Dynamic Power Management for the Iterative Decoding of Turbo Codes

Erick Amador, Raymond Knopp, Renaud Pacalet, and Vincent Rezard

***Abstract*—Turbo codes are presently ubiquitous in the context of mobile wireless communications among other application domains. A decoder for such codes is typically the most power intensive component in the baseband processing chain of a wireless receiver. The iterative nature of these decoders represents a dynamic workload. This brief presents a dynamic power management policy for these decoders. An algorithm is proposed to tune a power manageable decoder according to a prediction of the workload involved within the decoding task. By reclaiming the timing slack left when operating the decoder at a high power mode, the proposed algorithm continuously looks for opportunities to switch to a lower power mode that guarantees the task completion. We apply this technique to an LTE Turbo decoder and explore the feasibility of a VLSI implementation on a CMOS technology of 65nm. Energy savings of up to 54% were achieved with a relatively low loss in error-correction performance.**

***Index Terms*—Turbo codes, LDPC codes, iterative decoding, low power design, dynamic power management.**

I. Introduction

Iterative decoding techniques for modern capacity-approaching codes are currently dominating the choices for forward error correction (FEC) in a plethora of applications. Turbo codes, proposed in 1993 [1], triggered the breakthrough in channel coding techniques since these codes approach the Shannon capacity limit. This was followed by the rediscovery of low-density parity-check (LDPC) codes in the 1990s, originally proposed by Gallager [2] in 1963.

Modern wireless communication standards have already adopted these types of codes for FEC and channel coding applications. For example, Turbo codes are used in the 3GPP Universal Mobile Telecommunications System (UMTS) [3] and its Long Term Evolution (LTE) [4] system.

These codes are typically decoded by an iterative message-passing algorithm. Iterations are executed until a stopping criterion is satisfied or a preset maximum number of iterations is reached. Because of the iterative nature of the decoding algorithms, *iteration control* has been a recurrent topic for reducing the power consumption of these decoders. Iteration control techniques, also known as *early stopping criteria*, attempt to reduce the operational complexity of the decoder through an early detection of codeblock convergence or lack thereof before the maximum number of iterations is reached.

In this work, we address the problem of reducing the decoder power consumption from a different perspective. Our approach is based upon the following observations:

- Practical decoders are designed and dimensioned in order to execute a maximum number of iterations so that a timing deadline is fulfilled.
- It is well-known that error-free decoding can be achieved with a few iterations under good channel conditions.
- For bad channel conditions a codeblock might not even be decoded with the maximum number of iterations.

In the following, we argue that by monitoring the dynamics of the decoding process it should be possible to control a power manageable decoder such that energy efficiency is improved. We propose a dynamic power management policy that looks for opportunities to slowdown the system in order to reclaim the timing slack due to a codeblock that converges before the task deadline. Based upon a prediction of the workload of the decoding task, we take advantage of the fact that the decoder is designed for a maximum number of iterations that should take place within a particular timing deadline. We formulate an online algorithm that adjusts the operation mode of a decoder based upon the characteristic behavior of a convergence metric. Furthermore, we explore the feasibility of a VLSI implementation for such algorithm and control law in a CMOS technology of $65nm$ for an LTE Turbo decoder. This brief builds upon our work presented in [5] where dynamic power management for LDPC decoders was presented.

The rest of the paper is organized as follows: In Section II the targeted codes are briefly introduced. In Section III, dynamic power management and the proposed control policy are explained along with the prior art. Section IV outlines the hardware implementation of the proposed control technique along with results on policy performance, policy tuning and obtained energy savings. The achieved energy savings are compared with similar works from the prior art. Section V concludes the paper.

