# Low Power VLSI Architecture for Adaptive MAI Suppression in CDMA Using Multi-stage Convergence Masking Vector

Yuanbin Guo, Dennis McCain
Nokia Research Center
Irving, Tx 75039
Email: Yuanbin.Guo,Dennis.McCain@nokia.com

Joseph R. Cavallaro
Dept. of Electrical and Computer Engineering
Rice University
Houston, Tx, 77005
Email: cavallar@rice.edu

***Abstract*— In this paper, we propose a novel low power and low complexity multi-stage Parallel-Residue-Compensation (PRC) architecture for enhanced MAI suppression in the CDMA systems. The accuracy of the interference cancellation is improved with a set of weights computed from an adaptive Normalized Least-Mean-Square (NLMS) algorithm. The physical meaning of the complete versus weighted interference cancellation is applied to clip the weights above a certain threshold. Multi-stage Convergence-Masking-Vector (CMV) is then proposed to combine with the clock gating as a dynamic power management scheme in the VLSI receiver architecture. This reduces the dynamic power consumption in the VLSI architecture by up to $90\%$ with a negligible performance loss.**

## I. Introduction

Parallel Interference Cancellation (PIC) [1] has been well accepted as one of the most practical algorithms for real-time implementation to suppress the Multiple-Access-Interference (MAI) in CDMA systems. A multi-stage real-time VLSI architecture based on this work is reported in [2]. However, when the interference estimation is not accurate (e.g. when the system load is high or the iteration is in early stages), cancelling the wrong estimation may even add more interference to the signal. To achieve better performance, Divsalar *et al* [3] proposed a partial cancellation algorithm by introducing a weight in each stage, with the only intuitive constraint $0 < w_1 < w_2 < \cdots < w_m < 1$, where $w_m$ is the weight at stage $m$. However, the intuitive weights are far from the optimal solution because no optimization criteria is applied in finding the weights. To achieve better performance, an adaptive PIC algorithm based on the Minimum Mean Squared Error (MMSE) criteria was proposed in [4]. However, the introduction of the NLMS algorithm increase the system complexity considerably, which makes the real-time implementation very challenging.

On the other hand, the power consumption is a critical consideration for both real-time VLSI and DSP processor designs. Average power determines the battery life while the peak power affects the reliability. Power saving can be achieved by shutting down some idle blocks in the VLSI system either by shutting off the clock or in certain cases by shutting off the power supply. Reduced computation leads to fewer instructions in DSP implementation and fewer cycles in VLSI design. The power savings achieved in this manner can be significant but are very algorithm dependent. A proper addressing of when and how to shut down or scale the voltages can result in substantial improvement in energy efficiency with no or little loss in performance.

In [5], we proposed a symbol level Parallel Residue Compensation (PRC) architecture to avoid the direct interference estimation for each individual user, which reduces the complexity of the adaptive PIC from $\mathcal{O}(K^2N)$ to $\mathcal{O}(KN)$ without performance loss. However, it is still challenging to achieve low power design architecture because the NLMS-based weight updating procedure dominates the system complexity.

In this paper, we investigate the physical meaning of the weights in the adaptive cancellation algorithm. A conventional complete PIC is considered as a special case of adaptive scheme with weight '1' for all users. In the NLMS algorithm, the user and stage specific weight is initiated to be '1'. It can be shown that when the interference weight for one particular user in one stage is '1', it implicates that the symbol of the particular user is estimated *"almost"* correctly and the interference from that user is completely cancelled. Logically, if the interference is cancelled *"correctly"* and *"completely"*, there should be no need to do further cancellation in later stages. We then investigate the inter-stage features of the user-specific weights. In the early stages, the NLMS algorithm will adjust the weights more significantly since the symbol estimation is less accurate than later stages. But the weight tends to converge to '1' in later stages as the MSE converges. After the first stage, only a small portion of weights will diverge from the initial values. Thus, the distance of one user's weight from the initial value is used as an indicator of the accuracy of the symbol estimation of that user. A Convergence-Masking-Vector (CMV) is generated by comparing each user's weight with a given threshold at each stage. The vector only contains flags ( 0 or 1 ) to indicate if the weight has converged or not.

