

RAEF: A Power Normalized System-level Reliability Analysis and Estimation Framework

Rishad A. Shafik
Department of Computer Science
University of Bristol
Bristol, BS8 1UB
cstras@bris.ac.uk

Bashir M. Al-Hashimi
School of ECS
University of Southampton
Southampton, SO17 3AR
bmah@ecs.soton.ac.uk

Jimson Mathew, Dhiraj Pradhan
Department of Computer Science
University of Bristol
Bristol, BS8 1UB
jimson@cs.bris.ac.uk

Saraju Mohanty
Department of CSE
University of North Texas
Denton, Texas, USA
saraju.mohanty@unt.edu

Abstract—System-level reliability estimation is a crucial aspect in reliable design of embedded systems. Recently reported estimation techniques use separate measurements of power consumption and reliability to demonstrate the trade-offs between them. However, we will argue in this paper that such measurements cannot determine comparative reliability of system components with different power consumptions and hence a composite measurement in terms of reliability and power consumption is required. Underpinning this argument, we propose a system-level reliability analysis and estimation framework, RAEF, based on SystemC using a novel composite metric, power normalized reliability (PNR), defined as the ratio of reliability and power consumption. We show that PNR-based estimation enables insightful comparative reliability analysis among different system components. We evaluate the effectiveness of such estimation in RAEF using a case study of MPEG-2 video decoder with four processing cores considering single-event upset (SEU) based soft error model. Using this setup, we analyze and compare PNR based estimation with existing reliability evaluations at different system hierarchies. Furthermore, we demonstrate the advantages of RAEF in assessing design choices highlighting the impact of voltage scaling and architecture allocation.

I. INTRODUCTION

With continued technology scaling, device feature sizes are decreasing, making it possible to integrate more devices on a single chip. The reduced feature sizes have also enabled aggressive voltage scaling to reduce power consumption and to extend battery operating life. However, on the other hand, it is becoming harder to achieve reliability against different errors, particularly due to single-event upsets (SEUs) caused by electromagnetic radiations. This is because reduction of supply voltage causes an exponential increase in the number of errors induced [3]. Hence, there is a trade-off between power consumption and reliability, which has been investigated extensively [5], [12]. Due to this trade-off, reliability against soft errors is an emerging challenge in low power design of embedded systems [11].

A crucial aspect in reliable design of a system is the ability to accurately estimate the reliability of the system and its different components [2], [9]. Such estimation gives an insight into comparative reliability analysis of different components and identifies components that may require higher design efforts to incorporate error tolerance [1]. Over the years, researchers have proposed and empirically validated reliability estimation techniques at different design abstraction levels. Reliability estimation at circuit- or device-level [7], and at architectural-level [13], provide understanding of failure mechanism at lower-level abstraction. However, such estimation techniques are computationally extensive for complex circuits and do not capture high-level behaviour. An alternative is to employ simulation based estimation techniques at system-level design abstraction. Reliability estimation at this level captures high-level system behaviour and has multiple advantages as: (a) they are computationally less extensive and (b) they can be employed and repeated during the early design phase for evaluation [6].

To facilitate low power and high reliability based design optimization, recently a number of system-level reliability estimation techniques have been proposed. For example, Xiang *et al.* [11] showed a hierarchical reliability modeling framework for multi-processor system-on-chip (MPSoC). Their work considered reliability calculation at component-level first, followed by Monte Carlo simulations to estimate the system-level reliability in terms of mean time to failure (MTTF). Coskun *et al.* [4] reported an MPSoC reliability model based on statistical device-level model. Using such model, system-level reliability in terms of MPSoC lifetime was shown. Wu and Marculescu [15] proposed another reliability evaluation using mean error impact. Using such evaluation metric, reliability-aware power minimization technique was shown in [15]. A similar system-level design optimization technique was presented by Zhao *et al.* [16] using reliability function based estimation.

System-level estimation techniques reported in [4], [15], [11], [16] measure reliability and low power consumption separately showing that reliability exacerbates with reduced voltage settings. However, such measurement lacks insight as to how reliability compares among multiple components with different power consumptions. To understand the comparative reliability and design optimization requirements of system components, system-level reliability estimation need to consider reliability and power as a composite metric, which is the aim of this paper. In this paper, we propose a system-level reliability analysis and estimation framework (RAEF), using a novel composite metric, power normalized reliability (PNR), expressed as the ratio of component reliability and dynamic power. We show that, due to composite measurement, PNR based estimation enables insightful reliability analysis at various design hierarchies. The rest of the paper is organized as follows. Section II introduces PNR metric, while Section III details proposed framework RAEF using this metric at system-level. Section IV presents PNR-based experimental results underlining the advantages. Finally, Section V concludes the paper.

