Empirical Verification of Fault Models for FPGAs Operating in the Subcritical Voltage Region

Alex Birklykke*, Peter Koch*, Ramjee Prasad†, Lars Alminde‡, and Yannick Le Moullec*

*Technology Platform Section, †Center for TeleInfrastruktur
Department for Electronic Systems, Aalborg University
Fredrik Bajers Vej 7, DK-9220 Aalborg Ø
{alb, pk, prasad, ylm}@es.aau.dk
‡GomSpace APS, Niels Jernes Vej 10, DK-9220 Aalborg Ø
alminde@gomspace.com

Abstract—We present a rigorous empirical study of the bit-level error behavior of field programmable gate arrays operating in the subcritical voltage region. This region is of significant interest as voltage-scaling under normal circumstances is halted by the first occurrence of errors. However, accurate fault models might provide insight that would allow subcritical scaling by changing digital design practices or by simply accepting errors if possible. To facilitate further work in this direction, we present probabilistic error models that allow us to link error behavior with statistical properties of the binary signals, and based on a two-FPGA setup we experimentally verify the correctness of candidate models. For all experiments, the observed error rates exhibit a polynomial dependency on outcome probability of the binary inputs, which corresponds to the behavior predicted by the proposed timing error model. Furthermore, our results show that the fault mechanism is fully deterministic – mimicking temporary stuck-at errors. As a result, given knowledge about a given signal, errors are fully predictable in the subcritical voltage region.

I. INTRODUCTION

With approximately the same flexibility as software and performance as dedicated hardware, the field programmable gate array (FPGA) has become invaluable not only as a component in industrial products but equally as a platform to facilitate experimental research. One branch of research that has seen its widespread use is on low-power methodologies for CMOS architectures based on voltage-scaling. Specifically, the critical point at which voltage scaling results in errors has served as a pivot for much work: In [1] Chow et al. investigated a dynamic voltage scaling method which uses a monitoring scheme to ensure that the FPGA is operating as close to, but always above, the critical operating point (COP), i.e., the point where further scaling will result in observable errors. In this way, a power reduction between 4 and 54% was observed for a wide range of applications. Later, a similar approach was used with success on System-On-Chips [2]. FPGAs have also been used to investigate the limits of voltage scaling in more general settings: For example, in [3], Narayanan et al. confirmed the COP hypothesis [4] which states that for VLSI systems there exists a voltage $V_c$, frequency $f_c$, and temperature $T_c$ such that any scaling beyond the point $(V_c, f_c, T_c)$ will lead to system failure due to a massive number of simultaneous timing violations. The authors went on to show that by changing the slack distribution in the target microarchitecture the increase in error rate could be made more gradual. In [5], Roberts et al. went beyond the COP into the subcritical voltage region and captured the behavior of an 18x18 Xilinx DSP block multiplier with respect to supply voltage and error rate – showing a gradual increase in errors over an approx. $200$ mV range. The same line of work also helped gauge the power-saving potential of the Razor latch [6].

These examples underline the relevance of FPGAs in a research context and give a picture of the power-saving potential of FPGA-based implementations. Also, it indicates that much of the interesting work is done in the vicinity of the COP. However, although much insight can be found in the cited references regarding the fault mechanisms in voltage-scaled FPGAs, it has, to our knowledge, never been empirically illustrated and verified what happens at the bit-level. Roberts et al. [5] come close in the sense that they count the combined number of false positives and negatives on the 18-bit wide output of the multiplier given different voltage levels. This approach unfortunately hides the fault behavior at bit-level. Furthermore, the statistical properties of the test vectors are obscured due to a poorly described methodology, and the work
thus falls short at uncovering possible dependencies between the observed error-rates and statistical properties of the test data.

In this work, we therefore present statistical models for, and empirical verification of, bit-level fault models that describe the behavior of FPGAs operating in the sub-critical voltage region. Our goal is to add more detail to earlier work by exposing the interdependence between the statistical properties of the input signal and fault models.

Through our work, we gain a more accurate and confident understanding of how FPGAs fail during voltage-scaling. The experimental verification of fault models add further substance to assumptions made in earlier work (provided they hold) and help guide future research in the field by removing any doubt as to what happens when faults occur in over-scaled FPGAs. Finally, the probabilistic fault models constitute an effective primitive for error analysis of voltage-overscaled signal processing systems. Our work thus takes an important step towards an analytical evaluation of the accuracy-power trade-offs that exists for various DSP kernels.