II. Turbo Codes

Turbo codes consist of the parallel concatenation of two convolutional encoders separated by an interleaver. The decoding strategy for these codes consists of the decoding of the individual convolutional component codes and an iterative exchange of extrinsic information between the two decoders. Soft-input soft-output (SISO) decoders are used and typically execute the *maximum a posteriori* (MAP) algorithm [6] in the logarithmic domain. The general structure of a Turbo decoder is shown in Figure 1. Intrinsic messages (δ_s for systematic bits and $\delta_{p1,p2}$ for parity bits) in the form of log-likelihood ratios (LLR) are distributed in non-interleaved/interleaved form to two MAP decoders that generate and exchange extrinsic information ($\lambda^{1,2}$) in an iterative fashion. Each decoding round performed by a MAP unit constitutes a *half-iteration*. Iterations are performed until convergence is achieved or a maximum number of iterations is completed.

In the following, we argue that by monitoring the dynamics of the decoding process it should be possible to make predictions on the required decoder workload. We use the term workload to refer to either iterations or half-iterations.

Erick Amador and Raymond Knopp are with EURECOM, 2229, Route des Cretes, 06904 Sophia Antipolis, France (email: name.lastname@eurecom.fr), Renaud Pacalet is with TELECOM ParisTech, 2229, Route des Cretes, 06904 Sophia Antipolis, France (email:renaud.pacalet@telecom-paristech.fr), and Vincent Rezard is with Intel Mobile Communications, 2600 Route des Cretes, 06560 Sophia Antipolis, France (email:vincent.rezard@intel.com)

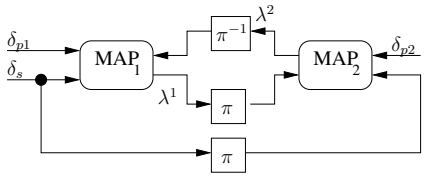


Fig. 1. General structure of a Turbo decoder.

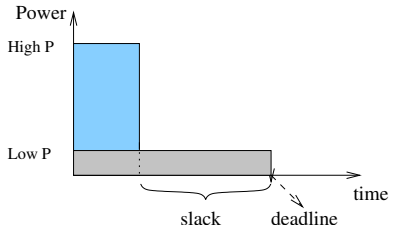


Fig. 2. Example power/slowdown scenario.

III. Dynamic Power Management

Dynamic power management (DPM) refers to a set of techniques used to achieve energy-efficient operation of a system. This is performed by judiciously adjusting or reconfiguring the system to provide a requested service and performance level with a minimum energy expenditure based upon run-time observations. Several techniques exist to achieve this at different levels such as sleep, slowdown modes and clock gating. In order to apply DPM usually two premises are considered, [7]: the system experiences a non-uniform workload during operation-time and it is possible to certain degree to predict the fluctuations of the workload. Usually, a *power manager* executes a control procedure known as a *policy* or *law* that is based upon observations of the task workload.

Iterative decoding is inherently dynamic in the sense that the number of iterations depends upon the reliability of the decoding process. This is basically determined by the level of noise present in the received codeblock. This is therefore a task with variable workload. Typically, these decoders are dimensioned to operate at a high performance mode in order to complete a maximum number of iterations within a given timing deadline. Nevertheless, this design paradigm is strictly pessimistic since a codeblock would typically reach convergence with fewer iterations than the preset maximum. This fact could be exploited in order to reclaim the timing slack and slowdown the decoder to a low-power mode.

Figure 2 shows a decoding task performed under both a high- and a low-power mode. Under the high-power mode the task is completed before a timing deadline, while under the low-power mode the task is also completed by the deadline but utilizes the full slack that remains from the high-power mode. The area under each curve indicates the total energy spent for each task. Depending upon the relationship among the power levels and the slowdown factor, energy efficiency may be improved by reclaiming the slack left when running under the high-power mode. Notice that the deadline is typically defined in order to comply with requirements like latency and/or throughput.