II. POWER NORMALIZED RELIABILITY

Existing reliability evaluations highlighting trade-offs with power consumption are carried out using separate measurements of reliability and power consumption [11]. Table I shows such reliability and power estimates of components in four processing cores with 128-bit registers each. The power (P) values (considering dynamic power only) are given by

$$P = \alpha C_L V_{dd}^2 f \quad , \quad (1)$$

where C_L is the processor load capacitance per cycle, V_{dd} is the supply voltage, f is the operating frequency and α is the switching activity factor ($0 \leq \alpha \leq 1$). The reliability values in Table I are found by

$$R = \exp[-\lambda v t] = \exp[-\lambda_0 k v t] \quad , \quad (2)$$

where t is the observation time (in seconds), v is the vulnerability factor and k is the factor by which nominal SER (SER at

nominal voltage settings), λ_0 , is increased due to V_{dd} scaling ($\lambda = \lambda_0 k$). The vulnerability factor (v) in (2) is related to the actual impact of errors and is defined as the ratio of the number of visible faults to the number of faults occurring over a given observation time [8]. From Table I it can be seen

Table I
RELIABILITY AND POWER ESTIMATES OF DIFFERENT COMPONENTS

Component	Scaling	Reliability	Power, mW
comp1	66.7MHz @ 0.55V	0.96	0.98
comp2	100MHz @ 0.6V	0.98	1.96
comp3	200MHz @ 1V	0.99	9.24

that, *comp1* gives reduced power consumptions due to lower voltage scaling ($V_{dd}=0.55V$ and $f=66.7MHz$). However, lower V_{dd} scaling also leads to degraded reliability (R) for *comp1* as SER value increases in (2). As expected, *comp2* and *comp3* give higher power consumptions and increased reliability due to higher voltage scaling ($V_{dd}=0.6V$, $f=100MHz$ for *comp2* and $V_{dd}=1V$, $f=200MHz$ for *comp3*) and reduced k . Comparing the reliability of these components, it can be seen that *comp3* gives the highest reliability among all components. However, since the component reliabilities are achieved at the cost of different power consumptions, such comparisons are clearly not justified when reliability and power consumption are considered jointly. Hence, for insightful reliability comparison and analysis of system components, composite measurement is much needed.

To jointly estimate and compare the reliability of different components of a system, we propose a novel composite metric, power normalized reliability (PNR), defined by the ratio of reliability (given by (2)) and power consumption (given by (1)) of a system component with equal weights, i.e.

$$PNR = \frac{R}{P} = \frac{\exp[-\lambda_0 k v t]}{\alpha C_L V_{dd}^2 f} \quad (3)$$

The PNR definition in (3) essentially gives a measure of component reliability for a given cost in terms of power consumption. As can be seen, PNR depends on the voltage scaling (through V_{dd} and f), processor activity factor (α), observation time (t) and the vulnerability factor (v) of the component. The vulnerability factor and observation time affect the reliability (defined in (2)), while voltage scaling and processor activity factor affect the power consumption (defined in (1)). Fig. 1 shows the impact of voltage scaling and processor activity on PNR. The power values in horizontal axis are evaluated through the product ($C_L V_{dd}^2 f$) in 1, while three different PNR values are evaluated in vertical axis through (3) assuming activity factors of $\alpha=1, 0.75$ and 0.5 . Two observations can be made from Fig. 1. First observation

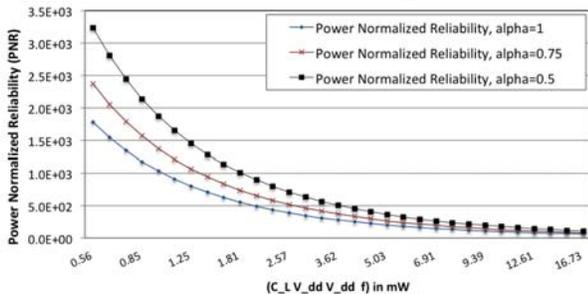


Figure 1. Power consumption versus PNR relationship

is that due to exponential variation in power consumption with voltage scaling, PNR is significantly high at lower voltage scaling. However, with higher voltage scaling power is increased and also reliability is improved due to reduction in the error rate in (2). This leads to less significant PNR variations at higher power consumption. Second observation is related to the fact that

with higher activity factor PNR is reduced as power consumption is increased linearly. From the above observations it is evident that PNR is an insightful metric for comparing the reliability per unit power consumption for a given voltage scaling. It can also be useful when comparing between components of different voltage scalings (Section IV shows the impact of voltage scaling on PNR based estimations).

III. PROPOSED FRAMEWORK: RAEF

System-level reliability estimation with power normalized reliability (PNR) metric (Section II) is carried out using proposed reliability analysis and estimation framework (RAEF). It is organized in three interconnected units: fault injection simulator, simulation monitor and PNR based reliability estimator (Fig. 2). Detailed description of each unit follows.

