PROCESSOR ARCHITECTURE FOR SOFTWARE IMPLEMENTATION OF MULTI-SECTOR G-RAKE RECEIVERS FOR HSUPA WIRELESS INFRASTRUCTURE

D. Sreedhar[1], *J. H. Derby*[1], *A. J. Vega*[1], *B. Rogers*[2], *C. L. Johnson*[1], *R. K. Montoye*[1]

[1]IBM Research Division, [2]IBM Systems and Technology Group

email : dhsreedh@in.ibm.com, {jhderby, ajvega, bmrogers, charliej, montoye}@us.ibm.com

ABSTRACT

The high speed uplink packet access (HSUPA) wireless standard requires extremely high-performance signal processing in the baseband receiver, the most challenging being the chip rate rake receiver. In this paper we describe the architectural enhancements on the IBM's PowerEN processor, to enable it to support the computational requirements of the rake receiver in a fully programmable and scalable fashion. A key feature of these enhancements is a bank-based very-large register file, with embedded single instruction multiple data (SIMD) support. This processor-in-regfile (PIR) strategy is implemented as local computation elements (LCEs) attached to each bank. This overcomes the limitation on the number of register file ports and at the same time enables high degree of parallelism. We show that these enhancements enable the integration of multi-sector HSUPA G-RAKE receivers on a single processor.

Index Terms— SDR, Vector Processors, SIMD, WCDMA, HSUPA

1. INTRODUCTION

There has been a continuous and increasing interest for a software-defined radio (SDR) where the layer 1 functions of basestation stack runs on a general-purpose processor platform. This would enable the signal processing as well as higher-layer functions to be run on a tightly integrated computing platform with a single architecture, a single tool set, and a single programming model. The evolving HSUPA mode of wideband code division multiple access (WCDMA) family of wireless standards requires extremely high-performance signal processing in the baseband receiver. In most of the current basestation platforms, these requirements are met using FPGA and ASIC technology. There is also a need to consolidate the processing of multiple sectors of a wireless infrastructure on a single processor which can reduce the cost at the same time enabling implementation of complex interference suppressing algorithms with ease. This paper describes such a general-purpose processing platform, which enables multiple sector HSUPA base stations to be implemented on a single processor.The starting point for this architecture is IBM's PowerEN processor [1],[2], a multicore, massively multithreaded platform that employs general-purpose Power processor cores and includes extensions that address functions appropriate for wired network-edge applications.

Generalized-RAKE (G-RAKE) receivers have been proven to effectively suppress multi-user interference [3]. G-RAKE receivers extends the conventional RAKE receiver by placing additional "noise" fingers along with the usual "signal" fingers to suppress interference. In this paper we show that the proposed processor can handle the work load of G-RAKE receivers implemented across multiple sectors. With the integration of multiple sector processing on a single processor, the implementation of "soft hand-off" in WCDMA systems, which typically requires communication across platforms, gets subsumed in the finger selection of G-RAKE receiver.

Section 2 puts the current paper in perspective with the prior work on this topic. Section 3 gives an overview of a multi-sector HSUPA work load. Section 4 gives an overview of the PowerEN architecture and describes the proposed architectural enhancements. We present the performance analysis of the proposed processor in Section 5. Our conclusions are given in Section 6.

2. RELATION TO PRIOR WORK

The main contributions of this paper in relation to the prior work in this area are as follows:

(i) The work presented here extends the VBA based architecture presented in [4, 5] where the focus was on FFT and Turbo decoding in LTE/LTE-A base stations. In this paper we focus on architectural enhancements for the chip rate processing in HSUPA base stations.

(ii) In [6, 7], the authors presented a software based solution for chip rate processing in WCDMA. In this paper, we have been able to increase the processing capability by 12x compared to that in [6] with a small increase in complexity/area of the silicon. This enables integration of multiple sector processing on a single silicon.

(iii) Solutions in [8, 9] use bus attached hardware acceleration for chip rate processing. Our software based-architecture results in considerable internal bus-bandwidth requirement in addition to the general benefits like ease of programming and maintenance.

(iv) G-RAKE receivers and its applications in WCDMA infrastructure is widely studied [3, 10, 11] . In this paper we borrow the algorithms presented in these papers and show the feasibility as well as advantages of multi-sector HSUPA processing on our proposed architecture.