The UUT behavior is characterized by the STALA unit by collecting bit-level error statistics, i.e., true and false positives, as well as negatives. The test vectors consist of independent binary random variables with adjustable outcome probability; each with a fixed length of $2^{41}$ bits. The randomness is sourced from a high quality random number generator based on the WELL1024a algorithm. The adjustable outputs and error statistics enable us to expose dependencies between error rates and the statistical properties of the test vectors as we operate the UUT in the subcritical voltage region. The models presented in Section IV describe the expected dependency between test vector properties and error rate for various fault models. Thus, by comparing the behavior of the voltage-scaled UUT to the behavior described by our models, it is possible to infer the most probable cause of the observed errors.

**III. UUT Architecture Analysis**

To design the experiments and identify possible fault mechanisms, it is necessary to analyse the FPGA technology to pin-point inherent limitations and technology features that might cause faults or failures in the UUT during subcritical voltage operation. Figure 2 is compiled based on information from Xilinx user guides [8] and patents [9], [10], and provides a general view of the path from input to output in an FPGA. In this case, we show the FPGA configured with a simple pipeline and differential signaling with the outside world.

First we note that FPGAs are multi-$V_{cc}$ devices, with separate voltage domain for IOs ($V_{ccio}$), core logic ($V_{ccint}$), and auxiliary circuits ($V_{ccaux}$) such as DCMs and JTAG components. This separation is practical since it allows us to isolate voltage scaling to the core logic, while running auxiliary and IO logic at nominal voltage levels. To ensure correct power-on and reliable behavior once the device is active, a voltage monitoring circuit is embedded in the FPGA. For Xilinx devices this is known as the power-on reset (POR) circuit. The POR insures that internal components are powered in the correct sequence during power-on. However, it also protects the user-application from faulty operation by resetting the device if the core voltage, which also supplies the SRAM, drops below a point where the configuration might get corrupted. Consequently, for voltage scaling operations, the POR trip-point constitutes a lower operational limit. For the Xilinx Spartan 3E (XC3S500E-PG208) used in our experiments, the trip-point is specified at a minimum of 400 mV with a nominal voltage of 1.2 V. But for the particular FPGA under test, the configuration is lost already at 500 mV, where the global reset is triggered. However, the DONE pin is upset at 550 mV, which at nominal levels indicates incorrect configuration. Subsequently, we chose 600 mV as the

Fig. 1. The experimental platform used to observe the effects of faults on binary signals when the UUT is running in the subcritical voltage region. The full specification of the platform can be found in [7].

**II. Experimental Setup**

The experiments in this work are performed using the two-FPGA setup illustrated in Figure 1. The top FPGA acts as a statistical logic analyzer (STALA) that generates test vectors and analyses the response from the unit under test (UUT). The UUT is configured with the desired circuit that we would like to characterize when subject to voltage and/or frequency scaling, and the
lower limit for our experiments to avoid corrupting the configuration while allowing potential fluctuations in the supply voltage. As the POR circuit prevents SRAM corruptions, we are guaranteed that the LUTs and routing will be configured correctly for all voltages above the POR limit. However, this does not imply that the LUTs and routing operate as expected; something we have to keep in mind when analysing the experimental results.

After the differential test vector in Figure 2 has been converted to single-ended, it is routed to the core logic through a step-down converter. However, since the logical HIGH-level in the IO block is sufficient to trigger the core logic which runs at a lower voltage, it is unclear whether the level-shifter from \( V_{ccio} \) to \( V_{ccint} \) actually contains any circuitry or is merely a direct connection. User guides and patent searches have not provided any information on this matter, but we assume that the circuitry needed is very limited. Similarly, we do not expect this transition to be the source of any faults. In contrast, the logical HIGH-level in the core logic is not sufficient to trigger logic running at higher supply voltages. As a result, the level-shifting from core to IO voltage levels requires inventive circuits, which is evident from the patents and papers found on the subject [9], [10], [11]. In relation to voltage scaling, the step-up transition might cause faults as lowering the core voltage potentially forces the level-shifter circuitry beyond its operational limits. In [1], such events where reported as IO errors. That is, errors caused by faults in the core-IO transition. Given the earlier reports of these types of faults, IO errors must be considered as a potential source of faults in our experiments. The last type of fault we might encounter is due to timing violations. It is well-known that voltage and propagation delay are correlated, hence, scaling the voltage might introduce errors due to timing faults. This is considered in more detail in Section IV, where a probabilistic model for both IO and timing errors will be introduced.