A. Prior Art

Previous works addressing power management on iterative decoders focus on reducing the number of iterations to avoid unnecessary decoder operation. *Iteration control* techniques attempt to detect or predict the convergence or not of a codeblock and decide whether to halt the decoding task. This decision is aided by so-called *hard* or *soft* decisions. Hard-decision aided (HDA) criteria are obtained as a function of binary-decision values from the decoding process; on the other hand the soft-decision aided (SDA) criteria use a non-binary-decision parameter from the decoding process that is compared against threshold values. For the case of Turbo decoding the authors in [8] proposed to monitor the sign changes of the LLRs in order to detect the codeblock convergence. Well-known methods for SDA criteria monitor the cross-entropy value [9] and the mean-reliability value [10].

In [11][12] the authors proposed a preprocessing stage for LDPC decoding that estimates the required decoding effort and proceed to adjust the system power mode in order to have a constant decoding-time. To the best of our knowledge, no other work within the prior art has attempted to follow online the iterative decoding process in order to make predictions about the codeblock convergence and look for opportunities to apply dynamic power management strategies.

DPM techniques based upon workload prediction have been previously proposed in different contexts. For example, the work in [13] predicts the MPEG frame decoding time and applies dynamic voltage scaling. In [14] the authors target embedded system applications and propose a software-based power manager that profiles the workload characteristics through a queuing model.

DPM has been a topic of intense research, comprehensive surveys can be found in [7][15]. As revealed by these works, DPM has been mostly investigated in the context of operating systems for general purpose and embedded computing. The main problem studied has been to find the optimal transition times to low-power or idle modes. In this work, we target a real-time kernel for mobile computing devices that must rely upon control policies of very low complexity in order to enable DPM. Following the taxonomy introduced in [7], we present an *adaptive predictive* control scheme for iterative decoders.

B. Problem Definition

We consider an iterative decoder to be a power manageable CMOS component governed by a power manager. The set $\mathcal{P} = \{P_0, P_1, \ldots, P_{n-1}\}$ defines n power modes where, [11]:

$$\begin{aligned} P_k &= P_k^{sw} + P_k^{sc} + P_k^{leak} \\ &= E_{sw} C_L V_k^2 f_k + I_{SC} V_k + I_{leak} V_k \,. \end{aligned} \quad (1)$$

P_k^{sw} is the power due to the switching activity when charging and discharging the load capacitance C_L with switching activity factor E_{sw}. P_k^{sc} is the power due to a short-circuit current when both NMOS and PMOS sections of the circuit are switched. P_k^{leak} is the power due to the leakage current I_{leak} (subthreshold plus reverse bias junction current), a technology dependent parameter. Each power mode P_k operates at a

particular voltage level V_k and frequency f_k. In the following, we assume that the first state P_0 consumes the most power, subsequent states consume each less power than the previous one. For each power mode P_k there is a corresponding slowdown factor α_k where for the fastest mode $\alpha_0 = 1$. Each power mode can also be described as a fraction of the highest power mode P_0 by a factor β_k, e.g., $P_k = \beta_k P_0$. Given the quadratic relation between power and voltage and the linear relation between power and frequency, it is possible to slowdown the system such that the total energy expenditure is reduced. This is the principle behind the well-known concept of dynamic voltage and frequency scaling (DVFS), [16].

Given the model of the power function $P_k \propto V_k^2$ and $f_k \propto (V_k - V_t^2)/V_k$, [17], it is in the best interest of the power manager to run a task as slowly as possible due to the convexity of the power function, [15][18].

Given a workload of I iterations to be executed before a timing deadline d, we wish to find a subset of power modes $\mathcal{P}' \subseteq \mathcal{P}$ such that the total energy is minimized. If an iteration is executed within a time duration t_k through a power mode P_k, the problem is stated as finding the optimal $\mathcal{P}'$ that minimizes the total energy spent:

$$minimize \sum_{i}^{I} P_k^{(i)} t_k^{(i)}, \quad P_k \in \mathcal{P}' \tag{2}$$

$$subject\ to \sum_{i}^{I} t_k^{(i)} \leq d\,,$$

where $P_k^{(i)}$ indicates the power mode used during the ith iteration. This problem can be formulated as well as a linear program that minimizes an energy function by finding a power $P \in \mathbb{R}$ that guarantees the constraint d. Such formulation, however, would not capture the adaptive online characteristic of choosing a power mode from a finite set under the uncertainty of the workload. In the following, we propose a heuristic to solve (2) based upon workload predictions.