3. MULTI-SECTOR HSUPA

Soft and softer handover (SoHo) are important features in WCDMA and HSUPA wireless infrastructure which can exploit the macro diversity thereby improving the overall spectral efficiency. With SoHo, a user can connect to multiple sectors between basestation sites. In [11], the authors present a unified view where SoHo is considered as a special case of a more general uplink co-ordinated multi-point (UL-CoMP) reception. Fig. 1 shows a multi-sector HSUPA uplink. With UL-CoMP, the SoHo can be replaced by a G-RAKE receiver which can combine over a subset of antenna streams.

3.1. G-RAKE Receivers in Multi Sector HSUPA Uplink

Let $y^{(r)}(n)$ denote the nth complex sample of the rth received stream, $r \in \{1, .., SR\}$,where S is the no. of sectors and R is the

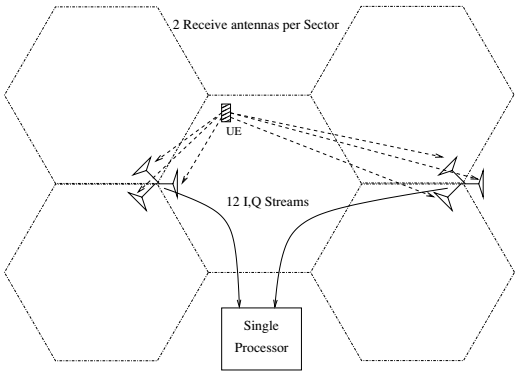


Fig. 1. Configuration where a user equipments (UE) uplink signal is received in 6 sectors and processed on a single processor.

no. of receive antennas per sector. Let $s_k^{(r)}(l,m)$ denote the mth despread symbol of the kth user at the lth delay of the rth stream. We assume $L_k^{(r)}$ delays at locations $\delta_k^{(r)}(1),...,\delta_k^{(r)}(L_k^{(r)})$ for the kth user on the rth received stream.

A generalized despreading operation can be expressed as

$$s_k^{(r)}(l,m) = \sum_{i=0}^{f-1} y^{(r)}((m-1)f + \delta_k^{(r)}(l) + i)c_k(i)\psi(\lfloor i/W \rfloor) \quad (1)$$

where $c_k(.)$ denotes the combination of the kth users known complex scrambling code and channelization code, which is of the form $\pm 1 \pm \sqrt{-1}$ and f denotes the spreading factor. $\psi(.)$ is a term for correcting carrier frequency offset (CFO) which is assumed to be constant over W chips. Let $\mathbf{s}_k(m)$ denote the $SRL_k^{(r)} \times 1$ vector $[s_k^{(1)}(1,m),...,s_k^{(1)}(L_k^{(r)},m),...,s_k^{(SR)}(1,m),...,$
$s_k^{(SR)}(L_k^{(r)},m)]$ formed by stacking all despread fingers and let $x_k(m)$ denote the symbol transmitted. Then we can write

$$\mathbf{s}_k(m) = \mathbf{h}_k x_k(m) + \mathbf{u}_k(m) \quad (2)$$

where $\mathbf{h}_k$ represents the combined effect of the transmit filter, radio channel and receive filter for the kth user and $\mathbf{u}_k(m)$ is an impairment comprised of interference and white noise contributions. The G-RAKE combining is $\mathbf{w}_k^H \mathbf{s}_k(m)$ where the combining weights are given by

$$\mathbf{w}_k = \mathbf{R}_k^{-1}\mathbf{h}_k \quad (3)$$

where $\mathbf{R}_k = E\{\mathbf{u}_k(m)\mathbf{u}_k^H(m)\}$. In conventional RAKE receiver, the number of fingers equals the no. of multipaths and the combining weights are chosen to be the path gains. But in G-RAKE, the finger positions and the gains on each finger are chosen to minimize the net interference.

G-RAKE processing chain is shown in Fig. 2. The parameters for the G-RAKE are estimated from the pilot symbols in dedicated physical control channel (DPCCH). On each received stream, for each user, the DPCCH is despread for $M_k^{(r)}$ delays for n_P pilot symbols. The despread symbols are averaged over all n_P pilot symbols and searched for the "signal" taps based on the signal energy. The final $L_k^{(r)}, L_k^{(r)} < M_k^{(r)}$ taps are selected by searching in the neighborhood of the "signal" taps for taps maximizing SNR [10]. We choose the algorithms in [10] for our analysis and the readers are requested to refer the same for details on computation of $L_k^{(r)}, \delta_k^{(r)}(1),...,\delta_k^{(r)}(L_k^{(r)})$,$\mathbf{h}_k$ and $\mathbf{R}_k^{(1)},...,\mathbf{R}_k^{(SR)}$. The data channels, dedicated physical data channel (DPDCH) and enhanced physical dedicated physical control channel (E-DPDCH) are despread only on the delays and streams selected from the finger allocation process. The data channels are combined using the weights $\mathbf{w}_k$.