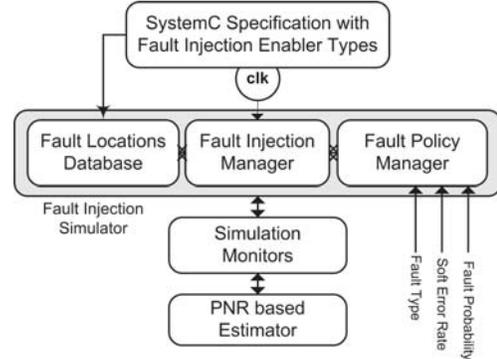


Figure 2. Block diagram of proposed framework, RAEF

A. Fault Injection Simulator

Fault injection simulator in RAEF, implemented through [10], is responsible for simulated fault injection into SystemC design specifications. Fault injection is initiated through replacement of variable or signal types in the original design specification to the equivalent fault injection (FI) enabler types (Fig. 2). These FI enabler types contain constructors (executed during module initialization) and destructors (executed when module scope expires) to automatically update the centralized fault locations database through insertion / deletion of their value holder addresses. The fault injections into the registers contained within the fault locations database are then carried out based on the policy specified by the fault policy manager, which takes user input of soft error rate (SER) and probability distribution of fault locations. Based on a given fault policy, the actual fault injection is carried out by fault injection manager through perturbation of registers in the fault locations database. To control the fault injection timing, system clock is connected to the fault injection manager (Fig. 2).

B. Simulation Monitors

Simulation monitors interact with the fault injection simulator (Section III-A) and capture simulation-specific information at register- and processing core-level. At register-level, the following simulation-specific information is logged over an observation time (t):

- 1) The total number of active cycles by i -th register in c -th processing core, $t_{i,c}^o$ (in clock cycles),
- 2) The total number of visible errors experienced, $\Gamma_{i,c}^V$, and the total number of errors injected, $\Gamma_{i,c}^A$ in i -th register of c -th processing core.

The above information are obtained through implementing simulation monitors in read, write, arithmetic and logical operator

definitions in the FI enabler types, which are used to specify the type of registers. In these operator functions, *setCycleBusy(..)* method in register-level monitor object, *regMonitor* contained in *Monitor* class, is used to update busy cycles in FI enabler types as shown below-

```
Monitor::regMonitor().setCycleBusy(this.pointer);
```

As can be seen, the current FI enabler type (i.e. register) address pointer is passed as a parameter of *setVisible(..)* to update the target register within the fault injection database. To determine and set visibility of errors, the following statements are used in the FI enabler type operation definitions-

```
tmp = this->old_value;
if(*this == tmp)
    Monitor::regMonitor().setVisible(this.pointer);
```

As can be seen, the visibility of injected errors is first determined by comparing the older register value (*old_value*, before injection of fault) with the post-injection register value (contained within *this*). If the outcomes are different, the fault in the register is marked as visible by *setVisible(..)* method with the register pointer address as the parameters. Due to operator based implementation of monitors with counters for each register, the simulation-specific information are updated automatically when they are used.

At processing core-level, simulation specific information are logged by the same monitors using a global counter for each processing core. The use of such global counters ensures that related register-level activities during the same clock cycle are updated only once. The following information are logged at processing core-level over an observation time (t):

- 1) The total busy cycles in c -th processing core, t_c^b .
- 2) The total number of visible errors experienced, Γ_c^V and the total number of actual errors injected, Γ_c^A by the c -th processing core.

The above simulation-specific information obtained through the simulation monitors are then used to analyze and estimate the reliability using PNR metric in the reliability estimator.

C. PNR based Reliability Estimator

The PNR based estimation in RAEF is carried out at three hierarchical levels: register-level, component-level and system-level. Register-level estimation shows how reliability of registers is affected due to lower-level perturbation, while processing core-level estimation underlines the impact on the reliability of a group of registers in a processing core. Finally, reliability estimation at system-level shows how overall reliability is affected for a given implementation. Such hierarchical estimation enables reliability analysis of system components and provides insight into their comparative design requirements for fault tolerance [11]. Reliability estimations at various levels are detailed in the following.

1) *PNR at Register-level:* We consider that a system S comprises of C voltage scalable processing cores, i.e. $s_c \in S$; $c = 1, 2, \dots, C$, where s_c is the c -th processing core of system S . Each processing core can be considered as a set of G_c registers in the reliability space, which are affected by the same voltage scaling. Consider SER expressed in terms of per bit failures in time (FIT) is λ_b (in faults per bit per cycle). For the i -th register (in c -th processing core) with size $g_{i,c}$, the effective SER experienced by the register, $\lambda_{i,c}$, is given by [8] as

$$\lambda_{i,c} = \sum_{b=1}^{g_{i,c}} \lambda_b k_c v_b, \quad (4)$$

where k_c is the factor by which λ_b increases due to voltage scaling on c -th processing core, v_b is the vulnerability of b -th bit. The vulnerability factor (v_b) in (4) is defined as the ratio of the number of visible faults to the number of faults occurring over a given observation time [8]. With the given SER per register ($\lambda_{i,c}$) in (4), the reliability of i -th register in c -th processing core, $R_{i,c}$, is given by (2) as

