Transmission Channel Noise Aware Energy Effective LDPC Decoding

Thomas Marconi[1], Christian Spagnol[2]
Emanuel Popovici[2], and Sorin Cotofana[1]

[1] Computer Engineering Lab, TU Delft, The Netherlands,
{T.Marconi,S.D.Cotofana}@tudelft.nl
[2] Electrical & Electronic Engineering, University College Cork, Ireland,
c.spagnol@ue.ucc.ie, e.popovici@ucc.ie

Abstract. In communication systems channel quality variation, mostly induced by interferences, mobility, and environmental factors, is an unhindered physical phenomenon, which is usually perceived as a threat in pursuing reliable communication. There is a direct relation between the channel condition and the amount of computational resources and energy that have to be spend in order to reconstruct the correct messages at the reception side. When the quality is good, the decoding requires less resources and energy to identify and correct channel condition induced message errors, while when the channel noise level is high more resources and energy are needed to correct the errors. To be able to properly handle high noise levels while keeping the QoS requirements satisfied, telecom platforms are built upon largely over-designed hardware, i.e., they rely on worse case designs, which results in a substantial energy waste during most of their operation. In this chapter we introduce a methodology to dynamically adapt the platform operation mode to the channel noise level. The main objective is to keep QoS requirements satisfied regardless of the actual channel conditions while minimizing the energy consumption footprint. In particular, we propose a technique to exploit channel noise variability towards energy effective LDPC decoding amenable to adaptable low-energy operation. Endowed with the instantaneous channel noise level knowledge, our technique dynamically adjusts the operating voltage on-the-fly, aiming to achieve the optimal tradeoff between decoder performance and energy consumption without ignoring the fulfillment of the QoS requirements expressed in terms of frame/bit error rate. To demonstrate the capabilities of our proposal we implemented it and other state of the art energy reduction methods in the framework of a fully parallel LDPC decoder mapped on a Virtex-6 FPGA. Our experiments indicate that the proposed technique outperforms state of the art counterparts, in terms of energy reduction, with 71% to 76% and 15% to 28%, w.r.t. early termination without and with DVS, respectively, while maintaining the targeted decoding robustness. Moreover, the measurements suggest that in certain conditions Degradation Stochastic Resonance occurs, i.e., timing faults caused by unpredictable underpowered components in the circuit unexpectedly become supporters rather than enemies of the decoding process.

1 Introduction

To communicate messages from senders to recipients channels are indispensable required. In one hand, we need channels to convey our messages. In the other hand, we want to avoid the channels from corrupting our messages. Channel quality variation is an unhindered physical phenomenon which is usually perceived as a threat in pursuing reliable communication. It is well known that channel quality variation may occur in communication systems as a result of, e.g., multi-path, interferences, mobility, and environmental conditions [1] as illustrated in Figure 1. When the quality is low a large number of errors might occur during the transmission and high performance error correctors are required to recover the original message. However, if the quality is good, the system experiences less errors case in which decoders with lower decoding capabilities are enough to fulfill the Quality of Service (QoS) requirements. Since the systems are designed to meet a target acceptable error rate even in the worst-case scenario, e.g., the highest expected channel noise level, the decoders are over-designed. Thus decoders have excess performance during good channel conditions and as the worst-case, i.e., bad condition, rarely occurs, significant energy consumption is wasted during most of their operation.

In view of this observation, the main question that we are addressing in this work is "Can we adapt the decoder performance to the channel status to prevent energy over-consumption?" We positively answer this question by proposing a technique to trade off performance with energy while fulfilling error rates requirements by exploiting channel noise variability towards energy effective Low Density Parity Check (LDPC) decoding [2]. Decoding at the required performance is the key idea behind our method to diminish energy consumption and we determine the operation conditions resulting in the highest energy savings by actively monitoring the channel noise. More specifically, we diminish the supply voltage value when the channel is in a good condition to save energy and vice versa we increase it when the channel is getting worse for meeting the target error rate. To be able to properly adjust at run time the power supply voltage to the channel status we perform at design time a decoder pre-characterization, i.e., we measure decoder's Frame Error Rate (FER), Bit Error Rate (BER), and energy/bit under voltage scaling on a variety of noisy channels. The main objective is to minimize the energy consumption while preserving the decoder performance in terms of FER and BER. The operating voltage adaptation is done based on the estimated Signal-to-Noise Ratio (SNR) of an Additive White Gaussian Noise (AWGN) channel with Binary Phase Shift Keying (BPSK) modulation.

Although our technique is applicable to any electronic system for which the power supply vs performance relation can be pre-characterized, in this work, we experimentally evaluate it for an LDPC decoder implemented on Field-Programmable Gate Array (FPGA). We utilize as discussion vehicle an LDPC
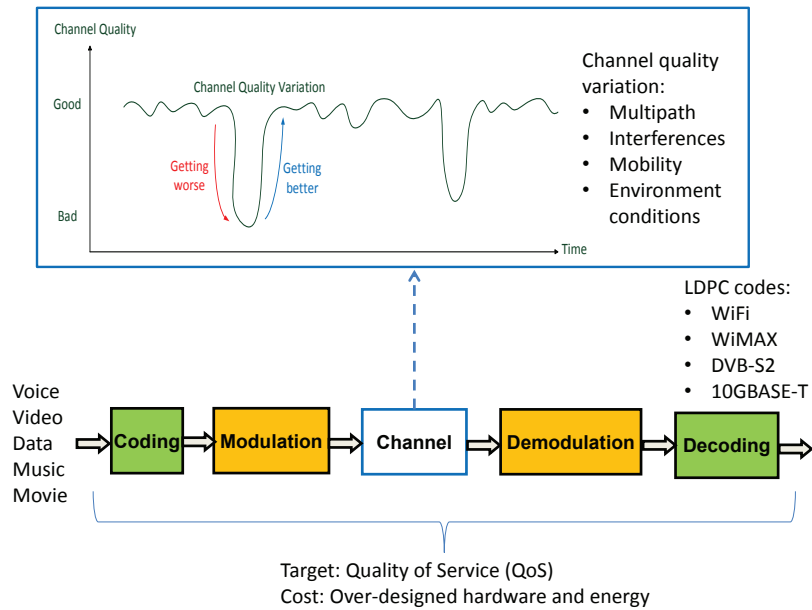
Fig. 1: Illustration of Communication Systems

decoder based on Log Likelihood Ratio (LLR) Belief Propagation (BP) algorithm implemented with fully parallel architecture. The reasons for choosing the LDPC decoder [2] for evaluating our proposed technique are (i) the superiority of its decoding performance, which is very close to the Shannon limit [3], and (ii) its popularity, being adopted by many communication standards, e.g., WiFi [4], WiMAX [5], DVB-S2 [6], 10GBase-T [7]. The decoder is automatically generated and the energy consumption is measured directly on the FPGA board by accessing the PMBus through the USB interface adapter. In contrast to [8], we adjust the supply voltage value by directly controlling the board internal power supply, thus we do not rely on an external power supply or need to operate board modifications for effective and efficient voltage scaling experiments. We note that we report measured data, gathered from experiments on a Virtex-6 FPGA, and not from theoretical analysis, e.g., density evolution, EXIT charts, or from, e.g., Monte-Carlo simulation.

For comparison purpose we mapped into the FPGA three versions of the considered LDPC decoder as follows: (i) equipped with our technique, (ii) with powering off capability/early termination (ET) technique, e.g., [9], operated at nominal supply voltage value, and (iii) with a hybrid scheme combining ET and the Dynamic Voltage Scaling (DVS) techniques in [8]. Our experiments indicate that the proposed technique outperforms the other schemes, resulting in 71% to 76% and 15% to 28% energy reduction w.r.t. ET without and with DVS,

respectively, while maintaining the required decoding performance. Moreover, we observe that in certain condition Degradation Stochastic Resonance (DSR) [10, 11] occurs, i.e., timing errors caused by voltage scaling improve the decoding performance, and by implication diminish the energy consumption, due to the fact that unpredictable underpowered components facilitate rather than impede the decoding process.

Many approaches have been proposed to reduce energy consumptions of LDPC decoders. Generally speaking voltage scaling has been frequently applied in Complementary Metal–Oxide–Semiconductor (CMOS) integrated circuits as by reducing the supply voltage we can gain energy efficiency at the expense of a longer circuit delay. However, as timing violation may occur if the circuit delay gets longer than the clock period, existing approaches either avoid this situation by tuning the operating frequency accordingly or correct potential errors by means of additional hardware.