The most compute intensive part is the despreading operation in Eqn. (1) which needs to be done $3n_P M_k^{(r)} SR$ times per user in a 2ms TTI. It can also be observed that there is considerable parallelism in the computations which can be exploited by the processor architecture which we will illustrate in the next section.

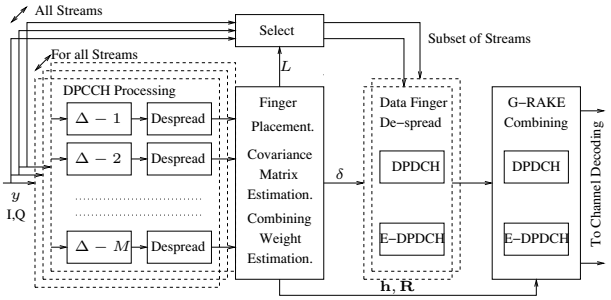


Fig. 2. HSUPA G-RAKE Processing Chain

4. POWER-EN$^{(TM)}$ AND VECTOR-BASED ACCELERATION

The PowerEN processor is a multicore, massively multithreaded platform that integrates 16 64-bit Power processor cores, identified as A2 cores. In the proposed enhancement, we attach a vector based acceleration (VBA) execution unit to an A2 core. It takes and executes instructions fetched and passed to it by the A2 core. VBA is derived from VMX, the Power SIMD architecture [12], but incorporates several innovations like the vector unit with an extremely large register file, the vector string register file (VSRF), and a processor in register (PIR) file strategy. The VSRF is managed with an indirection mechanism for dynamically addressing data contained in the register file. These features are discussed in sub-section 4.1 and the PIR mechanism for despreading is described in 4.2.

4.1. The VSRF with Indirect Access

The VSRF consists of 2K 256-bit registers, providing an aggregate 64KB of storage. It is physically arranged as a set of eight subarrays, each containing 256 registers, and each with four independent read ports and one write port. Access to data in the VSRF is via an indirection mechanism, which uses a special set of 32 map registers (MRs) that contain addresses that are offsets from the VSRF origin. MRs are 128 bits wide and support sub-word parallelism. The contents of the map registers are managed by software in SIMD fashion using a set of new "*map management*" instructions; which include arithmetic operations on the entries in an MR. The indirection mechanism has two basic forms, referred to as "*Operand-associated indirection*" and "*Generalized indirection*". *Operand-associated indirection* enables the specification of one out of 2K registers in a 5-bit register operand field. A 5-bit operand selects one 16-bit pointer in an MR, and the pointer indicates the VSRF register accessed. Since there are eight 16-bit pointers in one MR, a 5-bit operand indexes

four MRs. Four MRs are therefore logically grouped into an operand map. There are four operand maps, one for each of the four operand positions in a VBA instruction (three vector sources and one destination). Defining separate maps for inputs and outputs allows register specifiers to be reused in different contexts, easing register specifier selection. *Generalized indirection* permits access to up to eight data elements at arbitrary locations in the VSRF with a single instruction. The gather instructions will place the addressed data elements in a specified order in a target register in the VSRF. For example, a gather words instruction, **vgetw VT,MA,MB**, will take eight pointer values from map register MA and interpret the eight values in map register MB as lengths (in bits), extract eight data elements with the specified lengths from the specified locations in the VSRF, and place them in 32-bit slots in target register VT in the VSRF.

The VSRF differs from a typical L1 data cache in the following (a) Its access latencies completely hidden by pipe-lining and bypassing. (b) The VSRF provides fixed access latency for reading or writing a register which is very critical for real time applications. These features contribute to significant cycle saving in loading/storing vector registers.

4.2. PIR Strategy for De-spreading

Exploiting the large amount of parallelism would require moving a lot of data from the register file to the computation resources and vice versa. To alleviate the pressure on the register file interface, part of the computation resources is embedded into the register file. We denote this strategy as processor-in-regfile (PIR). PIR is intended to exploit local computation in each bank as much as possible. Taking advantage of the available parallelism, an application is partitioned into smaller parallel problems, whose working sets fit in each bank. In this way, the pressure on the register file interface (read/write ports) is significantly reduced. The embedded logic, referred to as local computation elements (LCEs), is attached to each bank and provides SIMD support.