C. Control Policy

DPM is at its core an *online* problem since a power manager must make decisions about the system operation mode before all of the input to the system is available. The input here refers to the total required number of iterations to achieve a codeblock convergence. Indeed, an online algorithm attempts to find an optimal power mode based upon information available only at run-time. On the other hand, an offline algorithm finds the optimal power mode assuming the total required number of decoding iterations is known.

In order to formulate the online policy it is necessary to look into the dynamics of the decoding process. We propose to monitor the number of hard-decision changes upon the posterior messages after each half-iteration, i.e., at the output of each component SISO decoder. We refer to this metric as a *convergence* metric and use it to make decisions in order to solve (2).

Figure 3 shows the number of sign changes in the posterior LLRs after each half-iteration for an instance of an undecodable and a decodable codeblock. This corresponds to the code

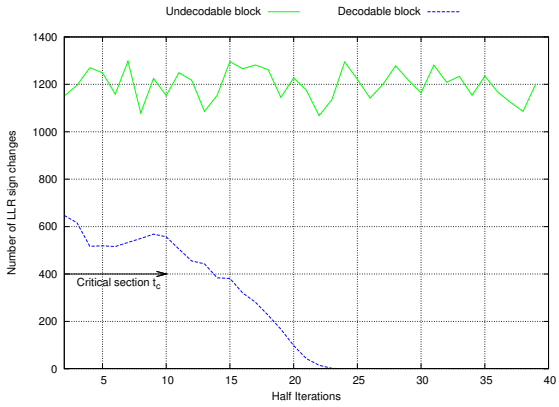


Fig. 3. Convergence metric example behavior.

defined in [4] with codeblock length 6144 and rate $2/3$ through the additive white Gaussian channel (AWGN) ($E_b/N_0 = 1dB$) with quadrature phase shift keying (QPSK) modulation and 20 maximum full-iterations. The decision metric fluctuates around a mean value for undecodable blocks, but for decodable blocks it fluctuates for a period of time t_c and later enters a *convergence* mode characterized by a monotonic decreasing behavior. We refer to the period t_c as a *critical* period since no decision can be made regarding the convergence of the code.

As shown in Figure 3, no predictions can actually be valid during the critical time section due to the repeated and irregular fluctuations of the metric. Nonetheless, once the convergence mode is entered predictions can be made at each time step in order to approximate the probable end of the task. Based upon the assumption that slowing the task speed is the optimal decision in terms of energy consumption, we propose to operate the decoder at a high-power mode (high speed) during the critical period and look for opportunities to slowdown (low-power modes) the system based upon the metric predictions.

We formulate the proposed control policy based upon the observations of the selected convergence metric. Figure 4 shows the decision flow of the control policy. Decision making is based upon the behavior of the monitored metric that reveals whether a convergence mode is entered or not and whether further decoding iterations may be triggered or not. The latter point in fact refers to an early stopping criterion just like the ones mentioned in the prior art. In Section IV-B, we will show how the performance loss in the error-correction sense is due to the stopping criteria used within the power control policy. Stopping criteria suffer from false alarms, i.e., codeblocks that would have been successfully decoded in the absence of such criterion.

IV. Hardware Implementation

In this section, we describe the proposed system for dynamic power management that implements the control policy outlined in Section III-C. Results from policy simulations along with synthesized components are presented as well.

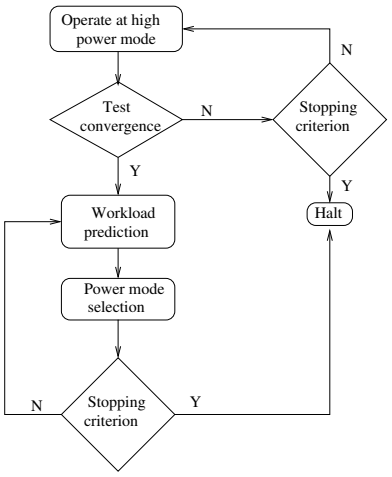


Fig. 4. Proposed DPM policy flow.