To reduce LDPC decoders energy consumption, Dynamic Voltage Frequency Scaling (DVFS) based on the estimation of the maximum number of iterations was proposed in [12] and [13]. In [14] a Signal to Noise Ratio (SNR) estimator is utilized to guide the operating frequency scaling based on the target throughput and channel conditions. Subsequently, assisted by an error detector and a critical path replica, the suitable voltage that ensures no timing violation is determined. In [15] Reduced-Precision Replicas (RPR) of the bit and check nodes are utilised to detect and correct voltage scaling induced errors while in [16], Voltage Over-Scaling (VOS) and RPR are combined.

In our proposal, the best operating voltage is chosen based on knowledge of the underpowered LDPC decoder behavior. The operation supply voltage is chosen such that the decoding performance of the decoder operated at the typical voltage, measured in terms of FER/BER, is maintained or a specific target decoding performance is achieved. Similar to [14], we utilize the SNR estimator of existing communication systems but we do not modify the frequency in order to maintain the throughput and we allow for timing violation as long as the target performance is still achievable. We do not make use of additional hardware for error detection/correction or reduced precision replicas thus our approach results in smaller hardware overhead. In [15][16], the supply voltage is fixed at runtime while our method adjusts the supply voltage according to the channel conditions at runtime. Last but not least our technique does not require any decoder modifications hence it can be easily combined with other techniques.

In summary, the main contributions of this work are: (i) A technique for exploiting channel noise variability, (ii) Measured results from FPGA based experiments demonstrating the energy savings capabilities of the proposed method, (iii) A new way to perform voltage scaling on commercial FPGAs without relying on an external power supply, and (iv) Evidence of DSR occurrence in LDPC decoder implementations. This chapter is an extended version of our previous conference paper in [17].

The rest of the chapter is organized as follows. It starts with a short introduction of Sum-Product LDPC Decoding in Section 2. In Section 3, we present the proposed technique in the context of an LDPC decoder and evaluate it in Section 4. In Section 5, we close the discussion with some conclusions.

2 Sum-Product LDPC Decoding

Before we introduce Sum-Product (SP) LDPC decoding, we start with a brief introduction within the LDPC concept and decoding methodology. Generally speaking, adding parity bits to source information bits, i.e., creating a codeword, is a way to achieve reliable communication through unreliable channels between two parties located at the two channel ends. Each codeword should satisfy a set of pre-determined parity check equations and parties need to send only codewords on the channel. By doing so, the receiving party gets the means to check if the received message is right and if this is not the case to apply error corrective changes. For example, $[BN_1 BN_2 BN_3 BN_4 BN_5 BN6] = [101110]$ is a codeword that satisfies the following set of pre-determined parity check equations:

$$BN_1 \oplus BN_2 \oplus BN_4 = 0 \tag{1}$$

$$BN_2 \oplus BN_3 \oplus BN_5 = 0 \tag{2}$$

$$BN_1 \oplus BN_5 \oplus BN_6 = 0 \tag{3}$$

$$BN_3 \oplus BN_4 \oplus BN_6 = 0 \tag{4}$$

These parity check equations, taken from [18], can be written as a parity check matrix H presented in Figure 2(a). Each row and each column of the H matrix represent a parity check equation and a codeword bit, respectively. This H matrix corresponds to a Tanner graph [19] visualized in Figure 2(b), which describes an iterative decoding process. The Tanner graph in Figure 2(b) consists of two sets of vertices: 6 bottom vertices for visualizing 6 Bit Nodes (BNs) and 4 top vertices for visualizing 4 Check Nodes (CNs). Bit node n is connected to check node m if the element of column n row m of the parity-check matrix H is '1'. We use this figure to introduce general idea of decoding and SP LDPC decoding as a particular example.

Now, we start with general idea of decoding. Let say we sent a message $[BN_1 BN_2 BN_3 BN_4 BN_5 BN6] = [101110]$ from the Moon to the Earth using the code in Figure 2. When it reaches the Earth, due to the noisy channel BN_4 flips from '1' to '0', the received message becomes $[BN_1 BN_2 BN_3 BN_4 BN_5 BN6] = [101010]$. Since the incoming message is not a codeword, the receiving decoder knows that this message is not the message transmitted by the sender. Subsequently the decoder tries to correct the error, e.g., comparing the received message with the closest valid codeword. In this case, the closest codeword is [101110] at a Hamming distance 1 away from the received message [101010]. The identification of the closest codeword is guided by the parity check equations which we can perceive as an agreement between two parties to conduct

reliable communications. If the agreement is violated, it means that something went wrong during the transmission on the unreliable noisy channel and corrective actions are required.

(a) Parity Check Matrix H

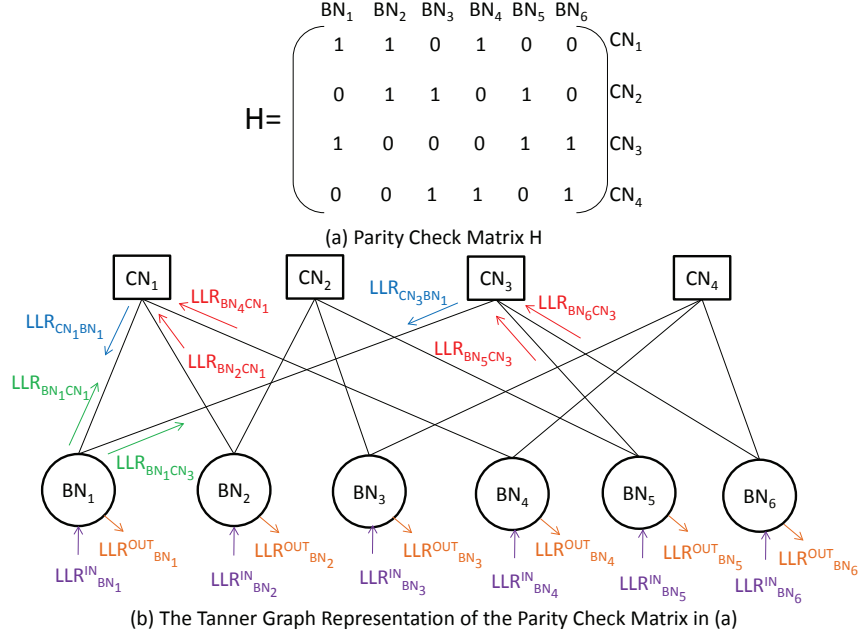(b) The Tanner Graph Representation of the Parity Check Matrix in (a)

Fig. 2: An Example of Parity Check Matrix (a) and its Tanner Graph Representation (b)

To get more insight into the decoding process we assume as a discussion vehicle the Sum-Product (SP) LDPC decoder which is a soft decision decoder based on the message-passing algorithm, as visualized in Figure 2(b). In soft decision decoding, each received bit x is represented as $P(x = 1)$, i.e., the probability that x is logic '1'. The probability is expressed as a Log-Likelihood Ratio (LLR) in SP decoding to reduce the implementation complexity by turning multiplications into additions. LLR of x is defined as:

$$LLR_x = log_e \left[\frac{P(x = 0)}{P(x = 1)} \right] \tag{5}$$

$LLR_{BN_n}^{IN}$ stands for a message received from channel in form of the LLR of the bit node n. Messages are iteratively passing by between Bit Nodes (BNs) and Check Nodes (CNs) along the edges of the Tanner graph as illustrated in Figure 2(b). A message from a bit node n to a check node m in LLR form is written as $LLR_{BN_n CN_m}$. In the opposite direction, a message from a check node m to a bit node n in LLR form is denoted as $LLR_{CN_m BN_n}$. Each output decoding in

LLR form for a bit node n, expressed as $LLR_{BN_n}^{OUT}$, is utilized for (i) making hard decision at the end of each iteration for decoders with early termination and (ii) the determination of the final output at the end of the decoding process.