To save the power, we apply algorithmic transformations such as pipelined and parallel processing in the VLSI architecture. The circuit is partitioned into pipelined small blocks

that have their own derived clocks. The CMV is combined with clock gating as power management scheme for the multi-stage components of the VLSI architectures. If the CMV indicates a convergence, then there is no need to update the weight for this user at all later stages and the NLMS and PRC components in later stages are shut down. The CMV can also be applied in a DSP processor implementation to dramatically reduce the number of MIPS. The dynamic power consumption is reduced based on the stochastic feature of the CMV vectors. Simulation shows that up to $40\%$ dynamic power savings in stage 1 and $90\%$ savings after stage 2 can be achieved with negligible performance loss.

## II. System Model

We consider the synchronous multi-code CDMA system using QPSK modulations scheme for the simplicity in notations. The $n^{th}$ symbol for the $k^{th}$ user at the transmitter is mapped to constellation points using a group of bits $\{b_k^0, b_k^1\} \in \{0, 1\}$. The symbol output at the modulator is $s_k^{(n)} = \{[-2b_k^0(n) + 1] + [-2b_k^1(n) + 1]j\}/\sqrt{2}$ with equal probability. In AWGN channel, the received complex base band signal at the $i^{th}$ chip of the $n^{th}$ symbol is expressed as

$$r^{(n)}(i) = \sum_{k=1}^{K} \alpha_k^{(n)} \sqrt{P_k^{(n)}} s_k^{(n)} c_k[i + (n-1)G] + z(i) \quad (1)$$

where $\alpha_k^{(n)}, P_k^{(n)}$ are the complex channel amplitude and the transmitted power for the $k^{th}$ user, respectively. $c_k[i + (n-1)G]$ is the $i^{th}$ chip spreading code of the $n^{th}$ symbol for the $k^{th}$ user and takes the value of $\{\pm 1\}$. $G$ is the spreading factor and $K \in [1, G]$ is the number of active users. $z(i)$ is the sample of the complex additive Gaussian noise with double-sided spectrum density $\mathcal{N}_0/2$.

## III. Adaptive Parallel Residue Compensation Receiver

For better accuracy of the interference estimation, a set of weights are introduced for each user in each stage. By defining a cost function in terms of the squared Euclidean distance between the received signal $r(i)$ and the weighted sum of all users' estimated signal, the optimal weights are given by minimizing the MSE of the cost function as

$$\mathbf{w}_{opt}^{(m)} = \arg\min_{w_k^{(m)}} E[|r(i) - \hat{r}_w^{(m)}(i)|^2] \quad (2)$$

where the weighted sum of all users' hard-decision symbols at the $m^{th}$ stage is given by

$$\hat{r}_w^{(m)}(i) = \sum_{k=1}^{K} w_k^{(m)} [c_k(i)\hat{s}_k^{(m-1)}] = \mathbf{w}^{(m)} \hat{\mathbf{\Omega}}^{(m-1)}(i). \quad (3)$$

Define the residual error between the desired response and its estimate in the $m^{th}$ stage as $\epsilon^{(m)}(i) = r(i) - \hat{r}_w^{(m)}(i)$, the MMSE optimization problem in (2) is solved by the Normalized Least-Mean-Square (N-LMS) algorithm in an iterative update equation operated within the bit interval at chip rate as

$$\mathbf{w}^{(m)}(i+1) = \mathbf{w}^{(m)}(i) + \frac{\mu}{\| \hat{\mathbf{\Omega}}^{(m)}(i) \|^2} [\hat{\mathbf{\Omega}}^{(m-1)}(i)]^* \epsilon^{(m)}(i)]$$
$$\mathbf{w}_{opt}^{(m)} = \mathbf{w}^{(m)}(G-1) \quad (4)$$

where $\mu$ is the step size and $\hat{\mathbf{\Omega}}^{(m-1)}$ is the input vector to the NLMS algorithm. The interference for each user in the adaptive PIC is estimated in a direct form for all the $K$ users as

$$\hat{I}_k^{(m)}(i) = \sum_{\substack{j=1 \\ j \neq k}}^{K} w_j^{(m)}(N-1)[c_j(i)\hat{s}_j^{(m-1)}] \quad (5)$$

The more accurate chip-level signal with interference cancelled is generated for each user as $\tilde{\gamma}_k^{(m)}(i) = r(i) - \hat{I}_k^{(m)}(i)$ and more accurate symbols are detected as

$$\tilde{s}_k^{(m)} = \frac{1}{G} \sum_{i=0}^{G-1} \tilde{\gamma}_k^{(m)}(i) c_k^*(i). \quad (6)$$

