Exploiting Power Supply Ramp Rate for Calibrating Cell Strength in SRAM PUFs

Wendong Wang*, Adit Singh*, Ujjwal Guin*, Abhijit Chatterjee†
* Department of Electrical and Computer Engineering, Auburn University, Auburn AL 36849,
† School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta GA 30332
*{wzw0027, singhad, ujjwal.guin}@auburn.edu
†abhijit.chatterjee@ece.gatech.edu

***Abstract*—SRAM arrays are particularly attractive for use as physically unclonable functions (PUFs) because each manufactured copy of an SRAM array displays a different memory pattern when initially powered-on. This is due to random differences in device parameters in individual memory cells from manufacturing process variations. However, instability in the SRAM PUF response over the expected range of operating voltages and temperature, as well as environmental noise and aging degradation over time, is a challenge. Recent proposals aim at identifying a subset of all the cells in an SRAM, the most robust or strong cells, and using only these to construct a PUF. However, the manner in which the SRAM is powered up has been largely ignored in earlier work. We show that the SRAM power-up state is strongly dependent on the power supply ramp rate and direction; very different power-up states are obtained under different power-on scenarios. Furthermore, analyzing the power-up states under different ramp rates and directions can provide considerable insight into which transistor pairs in each individual cell are mismatched, and even the extent of the mismatch. Since such threshold voltage mismatch is key to cell power-on bias, we finally show how such experiments can be exploited to reliably identify the most robust strong cells in SRAMs for use in PUFs. These cells can be expected to generating reliable keys for cryptographic operations across a wide range of operating conditions, noise and device degradation.**

***Keywords*—SRAM PUF, ramp rate, stability.**

I. Introduction

Physically unclonable functions (PUFs) take advantage of the unavoidable small random manufacturing variations in highly scaled integrated circuits (ICs) for different cryptographic applications [1]–[4]. The key idea is to exploit some set of highly process sensitive but easily readable electrical signals from the circuit. These signals are selected such that they are extremely unlikely to be exactly the same even in two identically manufactured copies of the design due to the random process variations, and can therefore uniquely identify each manufactured IC. Observe however, that while each PUF's response is required to be different to uniquely identify individual copies of the design, for reliable cryptographic operations, the key generated from the PUF must consistently be the same, even under differing environmental and noise conditions for the same circuit. Achieving this latter requirement has proven to be challenging. While a large number of circuit designs have been proposed as PUFs [4]–[11], few have been found to be sufficiently robust over environmental variations such as temperature, supply voltage, electromagnetic interference, as well as device degradation from aging to be practical without extensive error management.

A particularly attractive design for PUFs is based on the static random access memory (SRAM) array. When an SRAM is initially powered up, each individual cell acquires a '0' or '1' logic value. Figure 1 shows the circuit schematic for a 6-transistor SRAM cell. Each cell has a pair of NMOS pull down transistors, a second pair of PMOS pull-up transistors, and a third pair of NMOS pass transistors connecting each of the two (complimentary) cell outputs to the bit lines. In an ideal SRAM cell, if each transistor pair as described above is identical in every respect, including layout associated parasitic components, then the cell is perfectly balanced. In the absence of any asymmetric electrical noise, such a cell has a random 50% chance of acquiring either a '0' or '1' state at power up. However, even a small imbalance within a pair of transistors can result in a cell being biased towards either a '0' or a '1' power-up state. In nanometer scale technologies, because of small random manufacturing variations, no two transistors in an SRAM cell are truly identical in practice. Consequently, virtually all the SRAM cells display some bias and generally acquire the same consistent logic value at power up. Thus each manufactured copy of an SRAM array operated as a PUF displays a different memory pattern when powered up because of the randomness in manufacturing process variations, and provides a unique response when the contents of the same address locations are queried. Unfortunately however, the power-up states in the cells are also impacted by transient electrical noise effects, and longer term shifts in device parameters from degradation and aging. Cells which by random chance, turn out to be nearly balanced with respect to device parameters at manufacture, can display inconsistent logic levels at power up, depending on environmental conditions such as temperature, supply voltage or electromagnetic noise. They may even change their response over time due to device degradation. In current designs such unstable "weak" cells are typically identified through repeated power up cycles, and the corresponding addresses masked out in defining the PUF response. Nevertheless, the stability of SRAM PUFs over the range of operating voltages and temperature, as well as aging effects, remains a challenge. It is estimated that even after masking out the known weak cells, $5\% - 10\%$ of SRAM PUF response bits may be in error in any individual response, with a much higher statistical worst case error rate in a high noise environment. Consequently, conventional error correcting codes are generally not be practical in overcoming this problem. More complex statistical solutions have been proposed to extract a stable signature from the response bits [3], [12]–[14], but this can be expensive and extremely challenging to implement.