Let us focus our attention to CN_1 in Figure 2(b). Given that $BN_2CN_1 \oplus BN_4CN_1 = 0$ is true if both BN_2CN_1 and BN_4CN_1 have the same value either '0' or '1', the probability of $(BN_2CN_1 \oplus BN_4CN_1 = 0)$ can be computed as:

$$P(BN_2CN_1 \oplus BN_4CN_1 = 0) = P(BN_2CN_1 = 0)P(BN_4CN_1 = 0) + P(BN_2CN_1 = 1)P(BN_4CN_1 = 1) \quad (6)$$

Since $P(x = 0) = 1 - P(x = 1)$, Equation (6) becomes:

$$P(BN_2CN_1 \oplus BN_4CN_1 = 0) = \tfrac{1}{2} + \tfrac{1}{2}\{[1 - 2P(BN_2CN_1 = 1)][1 - 2P(BN_4CN_1 = 1)]\} \quad (7)$$

and:

$$P(BN_2CN_1 \oplus BN_4CN_1 = 1) = \tfrac{1}{2} - \tfrac{1}{2}\{[1 - 2P(BN_2CN_1 = 1)][1 - 2P(BN_4CN_1 = 1)]\} \quad (8)$$

By dividing Equation (7) by Equation (8), we obtain:

$$\frac{P(BN_2CN_1 \oplus BN_4CN_1 = 0)}{P(BN_2CN_1 \oplus BN_4CN_1 = 1)} = \left(\frac{\tfrac{1}{2} + \tfrac{1}{2}\{[1 - 2P(BN_2CN_1=1)][1-2P(BN_4CN_1=1)]\}}{\tfrac{1}{2} - \tfrac{1}{2}\{[1 - 2P(BN_2CN_1=1)][1-2P(BN_4CN_1=1)]\}} \right) \quad (9)$$

and by multiplying the numerator and the denominator of Equation (9) by 2, we get:

$$\frac{P(BN_2CN_1 \oplus BN_4CN_1 = 0)}{P(BN_2CN_1 \oplus BN_4CN_1 = 1)} = \left(\frac{1 + \{[1 - 2P(BN_2CN_1=1)][1-2P(BN_4CN_1=1)]\}}{1 - \{[1 - 2P(BN_2CN_1=1)][1-2P(BN_4CN_1=1)]\}} \right) \quad (10)$$

To satisfy check node 1, we need to submit to Equation (1):

$$LLR_{CN_1BN_1} = log_e \left[\frac{P(BN_2CN_1 \oplus BN_4CN_1 = 0)}{P(BN_2CN_1 \oplus BN_4CN_1 = 1)} \right] \quad (11)$$

By substituting Equation (10) to Equation (11) we obtain:

$$LLR_{CN_1BN_1} = log_e \left(\frac{1 + \{[1 - 2P(BN_2CN_1=1)][1-2P(BN_4CN_1=1)]\}}{1 - \{[1 - 2P(BN_2CN_1=1)][1-2P(BN_4CN_1=1)]\}} \right) \quad (12)$$

and since $log_e \left(\frac{1+x}{1-x} \right) = 2tanh^{-1}(x)$, Equation (12) becomes:

$$LLR_{CN_1BN_1} = 2tanh^{-1}\{[1 - 2P(BN_2CN_1 = 1)][1 - 2P(BN_4CN_1 = 1)]\} \quad (13)$$

Since $tanh \left[\tfrac{1}{2} log_e \left(\frac{1-x}{x} \right) \right] = 1 - 2x$, thus Equation (13) is transformed into:

$$LLR_{CN_1BN_1} = 2tanh^{-1} \left\{ tanh \left[\tfrac{1}{2} log_e \left(\frac{1-P(BN_2CN_1=1)}{P(BN_2CN_1=1)} \right) \right] tanh \left[\tfrac{1}{2} log_e \left(\frac{1-P(BN_4CN_1=1)}{P(BN_4CN_1=1)} \right) \right] \right\}$$
$$(14)$$

Given the LLR definition in Equation (5), Equation (14) becomes:

$$LLR_{CN_1BN_1} = 2tanh^{-1} \left[tanh \left(\frac{LLR_{BN_2CN_1}}{2} \right) tanh \left(\frac{LLR_{BN_4CN_1}}{2} \right) \right] \quad (15)$$

By separating signs and magnitudes of $LLR_{BN_2CN_1}$ and $LLR_{BN_4CN_1}$, Equation (15) can be written as:

$$LLR_{CN_1BN_1} = 2tanh^{-1}\left[tanh\left(\frac{sign(LLR_{BN_2CN_1})|LLR_{BN_2CN_1}|}{2}\right)tanh\left(\frac{sign(LLR_{BN_4CN_1})|LLR_{BN_4CN_1}|}{2}\right)\right]$$

(16)

Separating sign and magnitude of $LLR_{CN_1BN_1}$ in Equation (16) results:

$$sign(LLR_{CN_1BN_1}) = sign(LLR_{BN_2CN_1})sign(LLR_{BN_4CN_1})$$

(17)

and:

$$|LLR_{CN_1BN_1}| = 2tanh^{-1}\left[tanh\left(\frac{|(LLR_{BN_2CN_1})|}{2}\right)tanh\left(\frac{|(LLR_{BN_4CN_1})|}{2}\right)\right]$$

(18)

Since $log_e(xy) = log_e(x) + log_e(y)$, therefore Equation (18) also can be restructured into:

$$|LLR_{CN_1BN_1}| = 2tanh^{-1}log^{-1}\left(\left\{log\left[tanh\left(\frac{|(LLR_{BN_2CN_1})|}{2}\right)\right] + log\left[tanh\left(\frac{|(LLR_{BN_4CN_1})|}{2}\right)\right]\right\}\right)$$

(19)

Let us define $\phi(x)$ as:

$$\phi(x) = -log[\tanh(x/2)] = log\left(\frac{e^x + 1}{e^x - 1}\right)$$

(20)

If we find the inverse of $\phi(x)$ through very simple mathematics, we get:

$$\phi^{-1}(x) = log\left(\frac{e^x + 1}{e^x - 1}\right) = \phi(x)$$

(21)

Using the definition of $\phi(x)$, Equation (19) becomes:

$$|LLR_{CN_1BN_1}| = 2\tanh^{-1}\log^{-1}\{-[\phi(|LLR_{BN_2CN_1}|) + \phi(|LLR_{BN_4CN_1}|)]\}$$

(22)

If we find the inverse of Equation (20) through very simple mathematics, we obtain:

$$\phi^{-1}(x) = 2\tan^{-1}\log^{-1}(-x)$$

(23)

Substitute Equation (23) to Equation (22):

$$|LLR_{CN_1BN_1}| = \phi^{-1}[\phi(|LLR_{BN_2CN_1}|) + \phi(|LLR_{BN_4CN_1}|)]$$

(24)

Using $\phi^{-1}(x) = \phi(x)$ in Equation (21), Equation (24) is transformed into:

$$|LLR_{CN_1BN_1}| = \phi[\phi(|LLR_{BN_2CN_1}|) + \phi(|LLR_{BN_4CN_1}|)]$$

(25)

By doing the same computation for CN_3, we get:

$$|LLR_{CN_3BN_1}| = \phi[\phi(|LLR_{BN_5CN_3}|) + \phi(|LLR_{BN_6CN_3}|)]$$

(26)

and:

$$sign(LLR_{CN_3BN_1}) = sign(LLR_{BN_5CN_3})sign(LLR_{BN_6CN_3}) \qquad (27)$$

Using $LLR_{CN_1BN_1}$ and $LLR_{CN_3BN_1}$, we can compute $LLR_{BN_1CN_1}$, $LLR_{BN_1CN_3}$, and $LLR_{BN_1}^{OUT}$ as:

$$LLR_{BN_1CN_1} = LLR_{BN_1}^{IN} + LLR_{CN_3BN_1} \qquad (28)$$

,

$$LLR_{BN_1CN_3} = LLR_{BN_1}^{IN} + LLR_{CN_1BN_1} \qquad (29)$$

, and:

$$LLR_{BN_1}^{OUT} = LLR_{BN_1}^{IN} + LLR_{CN_1BN_1} + LLR_{CN_3BN_1} \qquad (30)$$

This $LLR_{BN_1}^{OUT}$ is used for making a hard decision for bit 1 using this equation:

$$BN_1^{OUT} = \begin{cases} 0, & \text{if } LLR_{BN_1}^{OUT} \text{ is positive} \\ 1, & \text{if } LLR_{BN_1}^{OUT} \text{ is negative} \end{cases} \qquad (31)$$

This hard value BN_1^{OUT} is used with other bits for parity check to determine whether the vector $[BN_1^{OUT}BN_2^{OUT}BN_3^{OUT}BN_4^{OUT}BN_5^{OUT}BN_6^{OUT}]$ is a codeword or not using Equation (1) to Equation (4). If the vector is a valid codeword, the decoder with early termination stops decoding and then sends this vector as the final output of decoding. $LLR_{BN_1CN_1}$ and $LLR_{BN_1CN_3}$ are used for next iteration using the same computation of bit 1. In the previous derivations we concentrated on the calculation of bit 1, computation of all the other bits can be done in the same way. The computation is stopped either when the decoder finds a codeword (for decoder with early termination) or the decoder gives up, i.e., the decoder could not find the codeword after a pre-defined maximum number of iterations.