The complexity of direct form PIC in one chip for $K$ users is $4K(K-1)$ real multiplications, $2K(K-1)$ real additions and $2K$ subtractions. Moreover, there is one"if" statement which is mapped to a hardware comparator for each user loop. This makes the loop structure irregular and not very suitable for pipelining. Considering the regularity of the computations for all users, we change the order of "*interference estimation*" and "*interference cancellation*". Instead, the new architecture has the following steps:

1) "Weighted-Sum-Chip Function": by summing up all users' weighted signal together, we get the weighted estimation of the received chip rate samples as

$$\hat{r}_{w,opt}^{(m)}(i) = \sum_{k=1}^{K} w_k^{(m)}(G-1)[c_k(i)\hat{s}_k^{(m-1)}]. \quad (7)$$

2) "Residual Error Generation": a common residual signal for all users is generated by a single subtraction from the original signal as

$$\epsilon^{(m)}(i) = r(i) - \hat{r}_{w,opt}^{(m)}(i). \quad (8)$$

3) "Parallel Residue Computation": in the final step, this residual error is compensated to each user to get the interference-cancelled chip signal,

$$\tilde{\gamma}_k^{(m)} = \epsilon^{(m)}(i) + w_k^{(m)}(G-1)[c_k(i)\hat{s}_k^{(m-1)}]. \quad (9)$$

4) Multi-user "chip matched-filter" can be carried out on the corrected signal as in (6). The afore-mentioned procedure thus constructs a "Chip-Level" PRC (CL-PRC) structure. However, by jointly considering the matched filter and the residue compensation step in (7), (8) and (9), the $0^{th}$ stage multi-user matched filter output can be utilized to generate the "Symbol-Level" PRC (SL-PRC) architecture. The "spreading" and then

"matched filter" procedure for the weighted symbols of each user is redundant in chip level. We only need to do matched filtering for the weighted-sum chips as in (10). The soft-decision matched filter output of the corrected signal is finally generated in the symbol level as (11). The optimally weighted symbol in (7) can be computed as $w_s(k) = w_k^{(m)}(N-1)\hat{s}_k^{(m-1)}$before the spreading in (7) and stored in registers or memory arrays.

$$\hat{\xi}_{w,MF}(k) = \frac{1}{G}\sum_{i=0}^{G-1} \hat{r}_{w,opt}^{(m)}(i)c_k^*(i). \tag{10}$$

$$\tilde{s}_k^{(m)} = \tilde{\mathbf{S}}_{MF0}(k) - \hat{\xi}_{w,opt}(k) + w_s(k). \tag{11}$$

## IV. Stochastic Convergence Masking Vector

Because of the MMSE criteria in the adaptive PRC scheme, the mean squared error will converge in the NLMS update recursion. Moreover, the mean squared error becomes smaller in the later stages than the early stages because of the non-linear interference cancellation. This implicates that both the inter-stage and intra-stage convergence features of the MSE could be applied to reduce the complexity. To analyze the stochastic feature of the user-specific weights, we plot the normalized optimal weights versus chip index for stages $1, 2$ and $3$ in Fig. 1. It is observed that the weights for some symbols converge to the initial values (a normalized weight of 1), which indicates that those symbols are detected almost correctly and at this stage the interference from this user can be completely cancelled by re-generating the signal with the initial weights. Moreover, if some weight converges for some chips at the $(m-1)^{th}$ stage, the weight for this user and symbol tends to converge also at the $m^{th}$ and later stages. If the symbol is already detected correctly, then a normalized weight "1" can cancel the interference from this user "*completely*". There is no need to continue the cancellation for the particular user symbol in later stages.

With more stages, more weights converge to the normalized "1" (in the case of BPSK, most of them converge at the second stage). Thus, after more interferences are cancelled and the signal is getting cleaner in later stages, the majority of the weights will be close to the normalized weight "1". Dynamic power management then can be applied for the design of the NLMS and PRC blocks. Specifically, for the System-On-Chip (SoC) architecture with pipelined multi-stage layout processing units, we can construct a Convergence-Masking-Vector (CMV) to provide the control logics to stop the multi-stage NLMS adaptation and the PRC block. It provides a stochastic shutdown scheme to save the dynamic power by controlling the clock gating for each stage. The procedure is summarized briefly as follows.