Generally, both the core logic and IO blocks are significantly more complex than depicted in Figure 2. However, to limit the scope of the paper and number of experimental factors, we restrict our focus to basic features of the FPGA i.e., basic IO capabilities, routing and LUTs. Hence, the \( V_{dd} \) characterization of e.g. RAM and DSP components is reserved for possible future work.

IV. PROBABILISTIC MODELING OF FAULTS

In the following, we introduce the mathematical framework necessary to link the statistical signal properties and fault models to the error rate measure. Figure 3 shows the graphical abstraction upon which we base our analysis. The basic idea is to separate the correct and incorrect circuit behavior into different components. The first component behaves as expected, whereas the second probabilistic components might add errors according to the fault scenario considered. In Figure 3, the faulty component is a latch and the (zero-delay) probabilistic component is denoted by \( \mathcal{F} \).

We characterize the behavior of \( \mathcal{F} \) using the transition probability matrix \( P_\mathcal{F} = [p_{y|x}(j|i)]_{i,j \in \mathcal{X}} \) where \( \mathcal{X} \) and \( \mathcal{Y} \) are the state space of the in and output, respectively, and \( p_{y|x}(j|i) \) is the probability of jumping to state \( i \) given the current state \( j \). In our
case, \(X, Y \in \{0, 1\}\). Thus, given a particular input state, say \(X_n = 0\), the output of \(\mathcal{F}\) will perform a probabilistic jump to a new state according to \(p_{Y|X}(Y_n = y|X_n = 0)\). The structure of \(P_\mathcal{F}\) depends on the fault model that we assume to affect the digital systems, and our challenge is to find a model for \(P_\mathcal{F}\) that accurately corresponds to a relevant fault-mechanism in the voltage-scaled UUT.

![Diagram of fault model](image)

Fig. 3. Bit-level faults are modeled as a noisy communication channel proceeding the circuit (in this case a latch) where the fault have occurred.

We model the binary signals on the input of \(\mathcal{F}\) as a stochastic process \(X = \{X_n : n \in \mathbb{Z}^+\}\) consisting of Bernoulli random variables with probability \(p = \Pr(X_n = 1)\). Furthermore, we assume that neighboring variables in \(X\) are linearly dependent. In this case, the temporal correlation coefficient is given by [12]:

\[
\rho = \frac{r_X(1)}{p(1-p)}
\]

where \(r_X(1) = \text{E}[X_{n-1}X_n]\) is the unnormalized lag-1 autocorrelation of \(X\). Note that for the independent case, \(r_X(1) = p^2\) and thus \(\rho = 0\). Bit-level correlation occurs as a result of correlation in physical signals which to some degree is maintained despite binary encoding [12]. However, as will be discussed later, faults also change the correlation coefficient of the signal which is why we include the property in our signal. As a result of the correlation, the joint distribution \(p_X(x_0, x_1) = \Pr(X_{n-1} = x_0, X_n = x_1)\) of the stochastic process \(X\) becomes:

\[
p_X(x_0, x_1) = \begin{cases} 
 00 & (1-p)^2 + \rho q \\
 01 & (1-p)q \\
 10 & (1-p)q \\
 11 & p^2 + \rho q
\end{cases}
\]

(1)

where \(q = p(1-p)\). Given the statistical properties of the input signal and a fault model \(\mathcal{F}\) defined by the transition probability matrix \(P_\mathcal{F}\), the error probability \(p_e = \Pr(X_n \neq Y_n)\) can be derived:

\[
p_e = \sum_{i \neq j} p_{Y|X}(j|i) = \sum_{i \neq j} p_X(i)p_{Y|X}(j|i)
\]

(2)

\[
= \Pr(Y_n = 1 \cup X_n = 0) + \Pr(Y_n = 0 \cup X_n = 1)
\]

False positives

False negatives

That is, the rate at which \(X_n = z\) does not imply \(Y_n = z\), given some input \(z \in X\). For binary experiments the error rate is composed by the probability of false positives and false negatives, which are the statistics captured by STALA. In the following, we derive the error rate according to different fault models that we expect to observe. Later we will compare error rate of these models with the actual error rate response obtained by STALA.

