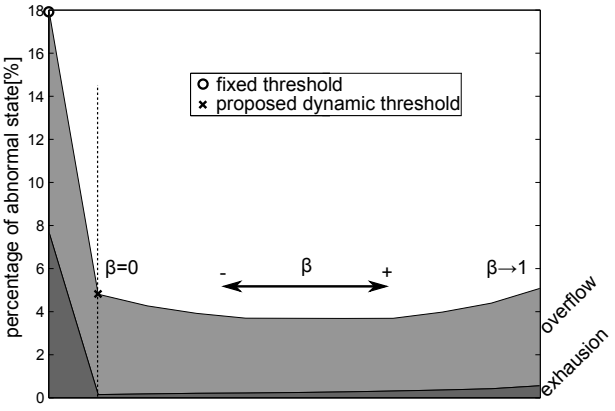


Fig. 5: Proportion of overflow state and exhaustion state with different threshold adjustment scheme in EVA-70.

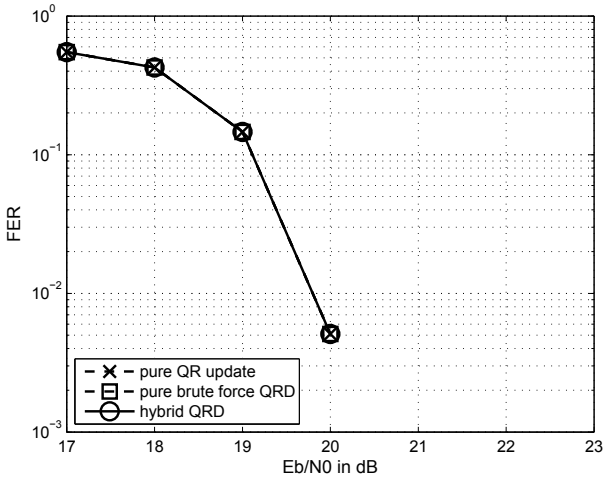
shows that run time threshold adjusting scheme achieves about 1.5 dB improvement at FER=10^{-2} in EVA-70 scenario. Moreover, we also investigate the influence of forgetting factor β in Fig. 7. Comparing with the different forgetting factor β, we do not find a big difference in FER. A variation range of SNR about 0.2 dB for a FER of 10^{-2} is provided.

IV. Hardware Implementation with DVFS

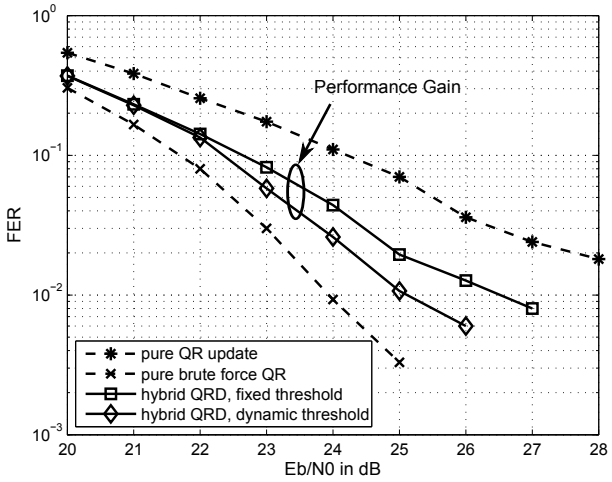
To evaluate the proposed scheme, we mapped the algorithms on a vector-oriented reconfigurable platform [4]. As previously discussed, the platform is implemented in 65 nm $STmicroelectronics$ CMOS technology and measured with a library set of different supply voltages in $Common\ Power\ Format$ (CPF) low power design flow. In Table. III, detailed number of multiplications in both full-H and half-H renewal and average numbers of multiplications per QRD procedure in different schemes are shown. The platform is capable for 16 complex-value multiplications simultaneously. Due to the high reconfigurable of the target platform, the utilization of multipliers approaches 100% with dedicated mapping. In order to handle against data dependency in QR procedure, multi QR procedure on different sub-carriers are paralleled mapped on the platform to fully leverage computing resource and compressed execution time.