First, we detect the $\hat{\mathbf{\Omega}}^{(0)}(i)$ from the $0^{th}$ stage matched filter. Then at the $1^{st}$ stage, we initialize $\mathbf{w}^{(1)}(0) = \hat{\alpha}$ and update the weights according to the NLMS recursion in (4). For each of the user, we set the CMV vector $V_k^{(1)}$ for $k = 1 : K$ using

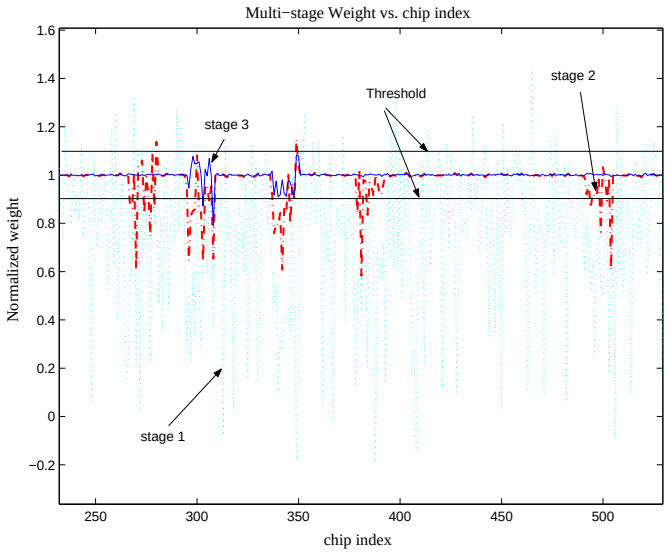


Fig. 1. Inter stage feature of the normalized weights in adaptive PRC.

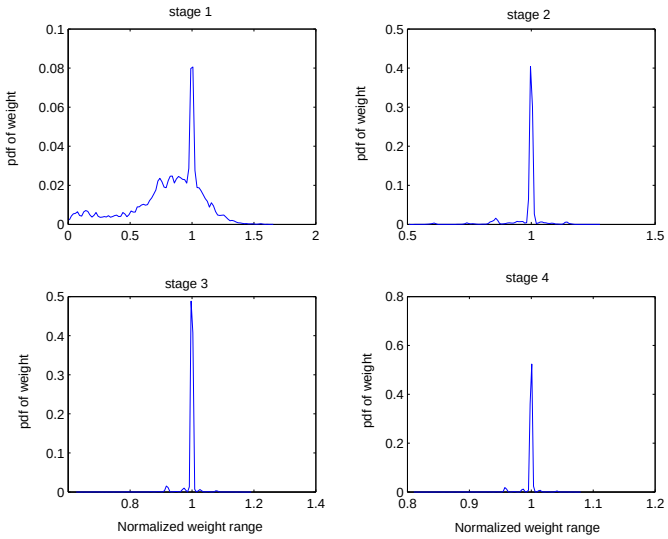


Fig. 2. The probability distribution function of the multistage weights in adaptive PRC.

a threshold $T_{sh}$ as

$$\begin{cases} V_k^{(1)} = 1 & |w_{opt,k}^{(1)} - \hat{\alpha}_k|/|\hat{\alpha}_k| < 1 - T_{sh} \\ V_k^{(1)} = 0 & o.w. \end{cases} \tag{12}$$

where $V_k^{(1)} = 1$ means that the $k^{th}$ user has converged to a "correct" symbol detection and there is no need to continue the detection of this symbol for this user in later stages. Else if $V_k^{(1)} = 0$, there is a need to continue the multistage weight update for the $k^{th}$ user. Moreover, we separate the weighted-sum chip signal in (7) into two terms: the converged term $\hat{r}_V^{(1)}$

and the not-converged term $\hat{r}_w^{(1)}$ as

$$\hat{r}_{w,opt}^{(1)} = \underbrace{\sum_{\substack{k=1,\\ V_k^{(1)}=1}}^{K} \hat{\alpha}_k[c_k(i)s_k^{(0)}]}_{\hat{r}_V^{(1)}} + \underbrace{\sum_{\substack{k=1,\\ V_k^{(1)}=0}}^{K} w_{opt,k}^{(1)}[c_k(i)s_k^{(0)}]}_{\hat{r}_w^{(1)}(i)}. \quad (13)$$