A. Threshold Effect Models

One theory on how errors occur in voltage over-scaled digital circuits is based on the threshold effect [13]. In this setting, the threshold voltage \(V_t\) of the CMOS circuit is considered a hard-limit. If the signal magnitude is above the threshold voltage, the circuit considers the signal as a digital ’1’ and if below as a digital ’0’. When the voltage difference between logical high and low grows sufficiently small, noise in the operational environment makes it harder to discriminate correctly between the states, and, eventually, logical misclassifications will occur. This behavior can be modeled by the general transition probability matrix \(P_\mathcal{G}\):

\[
P_\mathcal{G} = [p_{Y|X}(j|i)]_{i,j \in \{0, 1\}} = \begin{pmatrix}
0 & 1 - \alpha & \alpha \\
1 - \beta & \beta & 1 - \beta
\end{pmatrix}
\]

where \(\alpha\) denotes the probability of false positives and \(\beta\) false negatives. Using Eqn. 2, we find that the error rate of the threshold effect model is given by:

\[
p_{e, G} = p_X(0)p_{Y|X}(1|0) + p_X(1)p_{Y|X}(0|1)
\]

\[
= (1-p)\alpha + p\beta
\]

Two special cases of the threshold effect model are worth highlighting:

1) Symmetry, \(S\), \(\alpha = \beta\): This model reflects the scenario when the noise on the logical states is identically distributed and the threshold voltage is assumed always to equal half the supply voltage, ie. \(V_t = V_{dd}/2\). In this case, the error rate simply equals the transition probabilities:

\[
p_{e, S} = \alpha = \beta
\]
2) One-sided Asymmetry $A$, $\alpha = 0, \beta \geq 0$: If the threshold voltage does not scale with supply voltage but remains constant at some value, then it is very unlikely that the noise in the ‘0’ state will result in false positives once we scale the supply voltage. We approximate this by letting $\alpha = 0$. Instead, false negatives start to occur as the distance between $V_t$ and $V_{dd}$ becomes smaller, which increases $\beta$. The error rate for this model is:

$$p_{e,A} = p\beta$$

Hence, the error rate is correlated linearly with the bit-probability $p$ on the input.

Although the symmetric model has nothing to do with timing, it is widely used at higher levels of abstraction to model the occurrence of timing errors. Here, a common approach is to randomly flip bits in e.g. the MSB of output registers to evaluate the robustness of some program. Due to the widespread application of the model, we have included it in our analysis. In contrast, the asymmetric model mimics the behavior of faults occurring in the transition from core to IO voltage in the FPGA.

B. Timing Violation Model, $T$

It is well-known that reducing the supply voltage increases the latency in digital circuits. Hence, faults due to timing violations are very likely to cause errors. Figure 5 shows a pipeline section of a digital design, annotated with timing. The path slack quantifies the timing margin given by the period between the rising edges every $1/f_{clk}$ s. That is:

$$t_{slack} = (1/f_{clk}) - t_{cto} - t_{prop} - t_{su}$$

where $t_{cto}$ denotes the time between the rising edge of the clock and when the latched state is stable on the output, $t_{prop}$ the propagation delay of the combinational circuit between the latches, and $t_{su}$ the time (setup-time) required by the D-flip-flop to latch the signal correctly. If negative slack occurs, $t_{slack} < 0$, the combinational signal might be captured incorrectly and result in an error. Such events are known as transient faults and when the delay is large enough they result in temporary stuck-at errors [14]. We can model this behavior probabilistically with the transition matrix $P_T$:

$$P_T = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 
\end{pmatrix}$$

(3)

where $X$ denotes the event that $(X_{n-1} = x_0, X_n = x_1)$ and $Y = y$, the event that $Y_n = y$. Finally, $\alpha$ and $\beta$ denote the probability of transient faults on transitions from $1 \rightarrow 0$ and $0 \rightarrow 1$, respectively. By combining the transition matrix with the signal distribution in Eqn. 1, we can derive an expression for the error rate due to timing faults:

$$p_{e,T} = p_X(00)p_Y(0) + p_X(10)p_Y(1) + p_X(01)p_Y(01) + p_X(11)p_Y(01)$$

$$= 0 + p(1-p)(1-\alpha) + p(1-p)(1-\beta) + 0$$

$$= p(1-p)(1-\alpha + \beta)$$

(4)

Thus for timing errors, the error rate is linearly correlated with the correlation coefficient $\rho$ and exhibits a second order correlation with the bit-probability $p$.