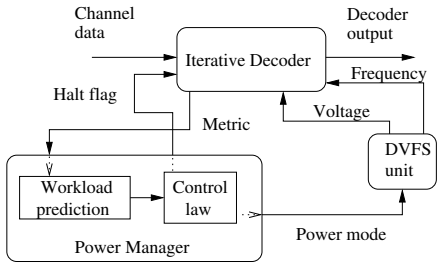


Fig. 5. System block diagram.

A. DPM System

The proposed DPM system is shown in Figure 5. An iterative decoder with adjustable voltage and frequency operation is governed by a power manager. The decoder receives intrinsic channel values in the form of LLRs and produces hard-decision bits for the decoded message. By constantly monitoring a convergence metric from the decoder, the power manager executes the proposed control policy. The power manager sets the state of a DVFS unit that provides the operating conditions for the decoder.

The energy savings obtained by the proposed DPM policy depend upon the characteristics of the power modes used and the implementation of the DVFS block. There are numerous works on how to implement a DVFS unit, using different techniques where several tradeoffs take place: size and power overhead, mode switching speed and conversion efficiency. The work in [19] provides a study on on-chip regulators for DVFS implementation on a dedicated core. This and similar work in [20] show sufficiently fast switching regulators (voltage transition times on the order of tens of nanoseconds) for demanding applications such as Turbo decoding. Based upon [19][20], we target an on-chip solution to implement the DVFS unit.

B. Results

The tuning of the control policy refers to the setting of the parameters within the stopping criteria used. This is done in conjunction with the workload characterization, which refers to observations from the average number of iterations as a function of SNR. This provides insights into the required workload based upon the channel quality. By observing the average number of critical iterations the stopping criterion is adjusted to limit the time the decoder will operate at a high-power mode.

TABLE I
AREA AND POWER COMPARISON

Component	Area $[mm^2]$	Power $[mW]$
Turbo decoder (No DPM)	0.62	180
Power manager	0.08	5
DVFS unit	0.12	25

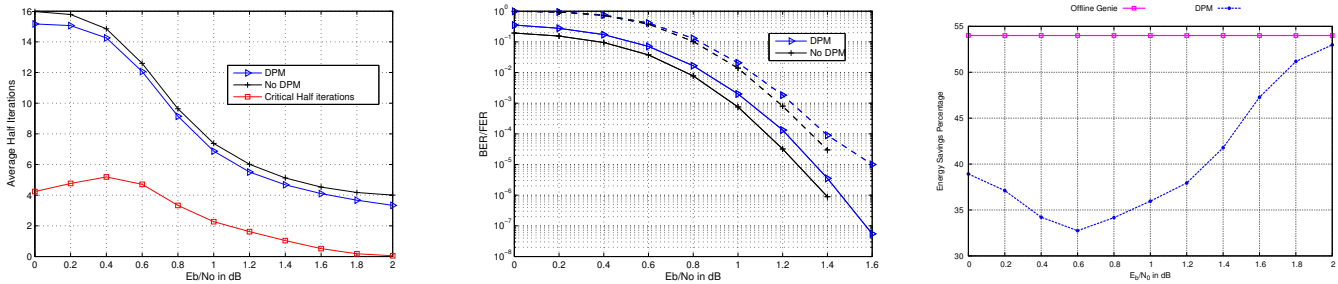
We applied the proposed DPM technique to an LTE Turbo decoder that uses 6 SISO radix-2 units with a window length of 32 samples with a message quantization of 6-bits. It provides a throughput of 95Mbps and completes a decoding task of 8 full iterations in $65\mu s$ (this value is used as the timing constraint). The decoder operates on two modes: 1.2V at 266MHz and 0.9V at 160MHz. The decoder and the power manager were synthesized with Synopsys Design Compiler and the power consumption was estimated with Synopsys PrimeTime using postlayout netlists along with pertinent activity files.