Then the symbols $\hat{\mathbf{\Omega}}^{(0)}(i)$ are detected from (10) and (11) with the interference cancelled. For the $m^{th}(m > 1)$ stage, if $V_k^{(m-1)} = 1$, then $V_k^{(m)} = 1$. There is no need to detect this symbol at later stages. Otherwise, we initialize and update the weight from the NLMS update equation (4). $w_{opt,k}^{(m)} = w_k^{(m)}(N-1)$ only for the users that have not converged, i.e., for those users whose CMV $V_k^{(m-1)} = 0$. For the newly updated weights, we compare them with the threshold again. If $|w_{opt,k}^{(1)} - \hat{\alpha}_k|/|\hat{\alpha}_k| < 1 - T_{sh}$ , it indicates the weight for the user $k$ already converges and we set $V_k^{(m)} = 1$. Again there is no need to detect symbol $\hat{s}_k^{(m)}$ at later stages and $\hat{\mathbf{\Omega}}^{(m)}(i) = \hat{\mathbf{\Omega}}^{(m-1)}(i)$. Moreover, because we separate the weighted-sum in (13) to the converged-term and not-converged term, we can add the newly converged users from the $m^{th}$ stage into the converged term in the weighted-sum chip signals in the accumulation as

$$\hat{r}_V^{(m)}(i) = \sum_{\substack{k=1,\\ V_k^{(m)}=1}}^{K} \hat{\alpha}_k[c_k(i)s_k^{(m-1)}] = \hat{r}_V^{(m-1)}(i) + \hat{r}_{V,new}^{(m-1)}(i)$$

$$\hat{r}_w^{(m)}(i) = \sum_{\substack{k=1,\\ V_k^{(m)}=0}}^{K} w_{opt,k}^{(m)}[c_k(i)s_k^{(m-1)}] \quad (14)$$

where

$$\hat{r}_{V,new}^{(m-1)}(i) = \sum_{\substack{k=1\\ V_k^{(m-1)}=0\\ V_k^{(m)}=1}}^{K} \hat{\alpha}_k[c_k(i)s_k^{(m-1)}]. \quad (15)$$

is the new term found converged at the $m^{th}$ stage. Else if $V_k^{(m)} = 0$, we need to continue the weight update and PRC computation for later stages.

$$\hat{\gamma}_{W,MF}(k) = \frac{1}{N}\sum_{i=0}^{N-1} \hat{r}_{w,opt}^{(m)}(i)c_k^*(i) \quad (16)$$

$$\hat{s}_k^{(m)} = \mathrm{sgn}\big(\tilde{\mathbf{S}}_{MF0}(k) - \hat{\gamma}_{w,MF}(k) + w_s(k)\big). \quad (17)$$

## V. Low Power VLSI with Clock Gating Dynamic Power Management

Low power design is an important factor in lowering system cost. The source of power consumptions in CMOS technology includes the switching current (dynamic power), short circuit current and leakage currents. The average power consumption of a CMOS gate due to the switching component is given by

$$P = \alpha C_L \cdot V_{dd}^2 \cdot f \quad (18)$$

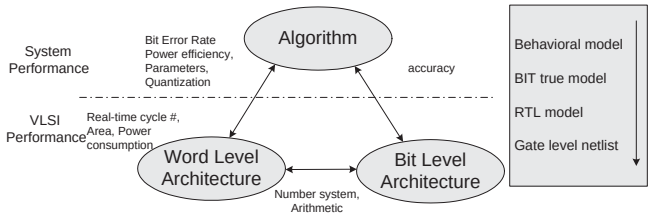


Fig. 3. Working space from algorithm to VLSI architecture.

where $f$ is the system clock frequency, $V_{dd}$ is the supply voltage, $C_L$ is the load capacitance and $\alpha$ is the switching activity (the probability of $0 \rightarrow 1$ transition during a clock-cycle), respectively. The above equation suggests many strategies for increasing the energy efficiency at various abstraction levels: from the algorithmic level down to the layout level. Algorithmic transformations like pipelining and parallel processing can be used to reduce power consumption by operating the system with lower supply voltage.

The working space from designing an algorithm to architecture is depicted in Fig. 3. On the algorithm side, the focus is the system performance in terms of bit error rate and power efficiency etc. On the architecture side, the focus is the VLSI performance in terms of the real-time cycle number, clock rate, silicon area and the dynamic power dissipation. To convert an algorithm to the real-time architecture, we work at different levels: from floating point algorithm to the behavioral model and the bit-true model, then to the RTL model and the gate level netlist. To achieve this, we need to specify the system parameters and quantize the floating-point algorithm to the fixed-point word level architecture and then to the bit-level architecture using the numerical arithmetic. The purpose is to keep the accuracy of the system performance while achieving the best VLSI efficiency.