3   Proposed Technique

The main concept behind the proposed technique is presented in Figure 3. Maintaining the decoding performance (in terms of FER/BER) to its required value by actively monitoring the channel noise is the key idea to prevent energy over-consumption. More precisely, we turn the supply voltage up when the channel is getting worse for meeting the target error rate and vice versa we turn it down when the channel is in a good condition to save energy. The question is "How far can we turn the supply voltage down in good channel conditions or up in bad channel conditions?" If we increase the voltage too much when the channel is getting worse, we may spill energy, while if the voltage is not big enough, the target error rate cannot be satisfied. Similar situations may also occur when turning down the voltage in good channel conditions. To determine the appropriate decoder operating voltage at a specific channel condition, we need to know the decoder behavior by means of a *pre-characterization* process as detailed in

Subsection 3.1. The pre-characterization results are then used to compute the decoding operating voltage for any specific channel condition. These values are stored in an LUT and utilized in guiding the decoder to meet the required target error rate while consuming as low energy as possible in the *adaptation* process detailed in Subsection 3.2.
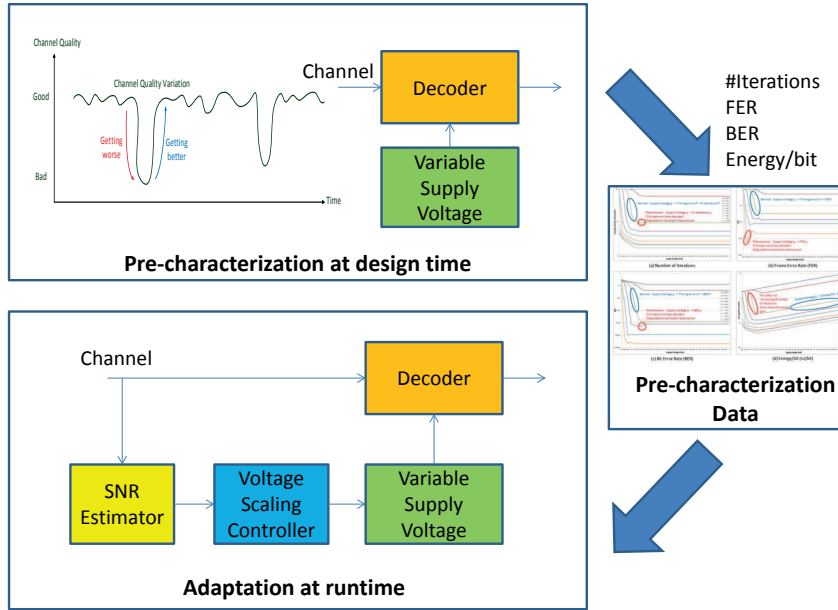

Fig. 3: Transmission Channel Noise Aware Energy Effective Decoding

3.1 Pre-characterization

It is clear that the decoder has to be aware of the channel condition for taking the best runtime decisions. Another important information the decoder needs to have is its own behavior when operating at different supply voltages and for various channel conditions in terms of performance/correction capability and energy efficiency. For this reason, in order to equip the decoder with the required knowledge, we need to make real measured data known to the decoder. Thus, during *pre-characterization*, we need to measure decoder FER, BER, and energy/bit on a variety of noisy channels and voltage conditions. There are 3 steps to do *pre-characterization*: (1) *generation*, (2) *setup*, and (3) *run* as depicted in Figures 4, 5, and 6, respectively.

Generation (see Figure 4) is the process of creating the decoder for *pre-characterization*. The decoder VHDL code is produced by a tool designed for
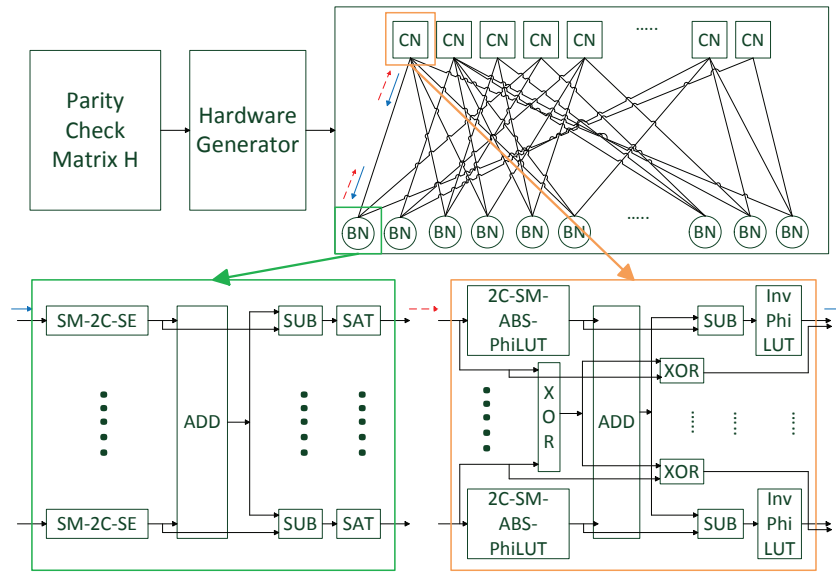
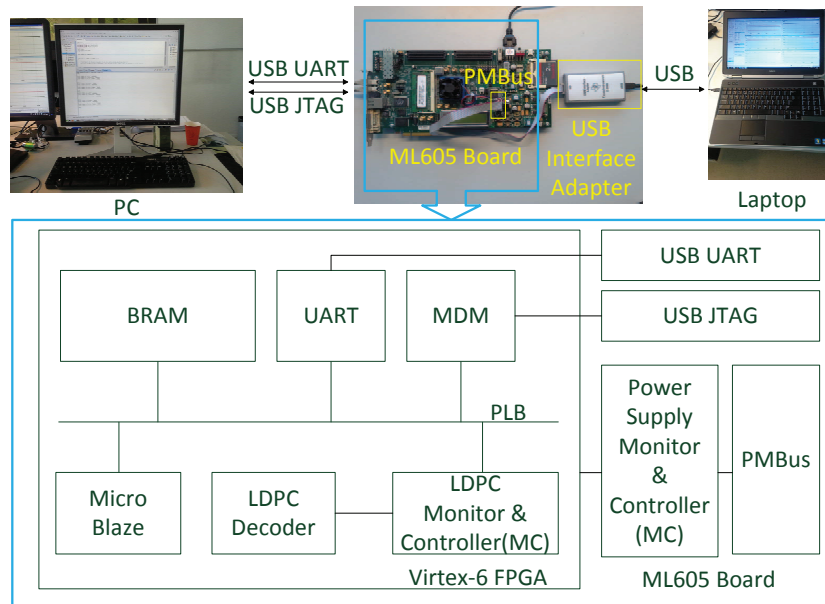Fig. 4: Step 1 of Pre-characterization: Generation
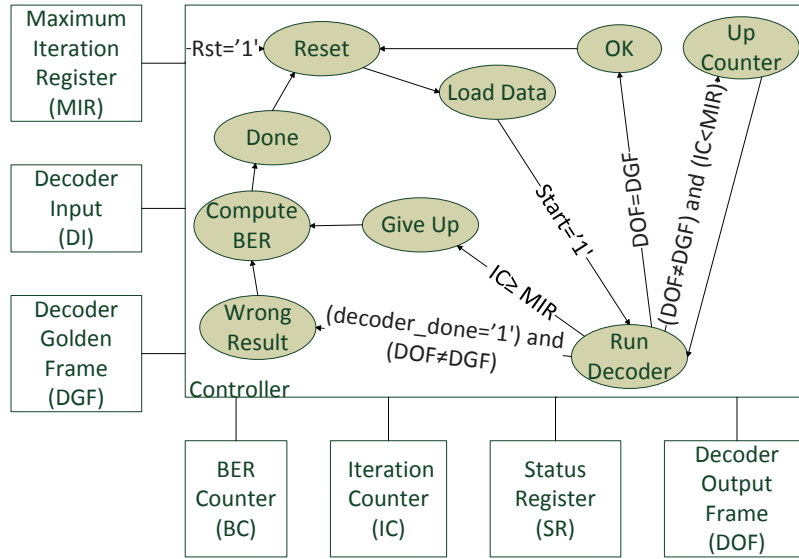