Several use cases were simulated but due to space constraints we present the case of code length $K = 1280$ and rate 2/3. Figure 6a shows the workload characterization as a function of SNR (with a max. 8 full iterations). The average half-iterations are shown for no DPM, DPM and the average critical half-iterations. The curve corresponding to *No DPM* uses as stopping criterion the HDA rule outlined in [8]. This criterion essentially halts the decoder once the hard-decisions upon the posterior messages cease to change between consecutive iterations. Based upon these results and the error-correction performance loss, the DPM policy is tuned in order to provide the biggest gains in energy savings at the lowest performance loss. Figure 6b shows the bit-error rate (solid lines) and frame-error rate (dashed lines) (BER/FER) for applying the DPM technique.

The achieved energy savings are shown in Figure 6c. The energy savings achieved by an offline strategy (*genie*) are provided as well to visualize how far the DPM policy behaves from its ideal performance. The offline policy produces around 54% of energy savings, this is asymptotically approached by the DPM technique on the high SNR region. This comes from the fact that at high SNR values the critical workload is very low, this suggests that the blocks enter convergence relatively fast, exactly what would be expected at a good channel quality. The achieved energy savings vary between 34% to 54% depending upon the channel quality.

The energy savings from the proposed policy have two components: one due to the inherent stopping criteria and another one due to the system slowdown. The former dominates on the low SNR region and the latter on the mid to high SNR region. At low SNR the stopping criteria detect the potential codeblocks that are not likely to be decoded, whereas at mid to high SNR values the system takes advantage of the fast convergence in order to reduce power consumption.

The area and power overheads for applying DPM on the synthesized decoder are revealed in Table I. The DVFS unit is characterized from the results presented in [19][20]. From



(a) Workload visualization (b) BER/FER performance (c) Average energy savings

Fig. 6. Results for test case $K = 1280$ and rate 2/3.

TABLE II
COMPARISON OF ENERGY SAVINGS TECHNIQUES

Work	Proposed	[21] (SC)	[21] (PR)
Energy savings %	54	17.5	24.5
SNR loss [dB]	0.08	0.34	0.48

those works we extract the data for a buck converter with a switching frequency of 100MHz and a conversion efficiency in the range of 75%-87% with an output voltage range of 0.9V-1.3V.

We compare the achieved average energy savings with works from the prior art in Table II. The SNR loss reported in the table corresponds to the point at $BER = 10^{-5}$ for the corresponding code from each work. Even though the codes among the cited works are different, the SNR loss provides a measure on the impact in performance for each applied power savings technique. The work in [21] analyses individual techniques proposed for energy reduction in Turbo decoding. Among them, we show the reported achieved energy savings for reduction in the number of paths (PR) and the LLR stopping criterion (SC). We acknowledge that in [21] several of the analyzed techniques therein were combined and savings of up to 66% were reported.

V. CONCLUSION

An online dynamic power management policy for iterative decoders has been presented. This technique is aided by the dynamics of the decoding process that can be extracted from a particular convergence metric. A judicious selection of a power mode is carried out during run-time by a power manager that considers the decoding task deadline and the predicted remaining decoding time once the decoder has entered a convergence mode. The proposed technique has been applied to the decoding of Turbo codes where the total number of hard-decision changes in the posterior messages was used as a decision metric. Energy savings of up to 54% were achieved with a relatively low impact on error-correction performance.

REFERENCES

[1] C. Berrou, A Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-correcting coding and decoding: Turbo Codes," *IEEE International Conference on Communication*, pp. 1064–1070, May 1993.

[2] R. Gallager, "Low-Density Parity-Check Codes," *IRE Trans. Inf. Theory*, vol. 7, pp. 21–28, January 1962.

[3] 3GPP UMTS, "General UMTS Architecture," *3GPP TS 23.101 version 7.0.0*, 2007.