Figure 4(a) shows the error rate relative to the bit-distribution $p = Pr(X_n = 1)$ for models $S$, $A$ and $T$. As evident, each model have a unique behavior, with the symmetric model being independent of $p$, the model asymmetric being linearly dependent, and the timing model exhibiting a polynomial dependency. Hence, by sweeping over different $p$ values during experimentation, we expect the error rate response to correlate with one or more of these fault models. We will use this property to verify the fault models experimentally.

V. EXPERIMENTS

In total three different experiments with various purposes are performed. The experiments are all variations to the setup in Figure 4(b), where some test remove certain latches, and others add additional logic in the UUT core. All experiments start with a factor screening experiment over a wide range of voltages and frequencies, and with a fixed input probability. These results expose the location of critical operation points (i.e., the point of transition between no errors and errors) and characterize the change in error rate once the voltage and frequency are scaled into the subcritical region. Based on the factor screening experiments, one or more points of interest in the subcritical region are singled out for more thorough analysis.
At these points, the bit-probability is swept from 0 to 1 and error statistics collected at each point. All experiments are randomized full factorial designs and each experiment is repeated 4 times to account for possible die temperature variations and other nuisance factors that might affect the experiments. The data points presented in the results section are the median of the 4 measurements. Once all data have been collected, the error rate responses are compared with the tendencies in Figure 4(a) whereby we can infer the most probable fault mechanism affecting the response.

A. Case 1 – Core-to-IO Transitions

This experiment aims at testing the transition from the core logic to the IO logic in the UUT. This is done by using the setup in Figure 4(b), but where latch A and B are removed and the input instead is connected directly to the output via the core logic. As a result, the test vector exercises the level shifter in the UUT output block (see Figure 2) which is the target of the investigation.

B. Case 2 – STALA-UUT Connection

Due to the large parasitic capacitance of the PCB traces relative to the routing in the UUT, the connection between the STALA and UUT FPGAs is associated with a significant propagation delay. Hence, the delay to and from the UUT is much greater than the delay in the UUT routing. That is, \( \min(t_{S \to U}, t_{U \to S}) \gg t_{core} \). Also, because the level shifter circuit in the output is more complex than in the input, an additional delay will be added, and thus we assume that \( t_{U \to S} > t_{S \to U} \). Consequently, in Figure 4(b), we expect to observe errors due to timing faults in the latches preceding the traces connecting the FPGAs before internal faults occur. That is, errors will occur in latch A and B prior to latch C, and in C prior to A.

C. Case 3 – Subcritical Operation of UUT Core Logic

The final experiment aims at characterizing the error behavior of the UUT core logic. To achieve this, the UUT is configured with an inverter chain to increase the propagation delay such that \( t_{core} > \max(t_{S \to U}, t_{U \to S}) \). Furthermore, to avoid timing errors in the STALA-UUT connection, the experiment is performed in the non-critical operating region of each inter-connection, whereby we are ensured that timing errors occur in the UUT and not in the inter-connections.

VI. Results

The results are shown in Figures 6 and 7. First we observe that for all points characterizing the subcritical region, except one, the error rate follows the polynomial tendency seen for timing faults in Figure 4(a). Also, the error probability curves in both cases 1 and 2 have a vertex at \((p, p_e) = (0.5, 0.25)\) with only one type of errors (false positive or negative) appearing. Using Eqn. 4 we see that this corresponds to a case where either \( \alpha \) or \( \beta \) is 1, depending on whether it is false positive or negatives appearing in the measurements. Finally, for case 3, the vertex of all curves are all at \((p, p_e) = (0.5, 0.5)\) which corresponds to a case where both \( \alpha \) and \( \beta \) is 1. When the transition matrix only contains ones and zeros it becomes a truth table, and the fault mechanism therefore is deterministic.