Fig. 5: Step 2 of Pre-characterization: Setup

Fig. 6: Step 3 of Pre-characterization: Run

the automatic generation of LDPC decoder fully parallel IP cores starting from an H matrix, regular or irregular, and the number of bits to represent the channel message. The presented results are obtained using a MacKay A matrix [3] with dimensions n=1000, k=500, and 4-bit 2's complement fixed point number representation of the message. Each fixed point number consists of 1 sign bit, 2 integer bits, and 1 fractional bit. The implemented decoder is based on Log Likelihood Ratio (LLR) Belief Propagation (BP) or the sum-product algorithm [2]. The parallel implementation consists in a one to one mapping of the bit and check nodes in the Tanner graph to the respective bit and check modules. The edges in the graph become physical buses of width equal to the chosen precision. In a fully parallel implementation the number and complexity of the interconnections may result in implementations where the area is dominated by the routing. However it is highly efficient from a power and speed point of view. For these reasons and given the high density and connection capability of modern FPGA chip, a fully parallel architecture has been chosen as the testcase for the research presented here. The maximum number of iterations is set at 100. Two clock cycles are needed for each iteration. LLRs represented by two's complement fixed point numbers are the inputs to the CN while its outputs are represented as Sign Magnitude (SM) numbers. The Check Node (CN) converts each 2's Complement (2C) number to a Sign Magnitude (SM) number using its 2C-SM converter. The absolute value of this SM number is then computed by ABS. This absolute value is further processed to obtain its $\phi(x) = -log(\tanh(x/2)) = log(\frac{e^x+1}{e^x-1})$ function.

The function ϕ is even and has the property that $\phi^{-1}(x) = \phi(x)$ for $x > 0$. In this work, each ϕ or ϕ^{-1} function is approximated by a 4-bit Look-Up Table (LUT) denoted as phiLUT or invphiLUT in the figure. This choice allows to take advantage of the structure of FPGA slices and the free availability of LUTs. To ease automatic hardware generation in a parametric way, all related $\phi(x)$ values are added (using ADD). The output of ADD is then subtracted (using SUB) by each specific $\phi(x)$ value. The subtraction result is then fed to $\phi^{-1}(x)$ function to obtain the magnitude bits as part of sign magnitude representation at the output of CN. In similar way like processing absolute value, each sign bit is processed separately using XOR. The input(output) of BN is represented as sign magnitude number(2's complement number). Each SM number at the input is converted to 2's complement number (denoted as SM-2C) and then sign extended (presented as SE) by $\frac{N+2}{2}$ before being processed further, where N is the number of connected CNs. The same reason like CN for making an easy automation of hardware generation, all related input values of each Bit Node (BN) are added (using ADD) and then subtracted (using SUB) by each specific output value of the SM-2C-SE. Since the result can be a number that cannot be represented by the number of bits of representation, we need to process it further by taking its saturated value using SAT operation.

Setup is the process of building the experimental platform in Figure 5. A PC is used for synthesizing the hardware platform targeting Xilinx Virtex-6 FPGA: XC6VLX240T-1FFG1156 using Xilinx CAD tools version 13.4. The PC is also used for downloading the bitstream file to the Xilinx ML605 board through the USB JTAG interface and for monitoring/capturing the number of iterations, FER, and BER through the USB UART. The energy/bit is obtained using the Fusion Digital Power Designer from Texas Instrument running on the Laptop through Texas Instrument USB Interface adapter by reading PMBus, accessing Power Supply Monitor and Controller inside the board. This technique allows for separately controlling and monitoring each on-board specific internal power supply. The measurements are done only for the internal circuits of the FPGA. The supply voltage is adjusted by directly controlling the internal power supply of the FPGA of the targeted board, keeping other supply voltages unchanged. In contrast to [8], no additional external power supply is needed. To simulate realistic scenarios we use an AWGN channel with BPSK modulation (with mapping $0 \to 1$ and $1 \to -1$) for our experiments. The generated input vectors are fed to the decoder by the MicroBlaze processor resident on the FPGA. BRAM is used to store software and data for Microblaze execution.

Run is the process of running the decoder on the experimental platform for *pre-characterization* purpose as depicted in Figure 6. This functionality is mainly operated by LDPC Monitoring and Controller (MC). The MC monitors the condition of decoding, feeds LLRs to the decoder, and computes BER. The Maximum Iteration Register (MIR) stores the maximum number of iterations allowed for decoding. This register is initialized by the MicroBlaze. If the decoder reaches the maximum number of iterations without reaching a valid code word, i.e., correcting the errors, MC goes to "Give Up" state. Soft messages in form

LLRs are fed by MC to the LDPC decoder through the register Decoder Input (DI). This register can be accessed by the MicroBlaze through the PLB bus. The Decoder Golden Frame (DGF) is the original frame sent by the transmitter and conveyed by the MicroBlaze to the MC for evaluating purposes. By comparing this frame with the Decoder Output Frame (DOF), the controller determines the decoding success rate for computing FER and BER. The computed BER is stored in the BER Counter (BC) and can be accessed by the MicroBlaze. The Status Register (SR) indicates the decoder status and it is accessible by the MicroBlaze. The possible states are: (1) successful decoding denoted as "OK" state, (2) giving up decoding stored as "Give Up" state, or (3) wrong result decoding written as "Wrong Result" state means that the decoder can satisfy all check nodes but it is not the right frame as it was sent by the LDPC encoder. This can happen if the severity of noisy channel alters the frame to another valid codeword, which is considered as an error. The MC enters "Compute BER" state and starts to compute BER when errors occur.