To implement the multi-scheme, we use the aforementioned library set with 3 supply voltages, 1.1 V, 1.0 V, and 0.9 V. Running pure brute force QRD with the highest supply voltage 1.1 V in 500 MHz is guiding the throughput for rest supply voltages. Based on that, 1.0 V and 0.9 V is used for testing hybrid algorithm and pure QR update separately. The proportion $\boldsymbol{p}$ in hybrid algorithm is decided by the achievable frequency with a precondition of fixed throughput. As a result, Table. IV shows the manual mapping result of clock cycle per QRD in different supply voltages and post-layout power simulation result. The results show that a 57.8% power reduction is achievable with supply voltage scaling technique.

Fig. 8 shows the relationship between voltage supply and power consumption with the system performance of wireless transmission. It must be noted that dash lines in Fig. 8 only



(a) Performance comparison for different workloads in EVA-5.



(b) Performance comparison for different workloads in EVA-70.

Fig. 6: FER performance of different scheme for transmitting 64-QAM symbol over 4×4 MIMO channels with turbo coding of code rate 1/2 in different scenarios.

denote the tendency of performance loss and do not imply measured result. The lowering of supply voltage and frequency will lead to shrinking in computing resource which inevitably causes increase in performance loss compared with brute force QRD. This tendency expands with the increase in the Doppler shift. Combining the resource allocation scheme and DVFS has a twofold advantage, firstly the tendency of performance degradation is restrained, and the energy consumption is lowered significantly. Furthermore, the proposed scheme is robust with respect to channel conditions and does not require a pre-defined threshold.

V. Conclusion

This paper presents a energy efficient real time self-adaptive QR decomposition scheme for LTE-Advance standard. As

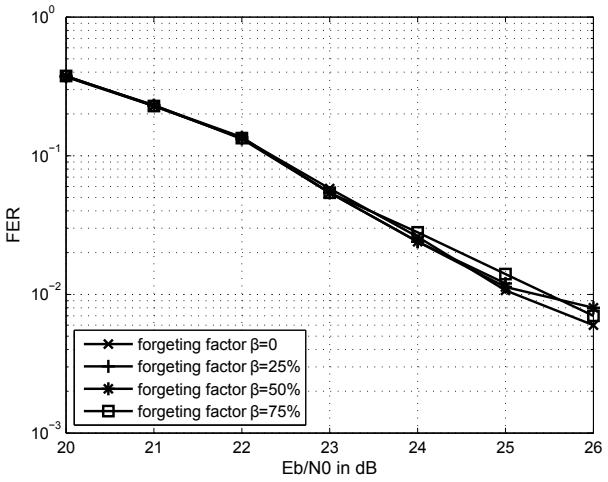


Fig. 7: Performance comparison for different forgetting factor β.

TABLE III: Amount of Complex-Value Multiplication in One Slot of a Subcarrier

	Full-H renewal	Half-H renewal	Average per QR
Pure brute force QRD	96	96	96
Hybrid algorithm*	96	78.4 **	87.2 **
Pure QR update	96	32	64
μ measurement	0	0	0
σ measurement	0	1	0.5

* 60% brute force and 40% QR update
** Expected value, including ED measurement

a case study, the proposed scheme depicts an optimization framework that enhance the performance of hybrid algorithm in a system with limited computing resource by dynamically adapting to the external environment. The simulation results have shown that the FER performance of our proposed scheme with 60% exact QRD is around 1.5 dB better for a FER of 10^{-2} in EVA-70 scenario with same resource compared with fix-threshold scheme. From the hardware aspect, we mapped our scheme on a DVFS coarse grained reconfigurable platform and maintained a stable throughput of 83.3 MQRD/s for each frame. Benefiting from the proposed scheme, DVFS is possible to be adopted in QRD computation, hence reducing the power consumption by 57.8% for pure QR update and 25.4% in case of the hybrid algorithm.