$$R_{i,c} = \exp[-\lambda_{i,c}t] = \exp[-g_{i,c}k_c\lambda_b t], \quad (5)$$

where t is the observation time in clock cycles (found through simulation monitors, Section III-B). From (5) it can be seen that for a given λ_b and observation time (t , in clock cycles), per register reliability ($R_{i,c}$) is affected by the register size and the voltage scaling applied on the processing core. Using the reliability definition in (5), PNR of i -th register in c -th processing core can be given by (3) as

$$PNR_{i,c} = \frac{R_{i,c}}{P_{i,c}} = \frac{\exp[-g_{i,c}k_c\lambda_b t]}{\frac{t_c^b}{t} C_L V_{dd}^2 f}, \quad (6)$$

where $P_{i,c}$ is the power consumption due to processor activity in the i -th register and can expressed by (1) as a function of t_c^b obtained from the simulation monitors (Section III-B). Considering the voltage scaling for a processing core does not change over a given observation time, the comparative PNRs of registers of a processing core show the impact of register size and processor activity factor in the registers on their reliability for a given power consumption (Section II).

2) *PNR at Processing Core-level:* At processing core-level, the SER experienced by c -th processing core, λ_c , is given as

$$\lambda_c = \sum_{i=1}^{G_c} \lambda_{i,c} v_{i,c}, \quad (7)$$

where G_c is the number of registers and $v_{i,c}$ is the vulnerability of i -th register in c -th processing core. The vulnerability at register-level ($v_{i,c}$) is found as the ratio of $\Gamma_{i,c}^V$ and $\Gamma_{i,c}^A$ obtained through simulation monitors (Section III-B), i.e. $v_{i,c} = \Gamma_{i,c}^V / \Gamma_{i,c}^A$. Using the $v_{i,c}$ value, the effective SER at processing core-level in (7) can be expressed as

$$\lambda_c = \sum_{i=1}^{G_c} \lambda_{i,c} (\Gamma_{i,c}^V / \Gamma_{i,c}^A). \quad (8)$$

With the effective SER in (8), reliability of the c -th component over the observation time t can be given by (2) as

$$R_c = \exp\left[-\sum_{i=1}^{G_c} \lambda_{i,c} (\Gamma_{i,c}^V / \Gamma_{i,c}^A) t\right]. \quad (9)$$

From (9), it evident that the reliability of a processing core will depend upon the number of registers used in the core, per register SER and the vulnerability at register-level. Using (9) in (2), the PNR of c -th processing core is given by

$$PNR_c = \frac{R_c}{P_c} = \frac{\exp\left[-\sum_{i=1}^{G_c} \lambda_{i,c} (\Gamma_{i,c}^V / \Gamma_{i,c}^A) t\right]}{\frac{t_c^b}{t} C_L V_{dd}^2 f}. \quad (10)$$

where P_c is the power consumption of the c -th processing core, found through (1) using t_c^b and t values obtained from simulation monitors (Section III-B). From (10) it can be seen that PNR_c depends on a number of factors: the number and size of component registers, the vulnerability of component registers, voltage scaling used and the activity factor of the processing core. For a given voltage scaling, the PNR estimations of different processing cores can give insightful reliability comparisons involving their activity and register usage.

3) *PNR at System-level*: At system-level the effective SER experienced, λ , is given as

$$\lambda = \sum_{c=1}^C \lambda_c v_c, \quad (11)$$

where v_c is the vulnerability factor of c -th processing core and is given as the ratio of Γ_c^V and Γ_c^A obtained from the simulation monitors (Section III-B), i.e. $v_c = \Gamma_c^V / \Gamma_c^A$. Equation (11) gives the effective overall SER experienced at system-level. Using (11), system-level PNR can be given as the ratio of system reliability and power consumption, i.e.

$$PNR = \exp \left[- \sum_{c=1}^C \lambda_c (\Gamma_c^V / \Gamma_c^A) t \right] / \sum_{c=1}^C P_c. \quad (12)$$

System-level PNR estimation in (12) depends on the reliability and power consumptions of the processing cores. For given voltage scalings on the MPSoC processing cores, achieving high *PNR* at system-level is desirable since it signifies high reliability for a given cost in terms of power consumption.

IV. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of RAEF using PNR based estimation (Section III), the PNR estimates at register-level and processing core-level are analyzed and compared with existing reliability evaluation using MPEG-2 video decoder as a case study. This is followed by investigation into the impact of voltage scaling, architecture allocation (allocation of number of processing cores) and system-level PNR comparisons among different applications. All experiments are carried out using single-event upset based fault model assuming an arbitrary SER (in terms of failures in time per bit) of $\lambda_b = 10^{-18}$ and Poisson's distribution of fault locations in RAEF (Section III-A).