The top-level block diagram of the multi-stage adaptive PRC architecture using the CMV to do the power saving is shown in Fig. 4. A PN generator generates the spreading codes either from a ROM block or from a simple combinational logic. The channel estimator takes the input samples and the pilot symbols to estimate the channel coefficients and feed them to the multi-stage APRC Processing Elements (PE). In each APRC stage, there is a NLMS block to update the weight and the PRC module to do the actual interference cancellation based on the optimal weights from the NLMS block. A CMV block will take the output of the weight update and set the values for the convergence masking vector $V_k^{(m)}$. The CMV is sent to the dynamic power management module to generate the control logic for the power saving of each stage. A pipeline controller also generates the control logics for the multi-stage pipelined processing units to reduce the processing latency.

The power management is also responsible for the generation of the clocks, which are supplied to the rest of the design. Clock gating is a commonly used technique to reduce dynamic power dissipation by gating off clock signals to registers, latches and clock regenerators. An example logic is shown in

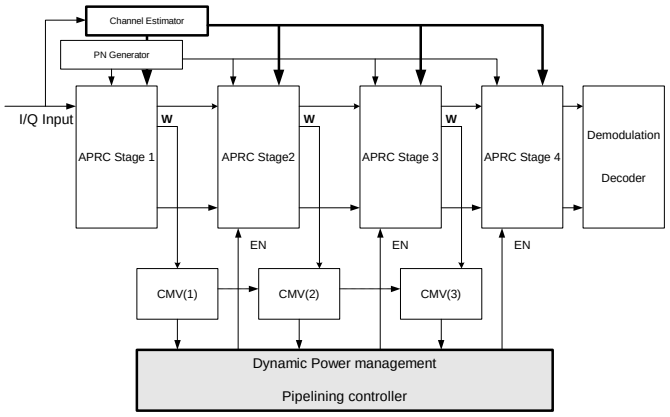


Fig. 4. The block diagram of pipelined APRC architecture with the dynamic power management.

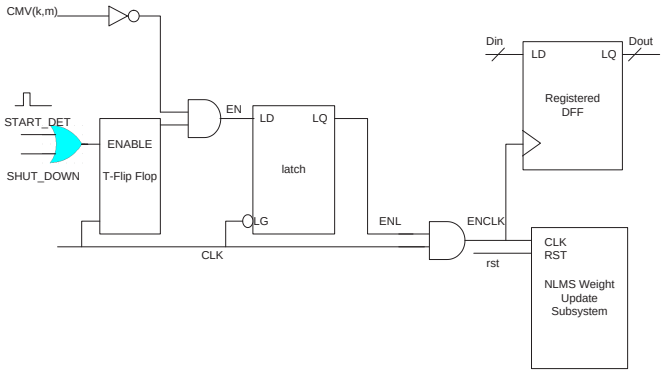


Fig. 5. The clock gating dynamic power management using Convergence-Masking-Vector.

Fig. 5. Gating may be done when there is no required activity to be performed by logic whose inputs are driven from a set of storage elements. Since output values from the logic will be ignored, the storage elements feeding the logic can be blocked from updating to prevent irrelevant switching activity in the logic. The "START_DET" and "SHUT_DOWN" signals are designed in a pattern to serve as the pulse into the T-flip flop to generate an enable signal output. This enable signal is "AND" with the inverse of the CMV for the $k^{th}$ user at the $m^{th}$ stage to generate an enable signal for the clock. It is worth noting that, in order to prevent glitches on the clock network, we are introducing a latch for each enable signals, which contributes an overhead in energy consumption. However, this overhead is negligible compared with the overall system complexity.

Consequently, in order to reduce the energy dissipation, a simple circuit detects the occurrence of convergence events for each user at each stage. It also detects the convergence events of earlier stages. When this happens, the clocks to the NLMS and the PRC modules are blocked, no further weight calculations are performed. The event transition of the CMV can be implemented by simple Finite-State-Machine(FSM).