A practical test technique that can reliably identify all the unstable weak cells in an SRAM PUF, over the full range of operating conditions including temperature, electrical noise, and aging would not require error correction. An early attempt to achieve such a test involved modifications to the cell circuitry [15]. More recently, an interesting approach exploiting cell "remembrance" has been presented to identify the strong cells [11]. Remembrance refers to attribute of an

978-1-5386-1472-3/18/$31.00

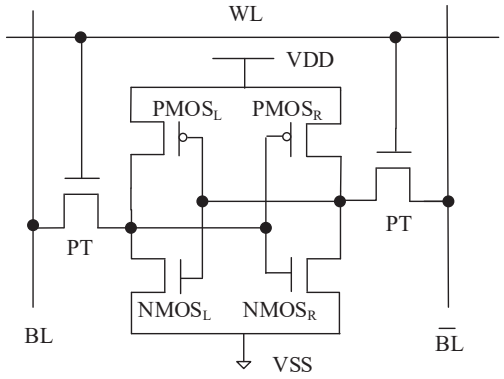

Fig. 1. A six-transistor SRAM cell.

SRAM "remembering" the previously stored values for a short while after power down. If power is restored quickly (within a few hundred milliseconds) after power down, many of the cells will be restored to their previous values. Cells that flip their contents would be those with a strong inherent bias to power up in the complementary logic state. In the aging experiments performed in [11], it has been shown that the strong SRAM cells identified using this remembrance based approach can be used to obtain a consistent PUF response over a reasonable operational life for the device.

While the practical approach exploiting cell remembrance for strong cell identification presented in [11] and discussed above appears to be effective for the specific aging experiments conducted in an experimental setting, it has not been analyzed to relate the observed power-up state of each cell to individual internal cell parameters. Specifically, the methodology does not attempt to identify parameter mismatches to specific pairs of transistors in individual cells, and obviously therefore, does not calibrate the extent of such a mismatch. In this paper, we show that more detailed understanding of parameter mismatches in individual transistor pairs within each cell is needed to reliably identify the strong cells under the full range of threats environmental noise and aging. Importantly, we show that the power supply ramp rate at power up (and critically also whether the power-on switch disconnects VDD or VSS) plays a key role in the SRAM power-up state. Indeed, entirely different power-up states can be obtained in the same SRAM under different power ramp scenarios. Some our key results show that if the SRAM is powered up by ramping VDD from low to high at a rapid rate, the power-up state of individual cells is decided entirely by the threshold voltage mismatch in the PMOS transistors in each cell. On the other hand, if SRAM is turned on by rapidly changing VSS from high to low (e.g., if the SRAM power-on transistor is connected to the VSS power rail) the power-up state of individual cells is decided entirely by the threshold voltage mismatch in the NMOS (pull down) transistors in each cell. For very slow power supply ramp rates, both PMOS and NMOS transistor pairs play a role in deciding the final power-up state of the cell, and surprisingly will equal weights independent of transistor sizing. Finally, we show in this paper how strong SRAM cells can be robustly identified by experiments exploiting different power supply ramp rates. To the best of our knowledge, the impact of power supply ramp rate on the power-up state of SRAMs has not been analyzed in earlier work.

II. Prior Research

PUFs have been extensively investigated as building blocks in hardware based authentication and other security features. For example, reliable PUF based key generation has been investigated in [16] for cryptographic applications achieving increased security levels at the cost of minimal added hardware. While our present work is focused on SRAM based PUFs, the arbiter PUF based on variability in path delays in the segments of a multi-stage circuit is another commonly utilized design. Recently, some FPGA based implementations, e.g., the Butterfly PUF circuit [17] have been shown to be quite stable under environmental variations; other FPGA based PUF designs utilize the FPGA logic and routing infrastructure for cost effectiveness [7], [18]. Attempts have also been made to control for the effects of noise, such as temperature, in ring oscillator based PUFs [8], which are otherwise also very cost effective in terms of hardware. Nevertheless, ensuring consistency in the PUF response over the entire range of operating conditions, expected environmental noise, and device degradation and wearout remains a challenge. For SRAM PUFs, preprocessing algorithms have been proposed to correct for the effects noise-induced bit-errors in the PUF responses [19]. For example, a design based on comparators and fuzzy extractors has been shown to be effective in making the SRAM PUF reliable, but at the cost of increased complexity of the circuit [10]. Mathematical models based on soft decision information have also been proposed to improve the efficiency and reliability with which keys can be extracted from SRAM responses [14] for authentication purposes.

The key factors that cause uncertainty in the power-up state of an SRAM array include transient electrical and electromagnetic effects from environmental noise, and shifts in the device parameters due to temperature, aging and wearout. It has been observed that some SRAM cells, with a strong inherent bias due to a significant mismatch in the transistor threshold voltage reliably power up to the same a '0' or '1' state, even in the presence of significant noise. On the other hand, weak SRAM cells, with closely matched transistor pairs, can be more susceptible to noise and parameter shifts from stress, and may display inconsistent states at power up. Fortunately, since SRAMs provide very high cell density, robust SRAM based PUFs can be constructed at reasonable cost by using only $5\% - 10\%$, or even fewer of the strongest individual SRAM cells in the PUF. Since as few as 128 random and reliable cells can be used to generate a secure on-chip key for a range of cryptographic operations, it is possible to use an SRAM PUF for on-chip key generation. The challenge clearly is to evaluate the strength of each individual SRAM cell, and reliably identify a set of very strong cells.