A. Case Study: MPEG-2 Video Decoder

MPEG-2 video decoder constitutes a major component of current and future multiprocessor system-on-chip (MPSoC) applications and is chosen as a case study to evaluate the effectiveness of RAEF. Fig. 3 shows the block diagram of a MPEG-2 video decoder setup with four processing cores implemented in RAEF. SystemC behavioral modeling is used for decoder cores, while partitioning and allocation are performed arbitrarily to reflect MPSoC. Each processing core consists of an ARM7 processor,

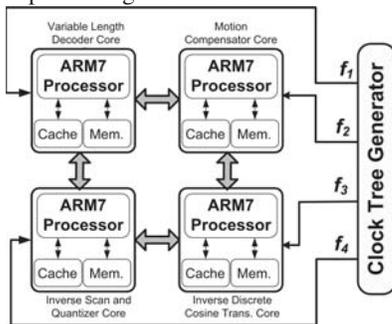


Figure 3. Block diagram of MPEG-2 video decoder used in RAEF data and instruction cache (8kbits and 16kbits), and private memory (512kbits). The cache and memory sizes have been chosen to provide high availability of data and parallelism among the processing cores. To minimize power, clock tree generator is used to feed different voltages and frequencies to the decoder processing cores (Fig. 3). Using such clock tree generator, two voltage scalings are used: $f=200$ MHz at $V_{dd}=1V$ (nominal voltage scaling) and $f=100$ MHz at $V_{dd}=0.6V$ (scaling by 2). All experiments using the decoder are carried out over an observation

time of 10 seconds, while decoding a flower sequence with 334 CIF frames (Source: <ftp://ftp.tek.com/tv/test/streams/Element/>).

To evaluate the effectiveness of PNR based composite measurement at register-level, Table II shows the reliability and power consumptions for registers in the MC decoder core (Fig. 3). For demonstration purposes, only 10 registers are included out of a total of 732 registers. The power consumption values are estimated for nominal voltage scaling (i.e. $V_{dd}=1V$ and $f=200MHz$), while reliability values are estimated in RAEF for an SER (in terms of failures in time per bit) of $\lambda_b=10^{-18}$ according to [11]. Columns 1 and 2 show the register names and their

Table II
RELIABILITY AND POWER ESTIMATES FOR TEN REGISTERS IN MC DECODER CORE (FIG. 3) AT NOMINAL VOLTAGE SETTING (200MHZ@1V)

Reg.	Size, g_i bit	Act. Cycl, $(t_{i,c}^b)$	Eff. SER, $\lambda_{i,c}$	Reliabil., $R_{i,c}$	Power, $P_{i,c}$ watts
reg1	1	1.1E+9	1.0E-16	0.99999	6.1E-3
reg2	8	6.2E+8	8.0E-16	0.99998	3.4E-3
reg3	8	1.2E+9	8.0E-16	0.99998	6.8E-3
reg4	16	6.1E+8	1.6E-15	0.99997	3.3E-3
reg5	16	5.5E+8	1.6E-15	0.99997	3.0E-3
reg6	32	1.4E+9	3.2E-15	0.99994	7.7E-3
reg7	32	9.5E+8	3.2E-15	0.99994	5.2E-3
reg8	64	5.4E+8	6.4E-15	0.99987	3.0E-3
reg9	128	9.6E+8	1.3E-14	0.99974	5.3E-3
reg10	128	5.9E+8	1.3E-14	0.99974	3.3E-3

sizes, columns 3 and 4 show total register activity cycles and the effective SER experienced by these registers ($\lambda_{i,c}$) and columns 5 and 6 show the reliability and power estimates for these registers (Table II). The register sizes and activity cycles per register are obtained through simulation monitors, while effective SER, reliability and power values are estimated using (4), (5) and (1) in RAEF. As can be seen, with increasing size of the registers, the effective SER experienced by the registers also increases. Due to such increase in SERs, the reliability over the given observation time (10 seconds, i.e. $t=200 \times 10^7$ clock cycles) decreases. Comparing the reliability estimates among all registers (column 5), it can be seen that *reg1* gives the best reliability among all registers. Comparing the power consumptions among all registers (column 6), it can be seen that *reg4* has the lowest power consumption. Clearly using such separate measurements of reliability and power consumption with different values do not give any insightful comparison when power consumption and reliability are considered jointly.

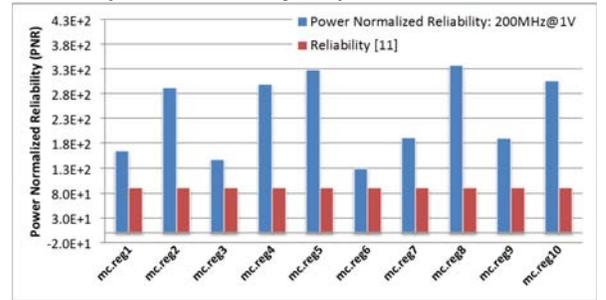


Figure 4. PNR for different registers of MC core (Fig. 3)