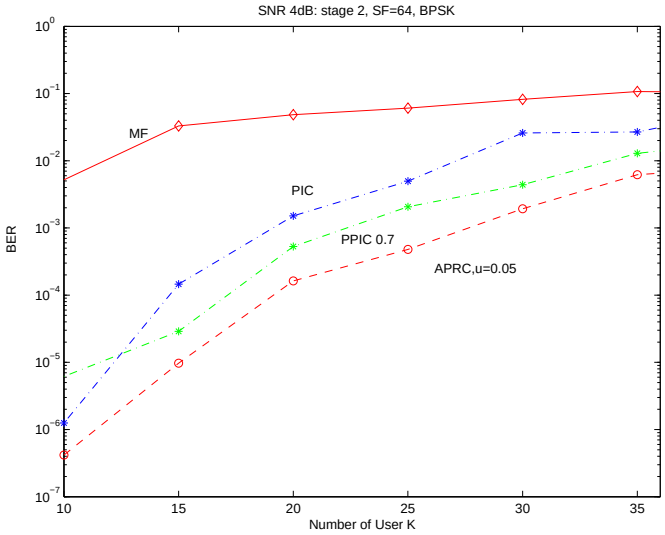


Fig. 6. Floating point Bit Error Rate versus user number K: stage 2.

## VI. BER Performance and Complexity Tradeoff

In Fig. 6, the BER performance versus the number of users for a fixed Signal-to-Noise Ratio (SNR) of 4dB is shown for Matched Filter (MF), conventional PIC, Partial-PIC (PPIC) and the Adaptive PRC (APRC) for stage 2 using random codes. The Spreading Factor (SF) is 64. For the PPIC case, a set of intuitive weights that satisfy $w_{m-1} < w_m$ are simulated. In the APRC algorithm, the step size $\mu$ controls the convergence speed of the weights. Since in the later stages, the APRC converges to a complete PIC, we choose smaller step size for fine adjustment of the weights. Because the interference estimation could be accurate in the low system load case, the PPIC with fixed weight 0.3 with 2 stages even converges slower than the PIC. It demonstrates that the intuitive weights in PPIC affect the performance dramatically. At stage 2, the PPIC starts to outperform the complete PIC after the number of users increases above 12. For the case of APRC, the performance outperforms both the PIC and PPIC significantly at both the $1^{st}$ and $2^{nd}$ stages. It can be concluded that when the system load is low, the complete PIC works fairly well. However, when the system load is high, PPIC starts to outperform complete PIC. On the other hand, by using the NLMS algorithm to compute the optimal weights for each stage, the adaptive PRC outperforms both the PIC and PPIC in a wide range of the system load. This demonstrates the superior performance of the adaptive PRC algorithm.

The BER performance of the power saving scheme with the CMV is compared with the original APRC algorithm in Fig. 7 for different relative thresholds. For a 10-user system, when the threshold is set to be 50% at stage 1, 70% at stage 2 and 90% at stage 3 and stage 4, the performance drop is negligible. However, if the threshold is below certain level, e.g., 80% at stage 4 or 40% at stage 2, significant performance degradation is observed compared with the original PRC scheme. For a 14-user system, the performance degradation is less sensitive

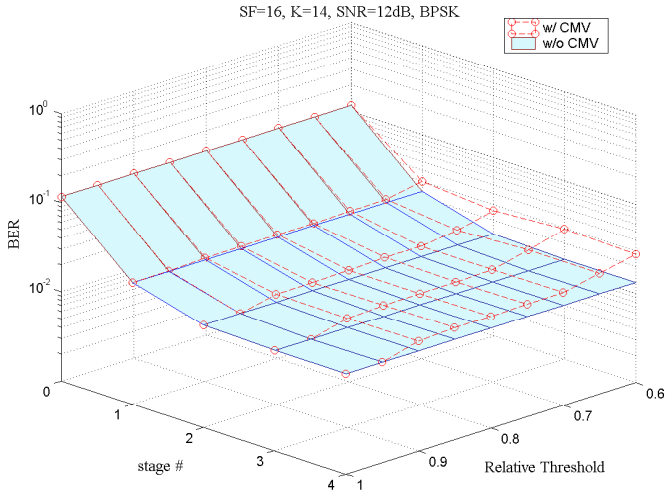


Fig. 7. The BER performance versus CMV threshold in multiple stages.