The despreading operation is carried out by a SIMD instruction that implements "correlation with a bit vector" operations. This is a multiply-accumulate instruction where the multiplication is by elements of a sequence that are represented by elements of a bit vector which represents $\pm 1 \pm \sqrt{-1}$ and hence there is only addition and subtraction involved in this operation. Each instruction has: (a) an input operand containing a vector of input values; (b) an input/output operand containing a vector of values usually representing partial accumulations of correlation values; (c) an input operand containing the vector of bit values to be used by the instruction; All operands are vector registers. The key novel element of this instruction compared to the DESPREAD in [7] is that there is sub-word parallelism the vector operation which implies that for a 256 bit vector containing 16 chips, (16bits for I&Q) 8 symbols can be despread in parallel (2 chips are combined into single lane to account for the increase in precision). Unlike in [7], this sub-word parallelism allows the complex multiply for CFO correction to be applied for any value of W and also despreading can be done independent of the spread factor.

The LCE operation is built over the despreading and is illustrated in Fig. 3. Each instruction would trigger 8 simultaneous SIMD despread operations on the VSRF. Each vector operation consumes 16 I-Q chip values (assuming 16 bits for I&Q) and hence 128 chip values are despread in one instruction cycle. Assuming a clock of 2.3 GHz, this amounts to 295G chip correlations in one sec. The gather instructions described in sub-section 4.1 helps to organize the data as required for the vector operations. There are many ways in which the parallelism across LCE's can be exploited. For e.g., the DPCCH processing in Fig. 2 involves $M_k^{(r)}$ parallel despread operations which could be done in parallel across LCE's.

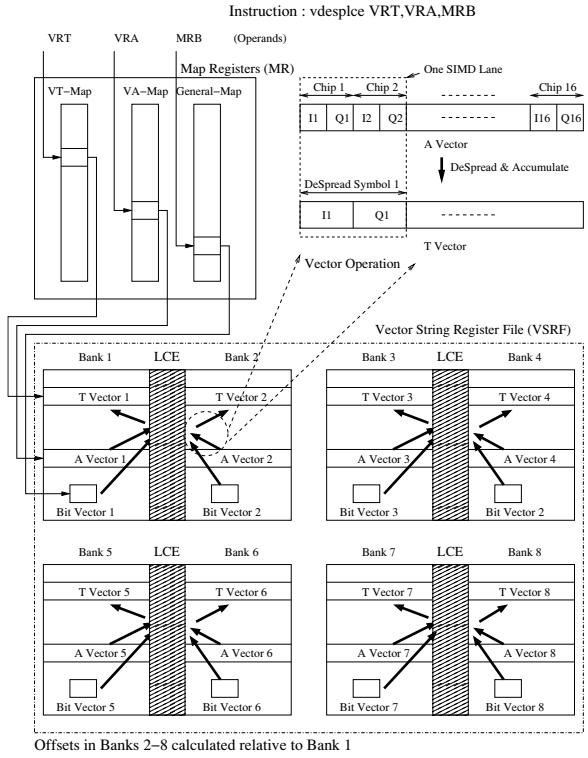


Fig. 3. LCE Despreading.

4.3. Strategy for Complex Multiply and Matrix Inverse

Complex multiply is required for the CFO correction in (1) and for the matrix inversion in (3). The matrix inversion in (3) can be approximated by Gauss-siedel iterations as

$$\mathbf{D}_k \mathbf{w}_k^{n+1} = \mathbf{h}_k - \mathbf{L}_k^H \mathbf{w}_k^n - \mathbf{L}_k \mathbf{w}_k^n \tag{4}$$

where $\mathbf{w}_k^n$ is the weights after nth iteration and $\mathbf{D}_k$,$\mathbf{L}_k$ are the main diagonal and lower triangular parts of $\mathbf{R}_k$ respectively. Under the assumption that the interference from multiple sectors and receive antennas are uncorrelated we can write $\mathbf{R}_k = diag[\mathbf{R}_k^{(1)}, ..., \mathbf{R}_k^{(SR)}]$. So the iterations in (4) requires vectors of dimension $L_k^{(r)}$ only. Assuming a maximum of 8 fingers per user (each finger has 32 bis for I & Q), it is easy to see that the iterations in (4) can be done using vector complex multiply instructions. It is possible to have a PIR strategy similar to despreading for complex multiply as well. This would enable independent instances of despreading and the iterations in (4) to happen in parallel across the different register banks.