Considering Case 1 in Figure 6, we see that both false positives and negatives occur in equal proportions once the voltage is scaled sufficiently. However, prior to this, we observe a gray region which corresponds to the timing model in Eqn. 3 with \( \alpha = 1 \) and \( \beta = 0 \). That is, bits are temporary 'hanging' in the HIGH state during \( 1 \to 0 \) transitions, but not for \( 0 \to 1 \) transitions. This is a somewhat unexpected behavior, as transitions in CMOS circuits usually exhibits a balanced behavior (or "slow to rise" behavior.
if the capacitive load is large). However, we might explain this behavior due to the low threshold of the level shifter circuit which is in the signal path. As the IO voltage in the input makes a $1 \rightarrow 0$ transition, it must drop lower than normally before the output is triggered, which causes the false negative bias.

In case 2, we again observe a gray region, but in this case corresponding to false negatives. This behavior reflects the capacitive load of the PCB traces which increases the rise time of the digital signals. However, once we scale further a right skew appears on the error rate curve. By simulating the timing error model, we found that this happens when faults composite, i.e., when faults occur more than once along the signal path. The observed behavior corresponds to a case where a latch first produces false negatives (latch A), and when passed through a second faulty latch (latch C) with $\alpha = \beta = 1$ the skew appears in the error rate curve. This behavior is due to the fact that fault models add correlation to the test vector signal relative to the value of $p$, $\alpha$, and $\beta$. As the error rate of downstream streams depend on the correlation coefficient (see Eqn. 4), the error rate response of these might be skewed.

Figure 7 (left to right) shows the output of the inverter chain at logic levels 12, 14, 16 and 18. These results show that the error rate has a transient behavior at the bit-level in the core-logic of the UUT. A small gradient is hinted in some places by the gray points in the critical region, but the resolution is too low to draw any definite conclusion on whether false positive occur before negatives or vice versa.

VII. CONCLUSION

For all experiments, the observed error rates exhibit a polynomial dependency on bit-probability, which corresponds to the behavior predicted by the timing error model. This provides a strong indication that for FPGAs operating in the subcritical voltage region, the observed errors are caused by setup-time violations in latches. Furthermore, for the observed voltage and frequency ranges, we consider that the threshold models are falsified as the observed error rates and that predicted by the threshold models do not compare. However, we will not dismiss that the threshold models are relevant if voltage-scaling beyond the POR limit is somehow made possible. With the possible exception of a very narrow region around the COP, our results also indicate that the fault mechanism is deterministic if the fault rates belong to $\alpha, \beta \in \{0, 1\}$ for all cases. Interestingly, we observe that $\alpha = \beta = 1$ in the core logic during subcritical operation, which is equivalent to imposing an extra delay element after each latch affected by timing errors. This insight, as well as the proposed timing error model, enable an interesting outset for future research on subcritical operation of FPGAs.

Finally, it is worth commenting the generality of the results: FPGAs are complex devices and a considerable amount of hardware is activated even for simple HDL kernels. As a result, many factors influence the behavior during voltage scaling, which makes the FPGA an unlikely candidate to preserve idealized CMOS behavior. Hence, if we, as we do in the FPGA under test, identify a fault mechanism general to CMOS, then we consider it is safe to assume that the same fault mechanism will be the source of errors in many other CMOS devices subject to voltage scaling. However, some caution should be taken when comparing with other FPGA devices, as architectures differ and vendors continuously make changes here to. Hence, we recommend performing a p-value sweep to verify that timing faults are indeed still the dominant cause of errors when experimenting with other FPGAs.

REFERENCES

Fig. 6. Experimental results for Case 1 and 2. The top pictures represent factor screening experiments over a wide range of voltage and frequency levels, and the bottom are bit-probability sweeps for the points marked in the top pictures.

Case 3 – Subcritical Operation of UUT Core Logic

Fig. 7. Subcritical operation of an inverter chain. The upper row shows the error rate after inverter 12,14,16,18 for different voltages and frequencies. The bottom row shows sweeps over different input probabilities at selected points in the subcritical region for each inverter stage.