Fig. 4 shows the comparative PNR based estimates of different registers (Table II) found through (6) in RAEF. These PNR values are also compared with reliability values found in RAEF using (5). To give comparable scales for between PNR and reliability estimates, the reliability values (column 5, found by reliability function presented in [11]) are normalized by the maximum power consumption at nominal voltage settings (found using (1) with $\alpha=1$). As can be seen, although reliability values (Table II) show that *reg1* is the most reliable, the PNR estimates (Fig. 4) suggest that *reg8* is the most reliable for a given power

consumption. This is because *reg8* has much lower register activity and hence lower power consumption compared to *reg1*. Similar observations are also made with *reg4* and *reg5*. Despite having similar reliability estimates, PNR estimates suggest that *reg5* is more reliable for given power consumption due to lower register activity (Table II). As expected, due to higher power consumption and lower reliability, *reg3* and *reg6* have the lowest PNR among the registers. Due to lower power consumption and lower register activity, *reg5* and *reg8* give the best PNR.

Table III

RELIABILITY AND POWER ESTIMATES FOR PROCESSING CORES OF MPEG-2 VIDEO DECODER (FIG. 3)

Core	Act. Cyc., t_c^b	Eff. Vuln., v	Eff. SER	Reliab., R_c	Power, P_c (watts)
VLD	8.60E+8	4.86E-13	0.87	9.303E-1	4.73E-3
ISQ	1.14E+9	5.72E-13	0.85	9.286E-1	6.27E-3
IDCT	1.74E+9	1.59E-12	0.95	9.083E-1	9.57E-3
MC	1.44E+9	5.99E-13	0.91	9.280E-1	7.92E-3

To analyze PNR estimation at processing core-level, Table III shows the reliability (according to [11]) and power estimates of decoder processing cores (Fig. 3) using RAEF (Section III). Columns 1-3 show the processing core names, the number of active cycles and the effective processing core vulnerability factor of the processing core, while columns 4-6 show the effective SERs experienced, reliability and power estimates for the processing cores (Table III). The active cycles and effective vulnerability factor per processing core are obtained through simulation monitors, while the effective SER, reliability and power values are estimated using (8), (9) and (1) in RAEF. As can be seen, VLD core experiences the lowest effective SER when compared to the other processing cores. With such low SER, VLD core gives the best reliability among all processing cores. VLD core also experiences the lowest power consumption due to lower activity factor. On the other hand, IDCT core experiences the highest power consumption due to high core activity. In contrast to the VLD core, IDCT core experiences the lowest reliability due to high effective SER experienced. The ISQ and MC cores exhibit higher power consumption and lower reliability when compared with VLD core due to their higher effective SERs and activity factors (Table III).

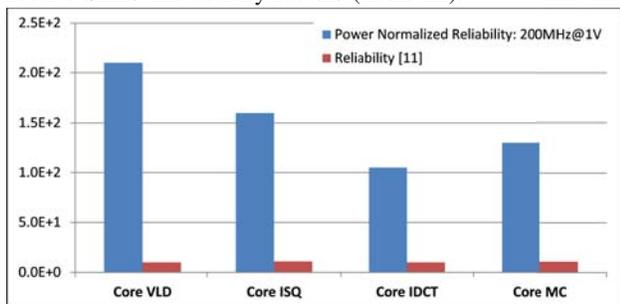


Figure 5. PNR estimates for different processing cores at nominal voltage setting (200MHz@1V) in MPEG-2 video decoder (Fig. 3)

Fig. 6 shows the comparative PNRs of the processing cores estimated using the reliability and power consumption values in Table III. As expected, due to high power consumption (Table III), IDCT core gives the lowest PNR among all processing cores. On the other hand, due to lower processor activity resulting in low power consumption (Table III), VLD core experiences the highest PNR among the processing cores. Fig. 5 also compares the PNR estimates with reliability values (defined by (9)). To give comparable scales for reliability estimates, the reliability values in column 3 are normalized by the maximum power consumption at nominal voltage settings (found through (1) with

$\alpha=1$). As can be seen, the reliability values do not show any significant variation among the processing cores. However, PNR based estimates clearly show that VLD core is most reliable and IDCT core is the most unreliable for a given power consumption. Due to composite measurement, PNR gives better insight in terms of reliability per unit power consumption.

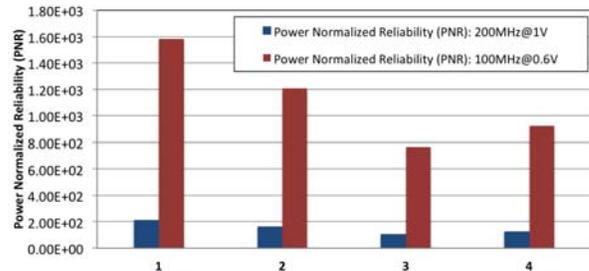


Figure 6. Impact of voltage scaling on PNR based estimation

To demonstrate the impact of power minimization through voltage scaling on PNR, Fig. 6 shows the PNR estimates of four processing cores of the decoder (Fig. 3) for two different voltage scalings (nominal voltage scaling and voltage scaling by 2). Comparing the PNR estimates of the same core at two different scalings, it can be seen that due to significant reduction in power consumption with voltage scaling by 2 ($V_{dd}=0.6V$ and $f=100MHz$), PNR is much higher (up to 7 times) when compared with the PNR at nominal voltage scaling ($V_{dd}=1V$ and $f=200MHz$). However, when the impact of voltage is de-emphasized by multiplying the PNR estimates with their respective maximum power consumptions for the given voltage scalings (using $\alpha=1$ in (1)), the result shows that the reliability per unit normalized power of lower voltage scaling is slightly lower due to increased SER and degraded reliability.