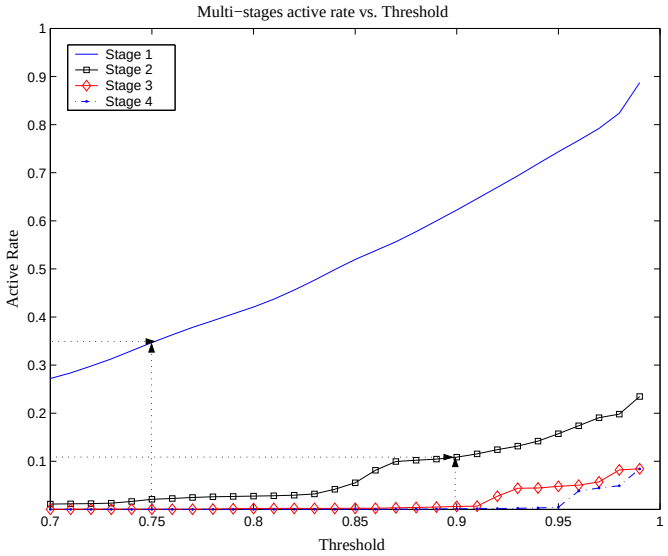


Fig. 8. The active rate of the NLMS update and residue compensation operation versus CMV threshold.

to the threshold level because the BER floor of the 14-user system is higher than the 10-user system. If the threshold is set to 95%, the BER is almost the same as the "always-update" adaptive PRC scheme.

## VII. IC Stage Active Rate

The active rate of one component $\zeta_{act} = \tau_{act}/T_{all} * 100\%$ is the percentage of time when the component is not shut down with the clock gating. Thus, the active rate is an indicator of the power saving for the pipelined VLSI architecture. In Fig. 8, we demonstrate the active rate of each stage under a different threshold level. The solid curve is the active rate of stage 1, and the solid curve with square is stage 2, the diamond curve is stage 3 and the dotted curve is stage 4. From the simulation results in Fig. 7, it is demonstrated that different threshold could be applied to different stages. If we choose a threshold of 75% for stage 1 and 90% for stage 2, 95% for stage 3 and 4, it leads to 35% active rate for stage 1, 10% active rate for stage 2 and roughly 5% for stage 3. For stage 4, only if the threshold is set to above 96% will the active rate increase to 5%. The low active rate indicates that a large power saving can be achieved over the original design with little or no loss in system performance.

Note that in a DSP implementation, the active rate also indicates the reduction in computation complexity and the number of instructions. Combining with the standby mode, this also leads to power saving in the DSP processor implementation. One difference between the DSP implementation and the pipelined multi-stage VLSI architecture is that the VLSI architecture can achieve fixed latency because of the parallel processing of multiple users, while the DSP implementation may generate variable latency depending on the stochastic feature of convergence masking vector. However, the reduction in the number of instructions indicates a $1/(1-\zeta_{act})$ speedup in obtaining the detected symbols.

## VIII. Conclusion

In this paper, we proposed a novel low power multi-stage PRC architecture for enhanced MAI suppression in CDMA systems. The CMV is combined with clock gating as a power management scheme for the multi-stage weight updating and residue compensation components. The dynamic power consumption in the resulting VLSI architecture is reduced by 90% after stage 2 with negligible performance loss.

## References


[1] M. K. Varanasi, B. Aazhang, *Multistage detection in asynchronous code-division multiple-access communications*, IEEE Trans. Commun, vol. 38, pp. 509-519, Apr. 1990.

[2] G. Xu, J. R. Cavallaro, *Real-time implementation of multistage algorithm for next generation wideband CDMA systems*, Proc. ASPA, IX, SPIE, vol. 3807, Denver, CO, pp. 62-73, July 1999.

[3] D. Divsalar, M. K. Simon, D. Raphaeli, *Improved parallel interference cancellation for CDMA*, IEEE Trans. Comm. vol. 46. No. 2, pp.258-268, Feb. 1998.

[4] G. Xue, J. Weng, T. L. Ngoc and S. Tahar, *Adaptive multistage parallel interference cancellation for CDMA*, IEEE JSAC, vol. 17, pp.1815-1827, Oct.1999.

[5] Y. Guo, D. McCain, J. R. Cavalaro, *Low complexity System-On-Chip VLSI architectures of optimal Parallel-Residue- Compensation for MAI suppression in CDMA systems*, in Proceedings of the 2004 International Symposium on Circuit and Systems, IEEE ISCAS04, pp. IV-77-80 vol.4, Vancouver, Canada, May 23-26, 2004.