Based on Post-PAR Static Timing Report of Xilinx tools, the minimum clock period is 19.992 ns (i.e., the maximum frequency is 50.020 MHz). The actual implementation is clocked at 50 MHz. Therefore, the throughput at its maximum iterations is 250 Mbps for all experiments. The complete experimental platform including the decoder requires 31433 FFs, 117231 LUTs, and 16 BRAMs whereas the decoder itself needs 29281 FFs and 114968 LUTs. The *pre-characterization* results of average number of iterations, FER, BER, and Energy/bit (nJ/bit) when varying the power supply value from 1V to 0.67V and the channel SNR from 10dB to 1dB are presented in Figures 7 to 10, respectively. Each SNR has its own minimum supply voltage after which the number of iterations starts to increase sharply as one can observe in Figure 7. In general, the increase starts earlier for lower SNR channels and this behavior can be related to the fact that the decoder can do self-correction easier for higher SNR channels where there is not much noise involved. Each SNR has its own specific minimum supply voltage after which its number of iterations goes to the maximum number of iterations which is 100. For the majority of the results it can be seen that when the supply voltage is lowered, the number of iterations stays constant for a while and then increases for the decoder to tackle timing errors. It is unexpected but interesting to note that sometimes, the average number of iterations decreases even if the supply voltage is reduced, which suggests that sometimes the timing errors can help the decoder converging to the correct codeword. This phenomenon is called Degradation Stochastic Resonance (DSR) [11] or Stochastic Resonance (SR) [10] and it can be also observed in Figures 8 and 9 where we present the measured results for FER and BER, respectively. This suggests that voltage reduction can sometimes help improving the decoder performance. Finally, in Figure 10, the measured energy/bit for various SNRs is presented. This figure shows the effect of the law of diminishing returns. The energy/bit decreases by scaling the voltage, however, after a certain point, an increase in energy/bit is visible. This is due to the effect of increasing of number of iterations that diminishes the energy gain we get from reducing the voltage.
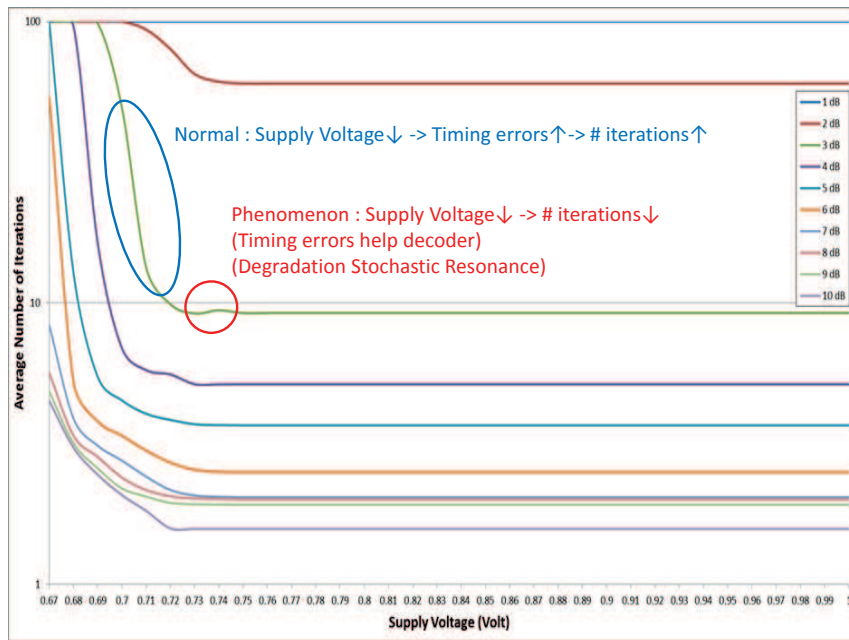
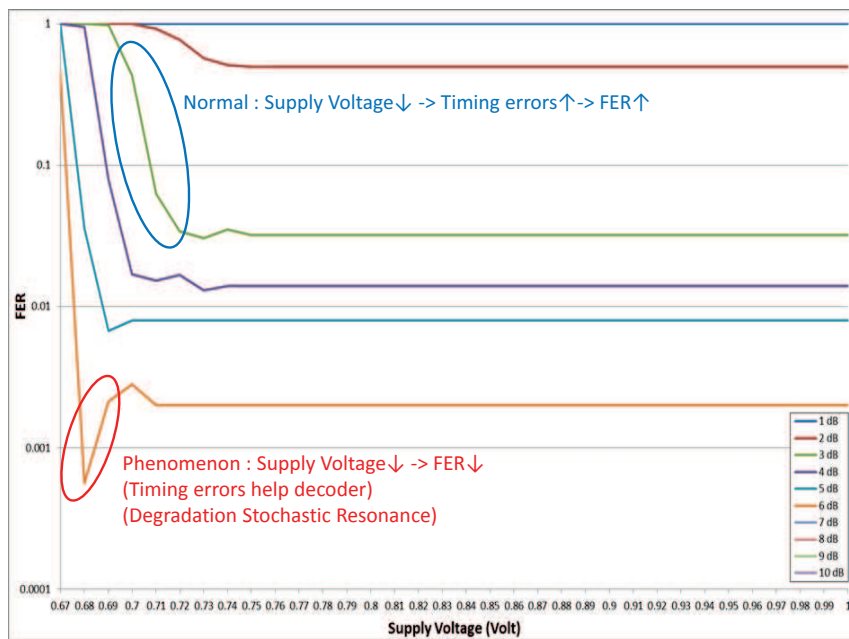Fig. 7: Pre-characterization Results: Number of Iterations
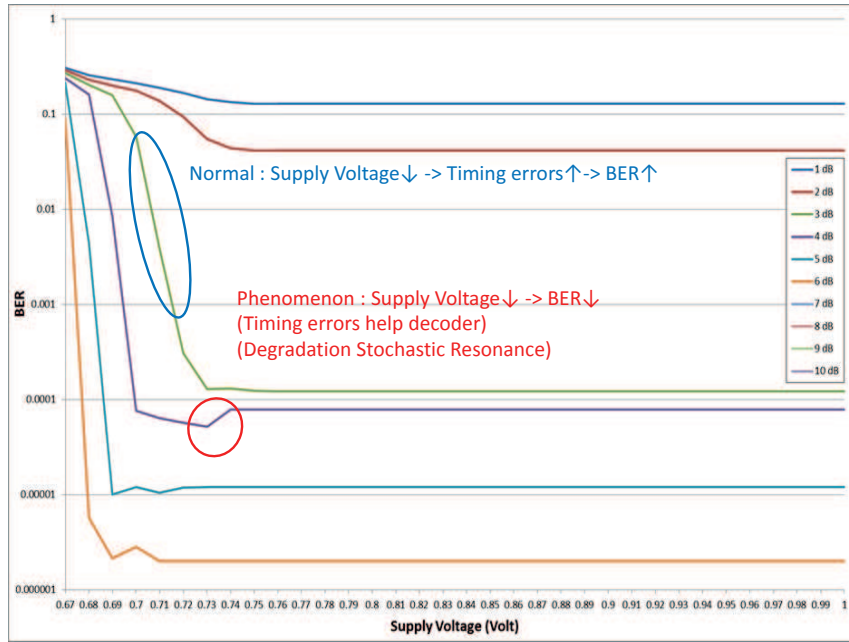

Fig. 8: Pre-characterization Results: FER

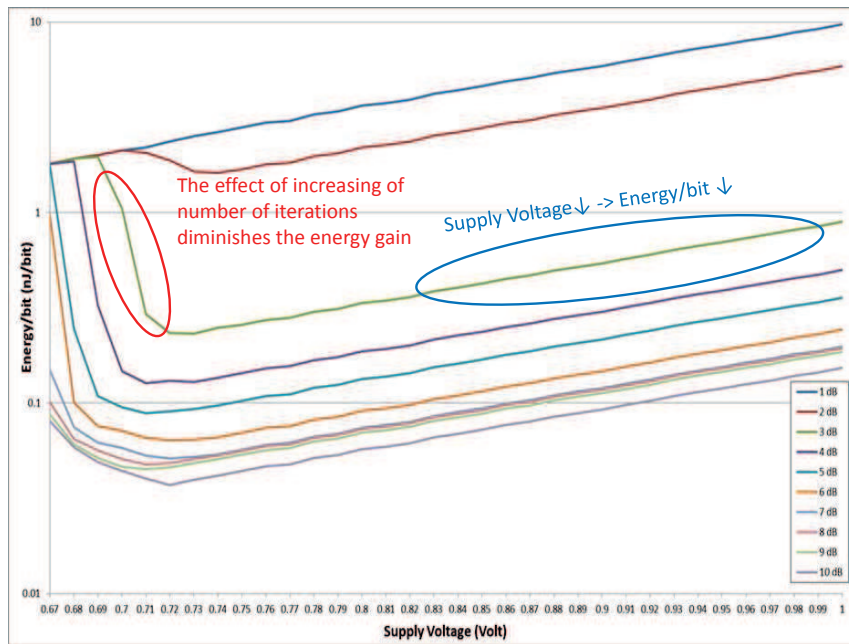Fig. 9: Pre-characterization Results: BER

Fig. 10: Pre-characterization Results: Energy/bit (nJ/bit)

3.2 Adaptation

The voltage scaling controller for the targeted LDPC decoder depicted in Figure 11 operates as follows. It gets SNR information from the SNR estimator and changes the operating supply voltage at runtime based on the knowledge it has from the measured information gathered during the *pre-characterization* stage. We note that given that in communication systems with adaptive coding and modulation, the SNR estimator is a standard system component. The basic principle of the adaptation is to trade off over-needed performance for energy saving through active channel quality monitoring. More precisely, we turn down the supply voltage when the channel is in good condition, hence allowing energy saving. However, for preserving target performance, it is required to turn the voltage up when the channel SNR is getting worse. The objective is to minimize the energy not the voltage while ensuring the decoder achieves its needed performance. Note that minimizing the voltage may not always improve the energy efficiency, because the number of iterations of the decoder may increase due to induced timing errors. Thus, this diminishing returns effect needs to be considered when choosing the operating voltage.
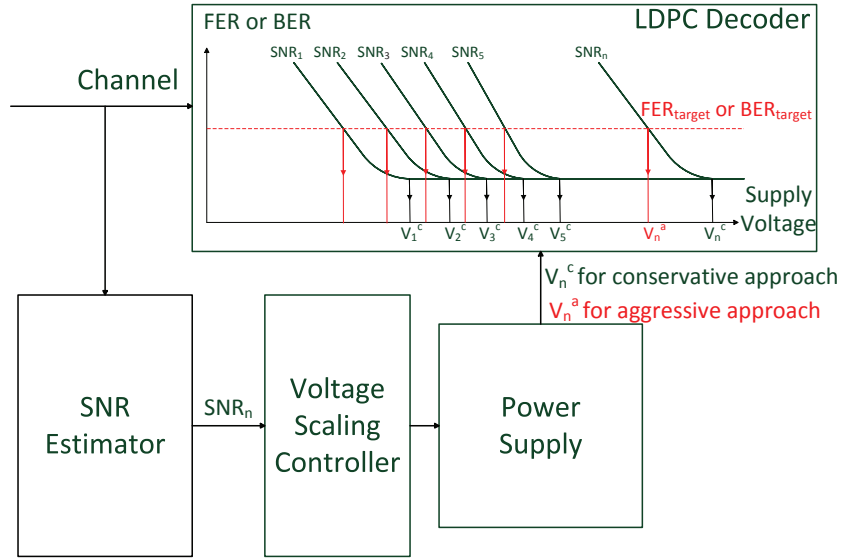