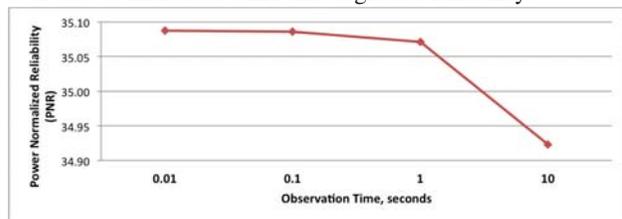


Figure 7. Impact of observation time on PNR estimation

B. System-level Reliability Comparisons

The PNR based analysis and estimation of MPEG-2 video decoder (Fig. 3) is further carried out at system-level. To examine the impact of observation time on PNR estimates at system-level, Fig. 7 shows the different PNR values estimated at nominal voltage scaling for the following observation times: 0.01, 0.1, 1 and 10 seconds (corresponding to $t=2 \times 10^6$, 2×10^7 , 2×10^8 and 2×10^9 clock cycles). The PNR values are estimated in RAEF using (12) with the information obtained from the simulation monitor (Section III-B). As can be seen, with increased observation times, PNR is gradually decreased. This is because reliability at system-level (defined by (12)) is degraded with increased observation times. Hence, PNR based estimations over considerably longer observation times will be dominated by poor reliability, while for shorter observation times it would be dominated by power consumption, as expected.

Architecture allocation is an important system-level design step that deals with the allocation of number of processing cores in an MPSoC architecture. The effect of architecture allocation on power consumption and reliability of an MPSoC has been investigated extensively [12]. To study the impact of

Table IV
TASK DISTRIBUTION OF MPEG-2 VIDEO DECODER AMONG CORES FOR DIFFERENT ARCHITECTURE ALLOCATIONS

Allocation	Core	Mapped Tasks
2 Cores	Core 1	variable length decoding & motion compensation
	Core 2	inv. scan, quantization & inv. disc. cos. transformation
3 Cores	Core 1	variable length decoding & inv. scan and quantization
	Core 2	motional compensation
	Core 3	inverse discrete cosine transformation
3 Cores	Core 1	variable length decoding
	Core 2	inverse scan and quantization
	Core 3	inverse discrete cosine transformation
	Core 4	motional compensation
5 Cores	Core 1	variable length decoding
	Core 2	inverse scan and quantization
	Core 3	discrete cosine transformation by row
	Core 4	discrete cosine transformation by column
	Core 5	motion compensation

architecture allocation (i.e. MPSoCs with varying sizes) on PNR based reliability estimates at system-level, Table IV first shows different architecture allocations of MPEG-2 decoder with 2, 3, 4 and 5 cores along with the mapped tasks per core. The application task mapping is carried out arbitrarily to reflect MPSoC. Fig. 8 shows the PNR estimates of the decoder with these four architecture allocations (Table IV). All PNR values are estimated at nominal voltage scaling ($V_{dd}=1V$ and $f=200MHz$). As can be seen, with higher architecture allocation, the PNR

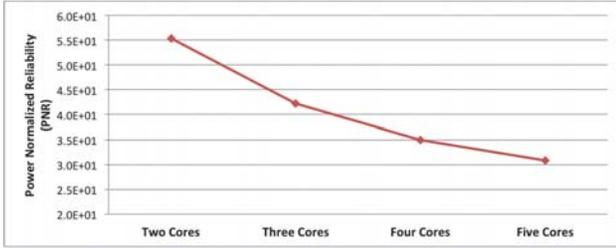


Figure 8. Impact of decoder architecture allocations on PNR based estimation values are decreased. This is because with more allocated cores in the decoder, power consumption of the decoder increases significantly. As a result of decreasing PNR values with higher architecture allocations, the decoder with 2 cores experiences the highest PNR, while the decoder with 5 cores experiences the lowest PNR when compared with the other cores.