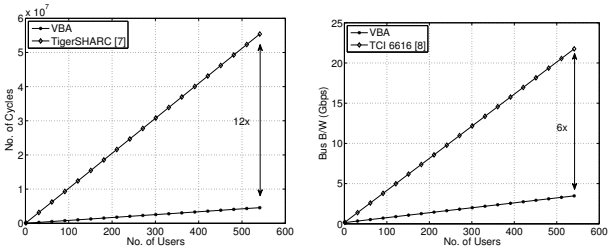
5. PERFORMANCE RESULTS

Performance analysis is carried out using functional and performance models of the A2 core with VBA. The functional model is used to verify its correctness of the code as well as to generate instruction traces. The instruction traces are passed through the performance model which gives the cycle-accurate estimates.

Assuming on an average 6 I,Q streams are combined per user, we estimate $343.2k$ cycles to process a 10ms TTI of one voice user within the latency constraints. Assuming 6 sectors with a total of 550 voice users and the processor clocked at 2.3GHz, It would take 9 VBA core threads to process this load. With 16 A2 and VBA's, this is 30% of the total compute power of the processor.

(i) Performance Comparisons. In Fig. 4(a), we compare the performance of VBA with the TigerSHARC processor [7]. We consider the DPCCH processing in Fig. 2. For each user we consider 4 I,Q streams sampled at twice the chip rate and search for 64 delays ($M_k^{(r)} = 64, \forall r, k$). We plot the total cycles required against the no. of users in Fig. 4(a). The cycles required for TigerSHARC is almost 12X times that of VBA. The TCI6616 [8] uses 2 receive accelerate co-processors (RAC) to do the chip-rate processing. Each RAC can perform 6144 chip-rate correlations simultaneously, whereas a single VBA can do 32k. With 16 VBA's in one processor, the net chip correlation power is 40X times that of TCI6616.

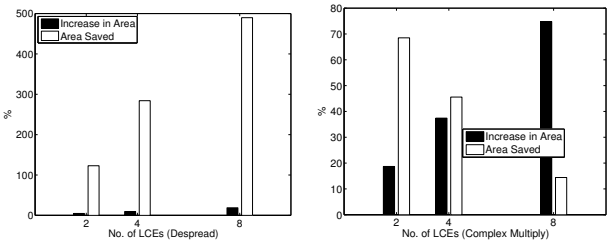
(ii) Internal Bus Bandwidth Savings. In Fig 4(b), we compare the total internal bus bandwidth for VBA with that of TCI6616 [8]. In TCI6616, the bus is used to feed the I,Q samples to RAC and get the correlation values back, whereas in VBA it is required only to transport the I,Q samples to the L2 memory of multiple cores. We use the same configurations as in the previous example. There is a 6X times reduction in the bus bandwidth utilization due to the *in-line* programming in VBA. The reduced bus bandwidth results in simpler bus design with lesser contentions and latency.



(a) (b)

Fig. 4. (a) Performance comparisons with [7]. (b) Internal Bus bandwidth comparisons with [8].

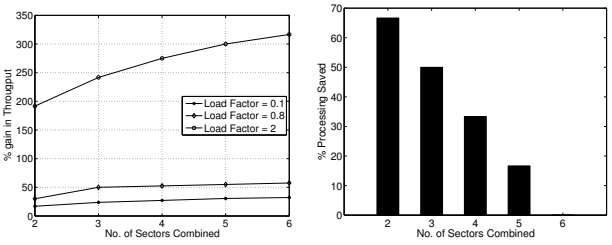
(iii) LCE Area Analysis. A large register file with an embedded LCE has an impact on the area requirement of the processor.In Figs. 5(a,b) we study the area-performance trade-off of both despread and complex multiply instruction in LCE. A single A2 core takes $2.3mm^2$ and a VBA without LCE takes $2.0mm^2$ in $22nm$ technology. For despreading we consider the DPCCH processing in the same configuration as in previous examples. To handle the workload of this module within the latency constraints, it takes 21 cores without LCE's. When we increase the no. of LCE's, there is a reduction in core count required but the size of each VBA increases. This trade-off for despread instruction is plotted in Fig. 5(a). For adding 8 LCE's, there is 19% increase in area for each core, but there is a net area saving of 480% compared to a non LCE based multi core implementation. We study the same trade off for complex multiply instruction in Fig. 5(b) using the matrix inversion workload. The reduction is not so dramatic for two reasons (a) complex multiply consumes a larger area (b) frequency of complex multiply instruction is much lesser than that of despread in the workload considered. This example also illustrates why a PIR/LCE parallel strategy is more effective than a conventional multi-core strategy.