TABLE IV: Manually Mapped and Implementation Result of Targeted Schemes

	pure brute force QRD	hybrid QRD	pure QR update
$\boldsymbol{p}$	100%	60%	0%
Execution clock per QRD	6	5.45	4
Frequency requirement	500 MHz	454 MHz	333 MHz
Throughput	83.3 MQRD/s		
Supply voltage	1.1 V	1.0 V	0.9 V
Average power	190.5 mW	142.1 mW	80.4 mW
Average energy/QRD	2.29 nJ	1.71 nJ	0.97 nJ

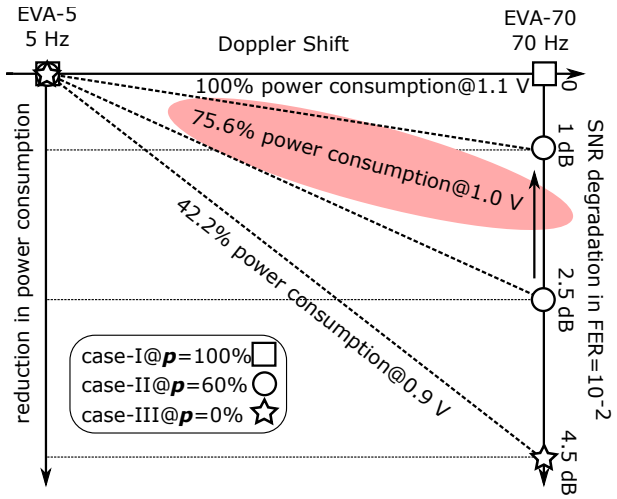


Fig. 8: Summary of relationship with power reduction and system performance.

REFERENCES

[1] Z.-Y. Huang and P.-Y. Tsai, Efficient Implementation of QR Decomposition for Gigabit MIMO-OFDM Systems, Trans. Circuits Syst. I, vol.58, no.10, pp.2531-2542, Oct. 2011.

[2] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," J. Solid-State Circuits, vol.40, no.7, pp.1566-1577, July 2005.

[3] C. Zhang, H. Prabhu, L. Liu, Ö. Edfors, and V. Öwall, "Energy efficient MIMO channel pre-processor using a low complexity on-line update scheme," NORCHIP, 2012, pp.1-4.

[4] C. Zhang, H. Prabhu, Y. Liu, L. Liu, Ö. Edfors, and V. Öwall, "Energy Efficient Group-Sort QRD Processor With On-Line Update for MIMO Channel Pre-Processing," Trans. Circuits Syst. II, vol.PP, no.99, pp.1-10.

[5] E. G. Larsson and O. Gustafsson, "The impact of dynamic voltage and frequency scaling on multicore DSP algorithm design [exploratory DSP]," Trans. Signal Process., vol. 28, no. 3, pp.127-144, May 2011.

[6] J Rabaey, Low Power Design Essentials. Integrated Circuits and Systems; Springer, 2009.

[7] H. Sakai, "Recursive least-squares algorithms of modified Gram-Schmidt type for parallel weight extraction," Trans. Signal Process., vol.42, pp.429-433, Feb 1994.

[8] S. Wang, E.E. Swartzlander, "The critically damped CORDIC algorithm for QR decomposition," Signals, Systems and Computers. vol.2, pp.908-911, Nov 1996.

[9] S.-F. Hsiao; J.-M. Delosme, "Householder CORDIC algorithms," Computers, IEEE Transactions on, vol.44, no.8, pp.990-1001, Aug 1995.

[10] D. Patel, M. Shabany, and P. Gulak, A low-complexity high-speed QR decomposition implementation for MIMO receivers, in Proc. IEEE Int. Symp. Circuits Syst., pp.33-36, May 2009.

[11] Po-L. Chiu; L.-Z. Huang; L.-W. Chai; C.-F. Liao; Y.-H. Huang, "A 684Mbps 57mW joint QR decomposition and MIMO processor for 44 MIMO-OFDM systems," Solid State Circuits Conference (A-SSCC), 2011 IEEE Asian , pp.309-312, Nov 2011.

[12] M.R. Garey, D.S. Johnson; Computers and intractability: a guide to the theory of NP-completeness W.H. Freeman and Company, San Francisco, 1979.

[13] Ajtai, Mikls, Jnos Komls, and Endre Szemerdi. "An 0 (n log n) sorting network." Proceedings of the fifteenth annual ACM symposium on Theory of computing. ACM, 1983.

[14] Robert Sedgewick, Algorithms, Addison-Wesley, ch. 8, pp. 95, 1893.

[15] L. Carlitz, "The Inverse of the Error Function." Pacific J. Math. 13, pp.459-470, 1963.