[4] 3GPP LTE, "Evolved Universal Terrestrial Radio Access (EUTRA) and Evolved Universal Terrestrial Radio Access Network (EUTRAN)," *3GPP TS 36.300*, 2008.

[5] E. Amador, R. Knopp, V. Rezard, and R. Pacalet, "Dynamic Power Management on LDPC Decoders," in *Proc. of IEEE Computer Society Annual Symposium on VLSI*, July 2010, pp. 416–421.

[6] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *Information Theory, IEEE Transactions on*, vol. 20, no. 2, pp. 284–287, 1974.

[7] L. Benini, R. Bogliolo, and G. De Micheli, "A Survey of Design Techniques for System-Level Dynamic Power Management," *IEEE Trans. on VLSI Systems*, vol. 8, pp. 299–316, 2000.

[8] R.Y. Shao, S. Lin, and M. Fossorier, "Two Simple Stopping Criteria for Turbo Decoding," *IEEE Trans. in Commun.*, vol. 47, pp. 1117–1120, Aug. 1999.

[9] J. Hagenauer, E. Offer, and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. on Inf. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[10] A. Matache, S. Dolinar, and F. Pollara, "Stopping Rules for Turbo Decoders," *TMO Progree Report*, vol. 42, pp. 42–142, Aug. 2000.

[11] W. Wang and G. Choi, "Speculative Energy Scheduling for LDPC Decoding," in *8th International Symposium on Quality Electronic Design*, 2007, pp. 79–84.

[12] W. Wang, G. Choi, and K. Gunnam, "Low-Power VLSI Design of LDPC Decoder Using DVFS for AWGN Channels," in *Proc. of the 22nd International Conference on VLSI Design*, 2009, pp. 51–56.

[13] Ying Tan, P. Malani, Qinru Qiu, and QingWu, "Workload Prediction and Dynamic Voltage Scaling for MPEG Decoding," in *Design Automation, 2006. Asia and South Pacific Conference on*, 2006, p. 6 pp.

[14] Hwisung Jung and M. Pedram, "Continuous Frequency Adjustment Technique Based on Dynamic Workload Prediction," in *VLSI Design, 2008. VLSID 2008. 21st International Conference on*, 2008, pp. 249 –254.

[15] S. Irani, G. Singh, S.K. Shukla, and R.K. Gupta, "An Overview of the Competitive and Adversarial Approaches to Designing Dynamic Power Management Strategies," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 13, no. 12, pp. 1349 – 1361, 2005.

[16] P. Macken, M. Degrauwe, M.V. Paemel, and H. Orguey, "A Voltage Reduction Technique for Digital Systems," in *IEEE International Solid State Circuits Conference*, 1990, pp. 238–239.

[17] Yanbin Liu and A.K. Mok, "An Integrated Approach for Applying Dynamic Voltage Scaling to Hard Real-Time Systems," in *Real-Time and Embedded Technology and Applications Symposium, 2003. Proceedings. The 9th IEEE*, May 2003, pp. 116 – 123.

[18] S. Irani, S. Shukla, and R. Gupta, "Algorithms for Power Savings," in *Proc. of the 14th ACM-SIAM Symposium on Discrete Algorithms*, 2003, pp. 37–46.

[19] W. Kim, M.S. Gupta, G.Y. Wei, and D. Brooks, "System Level Analysis of Fast, Per-Core DVFS using On-Chip Switching Regulators," in *IEEE 14th International Symposium on High Performance Computer Architecture*, 2008, pp. 123–134.

[20] F. Su, W.H. Ki, and C.Y. Tsui, "Ultra Fast Fixed-Frequency Hysteretic Buck Converter with Maximum Charging Current Control and Adaptive Delay Compensation for DVS Applications," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 4, pp. 815–822, April 2008.

[21] J. Kaza and C. Chakrabarti, "Energy-Efficient Turbo Decoder," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2002, pp. 3093 – 3096.