(a) (b)

Fig. 5. (a) LCE area analysis for despreading. (b) LCE area analysis for complex multiply.

(iv) G-RAKE efficiency and Computational Processing Reduction. In Figs. 6(a,b) we study the performance improvements with multi-sector G-RAKE combining in a single processor. We make use of the configurations and throughput in [11]. The base system with which we compare is a conventional network with only soft handover. Different sectors are processed on different platforms independently. Always 3 sectors with 2 receive antennas in each sector is processed and the soft values are added. Hence, the base system has a fixed computational load. On the x-axis in Figs. 6(a,b) we have the no. of sectors combined in G-RAKE processing. In Fig. 6(a) we plot the % throughput gained per user as compared to the base system. We plot the same for different load factors, load factor = 2 means that 2 users are always active in any sector. In Fig. 6(b), we plot the computational processing reduction for the same due to consolidation and due the fact that the no. of sectors combined is essentially software controlled.



(a) (b)

Fig. 6. (a) Throughput gained by multi-sector G-RAKE combining. (b) Processing power saved by consolidating multi-sector processing on a single processor.

6. CONCLUSIONS

We have presented in this paper a potential enhanced version of the IBM PowerEN chip, which can enable integration of multiple-sector HSUPA processing on a single processor. The key new element would be the augmentation of each Power processor core with an execution unit providing vector based acceleration attached with a large register file which has high parallel processing capability. Our aggregate results point to the feasibility of an essentially general-purpose computing platform supporting SDR for multi-sector HSUPA in a fully programmable and scalable fashion at the highest levels of performance required. We also establish several advantages of consolidated processing in the enhanced PowerEN chip.

7. REFERENCES

[1] J. D. Brown *et al*, "IBM Power edge-of-network processor: A wire-speed system on a chip," *IEEE Micro*, vol. 31, pp. 76–85, March/April 2011.

[2] C. Johnson *et al*, "A wire-speed Power^{TM} processor: 2.3GHz 45nm SOI with 16 cores and 64 threads," *Proc. ISSCC, San Francisco, CA*, pp. 104–106, February 2010.

[3] G. E. Bottomley *et al*, "A generalized RAKE receiver for interference suppression," *IEEE J. Sel. Areas Commun.*, vol. 18, pp. 1536–1545, August 2000.

[4] A. Vega *et al*., "Architectural perspectives of future wireless base stations based on the IBM PowerEN processor," *IEEE 18th International Symposium on High Performance Computer Architecture (HPCA)*, pp. 1–10, 2012.

[5] J. Derby *et al*., "Vector-based acceleration in the IBM PowerEN^{TM} processor to enable software-defined radio," *Proc. SDR 11 Technical Conf. and Product Exposition, Washington, DC*, pp. 85–93, November 2011.

[6] K. Lange *et al*, "A software solution for chip rate processing in CDMA wireless infrastructure," *IEEE Communications Magazine*, February 2002.

[7] Analog Devices Corp., "ADSP-TS201 TigerSHARC^{TM} processor programming reference," *available from www.analog.com*, 2005.

[8] Texas Instruments Inc., "TMS320TCI6616 Communications Infrastructure KeyStone SoC," *available at http://www.ti.com/lit/gpn/tms320tci6616*, March 2012.

[9] Freescale Semiconductor Inc., "QorIQ qonverge B4860 baseband processor," *available at http://www.freescale.com/*, 2012.

[10] G. Kutz and A. Chass., "Low complexity implementation of a downlink CDMA generalized RAKE receiver," *IEEE Veh. Technol. Conf., Vancouver, Canada*, pp. 1357–1361, September 2002.

[11] S. Grant *et al*, "Uplink CoMP for HSPA," *IEEE Veh. Technol. Conf., (VTC Spring)*, pp. 1–5, May 2011.

[12] IBM Corporation, "PowerISA^{TM} Ver. 2.06 Rev. B.," *available from www.power.org*, 2010.