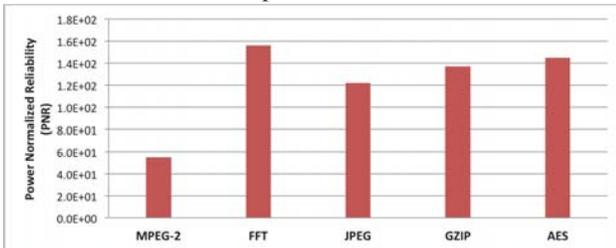


Figure 9. Comparative PNR estimates of different MPSoC applications
Reliability and power consumption of a system depends on the complexity and implementation of an application [14]. To evaluate the comparative system-level reliabilities of different applications, Fig. 9 shows the PNR estimates of five different applications: MPEG-2 video decoder, GZIP, AES, FFT and JPEG¹. These PNR values are estimated in RAEF using architecture allocation of 2 cores for each application at nominal voltage setting ($V_{dd}=1V$ and $f=200MHz$). As can be seen, MPEG-2 experiences the lowest PNR among all other applications. Lower PNR estimate in MPEG-2 is caused by two major factors. The first factor is related to higher activity factor due to highly parallel and computationally intensive nature of MPEG-2

¹Modified SystemC specifications are used. Sources of GZIP, JPEG and MPEG: <http://euler.slu.edu/~fritts/mediabench/>, AES and FFT: www.systemc.org

decoder processing [14], leading to higher power consumption. The second factor is related to higher register usage, which leads to higher effective SER and degraded overall reliability (given by (12)). Due to low activity factor and lower register usage (higher reliability), FFT application gives the best PNR among all applications.

V. CONCLUSIONS

In this paper, we proposed a system-level reliability estimation framework, RAEF (Reliability Analysis and Estimation Framework). The reliability estimation is carried out using a novel composite metric, power normalized reliability (PNR), expressed as the ratio of component reliability and power consumption. We showed that PNR is insightful for reliability comparison of system and its components as opposed to separate measurements reported to date, which fail to determine the comparative reliability of system components when power consumption is considered jointly. Using a case study of MPEG-2 video decoder, we evaluated the effectiveness of RAEF and showed the reliability analysis and estimation at different system hierarchies. Furthermore, we demonstrated the advantages of PNR based estimation in analyzing the design choices including voltage scaling, architecture allocation and system application. Incorporating various weights on power and reliability metrics, together with other system-level parameters, will be considered in future research.

REFERENCES

- [1] J. B. Bernstein *et al.* Electronic circuit reliability modeling. *Microelectronics and Reliability*, 46(12):1957 – 1979, 2006.
- [2] D. Brooks *et al.* Power, thermal, and reliability modeling in nanometer-scale microprocessors. *IEEE Micro*, 27:49–62, 2007.
- [3] V. Chandra and R. Aitken. Impact of voltage scaling on nanoscale SRAM reliability. In *Proc. DATE'09*, pages 387–392, 20–24 April 2009.
- [4] A. K. Coskun *et al.* Analysis and optimization of MPSoC reliability. *ASP Journal of Low Power Electronics*, pages 56–69, 2006.
- [5] F. Dabiri *et al.* Reliability-aware optimization for DVS-enabled real-time embedded systems. In *Proc. ISQED'08*, pages 780–783, USA, 2008.
- [6] A. Israr and S. A. Huss. Specification and design considerations for reliable embedded systems. In *Proc. DATE'08*, pages 1111–1116, NY, USA, 2008.
- [7] N. Miskov-Zivanov and D. Marculescu. Circuit reliability analysis using symbolic techniques. *IEEE Trans. on CAD*, 25(12):2638 –2649, Dec. 2006.
- [8] S. S. Mukherjee *et al.* A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor. In *Proc. 36th IEEE/ACM Int. Symposium on MicroArch.*, pages 29–40, Dec. 2003.
- [9] R. A. Sahner *et al.* *Performance and reliability analysis of computer systems: an example-based approach using the SHARPE software package*. Kluwer Academic Publishers, Norwell, MA, USA, 1996.
- [10] R. A. Shafik *et al.* SystemC-based minimum intrusive fault injection technique with improved fault representation. In *Proc. of IOLTS'08*, pages 99–104, Rhodes, Greece, July 2008.
- [11] Y. Xiang *et al.* System-level reliability modeling for MPSoCs. In *Proc. of CODES/ISSS'10*, pages 297–306, NY, USA, 2010.
- [12] T. Simunic *et al.* Optimization of reliability and power consumption in systems on a chip. In *Integrated Circuit and System Design*, pages 237–246, 2005.
- [13] J. Srinivasan *et al.* The case for lifetime reliability-aware microprocessors. In *Proc. of the 31st Int. Symposium on Computer Arch.*, ISCA '04, pages 276–, Washington, DC, USA, 2004.
- [14] K. Srinivasan *et al.* A technique for energy versus quality of service trade-off for MPEG-2 decoder. In *Proc. of IEEE CS Annual Symp. on VLSI*, pages 313–316, 19-20 Feb 2004.
- [15] K. Wu and D. Marculescu. Power-aware soft error hardening via selective voltage scaling. In *Proc. ICCD'08*, pages 301–306, 12–15 Oct. 2008.
- [16] B. Zhao *et al.* Reliability-aware dynamic voltage scaling for energy-constrained real-time embedded systems. In *Proc. of ICCD'08*, pages 633 –639, Oct. 2008.