Fig. 11: Hardware Design of Adaptation

We developed two different adaptation strategies as follows: (i) the conservative approach which guaranties that the original decoder performance is always preserved and (ii) the aggressive approach which only concentrates on achieving
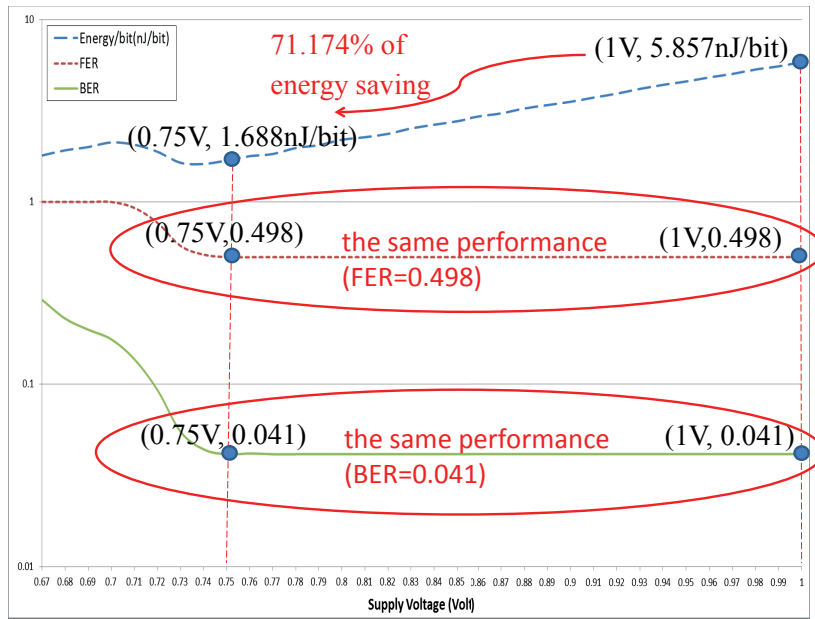
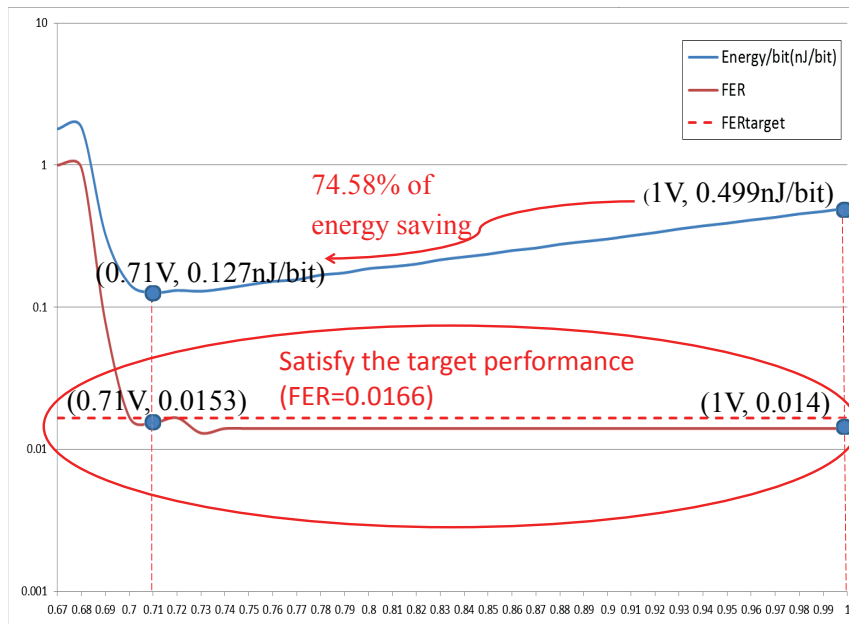Fig. 12: An Example Result of Conservative Approach

Fig. 13: An Example Result of Aggressive Approach

the required target performance. To maintain identical performance while ensuring higher energy efficiency for the channel condition characterized by SNR_n, the operating voltage V_n is moved towards the point where the energy/bit is minimized and at the same time the FER and BER remain identical to those of the decoder operated at the typical voltage $V_{Typical}$. By its conservative nature this approach may still sometimes result in energy waste as it tries to mimic the worse case designed decoder and not to just fulfill the target performance requirements. In view of this the aggressive strategy is designed to enable the decoder to adapt itself such that it delivers the required correction capability while minimizing the energy consumption.

To discuss how the Voltage Scaling Controller (VSC) determines the operating voltage of the decoder formally, some definitions are introduced as follows.
Definition 1: $FER_{Typical}$ $(BER_{Typical})$ is the FER (BER) corresponding to the supply voltage $V_{Typical}$.
Definition 2: $V_{min}^{FER_{Typical}}$ $\left(V_{min}^{BER_{Typical}}\right)$ is the minimum supply voltage for which FER $\leq FER_{Typical}$ (BER $\leq BER_{Typical}$) is satisfied.
Definition 3: $V_{min}^{FER_{Target}}$ $\left(V_{min}^{BER_{Target}}\right)$ is the minimum supply voltage for which FER $\leq FER_{Target}$ (BER $\leq BER_{Target}$) holds true.
Definition 4: $V_{best}^{(Energy/bit(x,y))}$ is the voltage corresponding to the minimum energy/bit value within the voltage range $[x,y]$, $x < y$.

Based on these definitions, the conservative approach required operating voltage at signal-to-noise ratio SNR_n denoted as $V_n^c(SNR_n)$ can be computed as $V_{best}^{Energy/bit\left(max\left(V_{min}^{FER_{Typical}},V_{min}^{BER_{Typical}}\right),V_{Typical}\right)}$. Similarly the aggressive approach operating voltage targeting FER_{Target} or BER_{Target} at signal-to-noise ratio SNR_n denoted as $V_n^{a_{FER}}(SNR_n)$ and $V_n^{a_{BER}}(SNR_n)$ can be calculated as $V_{best}^{Energy/bit\left(V_{min}^{FER_{Target}},V_{Typical}\right)}$ and $V_{best}^{Energy/bit\left(V_{min}^{BER_{Target}},V_{Typical}\right)}$, respectively.

For example, using the above equations, the conservative approach can operate the decoder at $V_n^c = 0.75$V preserving the performance (i.e., FER=0.498 and BER=0.041) of the original LDPC decoder operated at $V_{Typical} = 1$V. In this situation much less energy is consumed (i.e., 71% of energy saving) as illustrated in Figure 12. According to the aggressive scaling we adjust the power supply value to $V_n^a = 0.71$V while maintaining FER $\leq FER_{Target}$ (i.e., FER=0.0166), reducing the energy by 74% as illustrated in Figure 13. Note that to minimize the adaptation process energy overhead all of these computations are performed at design time and their results are placed in an LUT mapping SNR_n to V_n.

To sum up, although our conservative approach runs at lower supply voltage for saving energy, it insures of having an identical decoding quality like the one operated at standard voltage thanks to its endowed knowledge during design time and its smart adaptation at runtime. Awareness of its target performance is the added value of our aggressive approach to surpass the conservative one

in terms of energy efficiency at the cost of performance degradation while still satisfying the required error rate.

4 Evaluation

To evaluate our technique, we utilize the platform and LDPC decoder as in the pre-characterization stage augmented with the following energy reduction schemes: (i) powering off capability using Early Termination (**ET**) technique operated at the original supply voltage as presented in [9], (ii) a Hybrid Early Termination Scheme (**HS**) which includes the DVS technique in [8], (iii) our Conservative Approach (**CON**), (iv) our Aggressive Approach targeting FER (**AGF**), and (v) our Aggressive Approach targeting BER (**AGB**).
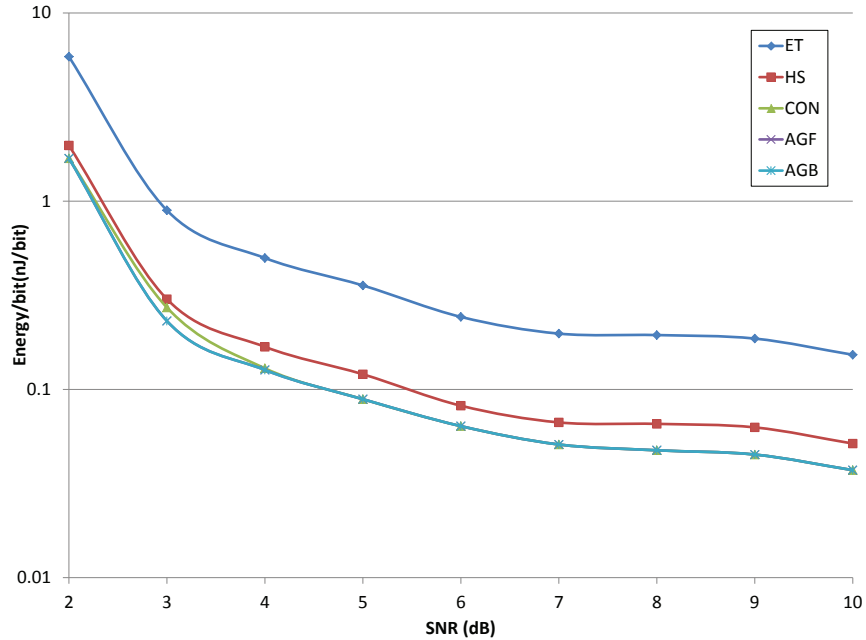

Fig. 14: Experimental Results

We evaluated the energy consumption of all the approaches when changing the channel SNR from 2dB to 10dB and the results are plotted in Figure 14. The energy/bit is obtained by accessing Power Supply Monitor and Controller inside the ML605 board through PMBus. Because AGF and AGB result in identical energy consumptions their plots are overlapped in the figure.

One can observe in Figure 14 that: (i) Regardless of SNR value ET always consumes more energy than the other approaches and this can be explained

by the fact that it has no capability to adapt to channel conditions. Energy consumed by ET decreases when the channel quality is getting better due to its early termination capability. At good channel quality, number of flipped bits decreases. Fewer flipped bits make decoding faster to converge and as a result, ET turns it power off earlier, reducing the consumed energy. (ii) Our technique always outperforms both ET and HS. However at low SNRs (2 to 3 dB) the energy reduction is limited (only 10-15% and 15-23% reductions over HS, for CON and AGF/AGB, respectively) by the fact that there is not that much excess performance to exploit. However, for less noisier channels, i.e., SNRs from 3 to 10 dB, more excess performance is available and CON achieves a 22-28% energy reduction over HS thanks to its adaptability to exploit channel noise variability; (iii) Because of its additional DVS technique, HS consumes 66% less energy than ET. CON (AGF/AGB) consumes around 71% (73%) and 76% (76%) less energy than ET, for bad and good channel quality, respectively; and (iv) At high SNR values CON, AGF, and AGB consume almost the same energy due to diminishing returns effect, while at low SNR values both AGF and AGB provide 15% energy reduction over CON.

We note that given that our technique does not alter the operating frequency it results in a better performance in terms of decoding throughput when compared to other decoders utilizing dynamic frequency scaling technique.

5 Conclusions

In this chapter, a technique towards energy effective LDPC decoding by exploiting channel noise variability to transform excess performance for energy saving was proposed. The proposed technique adaptively tunes the operating voltage on-the-fly based on the knowledge of the instantaneous channel noise level at runtime by utilizing its pre-characterization results, aiming to achieve the optimal tradeoff between decoder performance and power consumption, while fulfilling the QoS requirements. To demonstrate the capabilities of our proposal we implemented it and other state of the art energy reduction methods in the framework of a fully parallel LDPC decoder on a Virtex-6 FPGA. Based on the experimental results, when applied to a Sum-Product LDPC decoder our conservative approach improves energy efficiency by 71% to 76% and 15% to 28% compared to early termination without and with DVS, respectively, while preserving the decoder performance in terms of bit/frame error rate. With respect to the conservative version, our aggressive approach meets a given target error rate with up to 15% additional energy reduction thanks to its robustness-on-demand capability. In addition, since our technique maintains the operating frequency regardless of the utilized power supply voltage value, the frame throughput is preserved. Moreover, the FPGA-based experiments suggest that in certain conditions Degradation Stochastic Resonance occurs, i.e., the energy consumption is unexpectedly diminished due to the fact that unpredictable underpowered components facilitate rather than impede the decoding process.

Acknowledgment

References

1. Tse, D.: Fundamentals of wireless communication. Cambridge university press (2005)
2. Gallager, R.: Low-density parity-check codes. Information Theory, IRE Transactions on **8**(1) (January 1962) 21–28
3. MacKay, D.J., Neal, R.M.: Near shannon limit performance of low density parity check codes. Electronics letters **32**(18) (1996) 1645–1646
4. IEEE: The ieee 802.11n working group std. http://www.ieee802.org/11/
5. IEEE: The ieee 802.16 working group std. http://www.ieee802.org/16/
6. ETSI: The digital video broadcasting standard. https://www.dvb.org/standards/dvb-s2
7. IEEE: Ieee p802.3an (10gbase-t) task force. http://www.ieee802.org/3/an/
8. Chow, C., Tsui, L.S.M., Leong, P.W., Luk, W., Wilton, S.J.E.: Dynamic voltage scaling for commercial fpgas. In: Field-Programmable Technology, 2005. Proceedings. 2005 IEEE International Conference on. (Dec 2005) 173–180
9. Darabiha, A., Carusone, A., Kschischang, F.: Power reduction techniques for ldpc decoders. Solid-State Circuits, IEEE Journal of **43**(8) (Aug 2008) 1835–1845
10. Gammaitoni, L., Hänggi, P., Jung, P., Marchesoni, F.: Stochastic resonance. Rev. Mod. Phys. **70** (Jan 1998) 223–287
11. Aymerich, N., Cotofana, S., Rubio, A.: Degradation stochastic resonance (dsr) in ad-avg architectures. In: Nanotechnology (IEEE-NANO), 2012 12th IEEE Conference on. (Aug 2012) 1–4
12. Wang, W., Choi, G., Gunnam, K.: Low-power vlsi design of ldpc decoder using dvfs for awgn channels. In: VLSI Design, 2009 22nd International Conference on. (Jan 2009) 51–56
13. Zhang, X., Cai, F., Shi, C.J.: Low-power ldpc decoding based on iteration prediction. In: Circuits and Systems (ISCAS), 2012 IEEE International Symposium on. (May 2012) 3041–3044
14. Ahn, Y., Park, J.Y., Chung, K.S.: Dynamic voltage and frequency scaling scheme for an adaptive ldpc decoder using snr estimation. EURASIP J. Wireless Comm. and Networking **2013** (2013) 255
15. Kim, E., Shanbhag, N.: Energy-efficient ldpc decoders based on error-resiliency. In: Signal Processing Systems (SiPS), 2012 IEEE Workshop on. (Oct 2012) 149–154
16. Cho, J., Shanbhag, N., Sung, W.: Low-power implementation of a high-throughput ldpc decoder for ieee 802.11n standard. In: Signal Processing Systems, 2009. IEEE Workshop on. (Oct 2009) 040–045
17. Marconi, T., Spagnol, C., Popovici, E., Cotofana, S.: Towards energy effective ldpc decoding by exploiting channel noise variability. In: Very Large Scale Integration (VLSI-SoC), 2014 22nd International Conference on. (Oct 2014) 1–6
18. Johnson, S.J.: Introducing low-density parity-check codes. Technical report, Department of Electrical and Computer Engineering, University of Newcastle, Australia (2006)
19. Tanner, R.: A recursive approach to low complexity codes. Information Theory, IEEE Transactions on **27**(5) (Sep 1981) 533–547