An early approach aimed at the identification of strong cells in SRAM arrays required a modification to the cell design and introduction of a second independent VSS (ground) rail. The source terminals of the two pull down NMOS transistors in each cell [15], are then connected to the two different ground rails: VSS(Left) and VSS(Right). By introducing a sufficient differential voltage between the two VSS lines, all the cells in the SRAM array can be externally biased to always power up to the same state. However, if the differential voltage applied is kept sufficiently small, a strong inherent bias in a cell in the opposite direction will result in the cell being powered up in a state contrary to the externally injected bias. Strong cells can thus be identified in this manner, with the strength of the cell reflected in the magnitude of the externally applied voltage differential overcome by the internal cell bias.

Most recently, the use of cell "remembrance" has been proposed to identify the strong cells [11] without any need for modification of conventions SRAM arrays. Remembrance refers to attribute of an SRAM "remembering" the previously

stored values for some window of time after power down. If power is restored quickly (within a few hundred milliseconds) after power down, many of the cells will be restored to their previous values. Cells that flip their logic values would be those with a strong inherent bias to power up in the complementary logic state. Thus, if the SRAM array is initialized with all '0' and briefly powered down before being restored, all cells that flip to a '1' at power up can be identified as strong '1' cells. In a similar manner, strong '0' cells can also be identified with the array initialized to all '1'. Furthermore, cell strength can be calibrated by the length of the power down window. aging experiments reported in [11], it have shown that the identified strong SRAMs cells can be used to obtain a reliable PUF response over a reasonable operational life for the device.

III. Dependence of power-up states on ramp rate

A key issue that has not been investigated in prior work on SRAM power-up states is the role of the power supply ramp rate and polarity (direction). In theory, the power supply to an SRAM can be turned on very quickly, for example raising the high voltage power rail from VSS (ground) to VDD in nanoseconds or less. It can also be raised very slowly, over several seconds. Observe that here fast and slow time ramp rates must be defined relative to the charging/discharging time constants of the internal capacitances at the circuit nodes of the SRAM. These can range from hundreds of picoseconds (ps) to hundreds of milliseconds (ms) depending on whether the nodes are being charged/discharge by actively conducting transistors or by extremely small leakage currents. In practice, while there are generally no lower limits on allowable slow power supply ramp rates, very fast power supply ramp can be limited by the drive strength needed to charge the large power rail capacitance. This is typically design limited, particularly if the SRAM power supply has to be switched on chip, as would be the case if the SRAM PUF is to be read while the system-on-chip (SoC) containing the PUF is operating.

Observe also that in general the SRAM can be powered up by ramping either the VDD or VSS power supply rails. The common assumption is that initially both circuit power rails, VDD and VSS, are at the same low (ground) potential, and the circuit is turned on by ramping up (raising) the VDD rail from ground to the high level power supply voltage. However, equivalently, both power rails VDD and VSS can also initially be at the high supply voltage, and the circuit can be turned on by ramping down the VSS rail to the low voltage (ground). In practice, the latter scenario would be encountered if the SRAM PUF power-on switch (within the SOC) is positioned in between the circuit and the VSS (ground) power rail.

Based on the above discussion, there are four corner cases for power on scenarios for SRAM memories. We assume that initially both VDD and VSS lines are at a common voltage potential. Power can be turned on by either (a) Ramping up the VDD power rail until the desired voltage differential between VDD and VSS is reached, or (b) Ramping down the VSS power rail until the desired voltage differential between VDD and VSS is reached. Additionally, for each of the above, the ramp rate can be fast or slow. Thus, we have (1) Fast VDD Ramp, (2) Fast VSS Ramp, (3) Slow VDD Ramp, and (4) Slow VSS Ramp. We next show, through detailed SPICE simulations, that the power-up state of an SRAM can be very different for these four different ways of powering up the SRAM. For the purpose of these simulation, the fast ramp linearly reaches full voltage over a 1 nanosecond (1 ns) period, while the slow ramp takes 1 second (s), a billion times longer.

We evaluate a 1024 cell (1K) SRAM array in this experiment, using the circuit in Figure 3 for an individual cell. Note that we have dropped the cell access transistors for the simulations, under the assumption that they are strongly off during power up. Any parasitic capacitance, such as the drain junction capacitance, from the access transistors is modeled as part of capacitances C1 and C2 in Figure 3. All simulations are performed in HSPICE for $32nm$ bulk technology (PTM). To account for the random process variations from manufacturing that critically determine the SRAM power-up states, all circuit parameters for individual components in the SRAM array have been randomly drawn from normal statistical distributions. The nominal value of the parameter is taken to be the mean of the distribution, and the standard distribution is appropriately chosen to realistically reflect typical manufacturing variability. In particular, to mimic manufacturing variations in the transistors, the threshold voltage parameter V_{TH0} in the technology model file of each individual MOSFET in the design has been changed to a value randomly drawn from a normal distribution with a nominal mean value and a standard distribution of 30mV. (Standard deviations in the range 10-50mV showed no change in the general trend of the results.) From the 32nm technology files, the NMOS nominal V_{th} is 0.42252V and the PMOS nominal V_{th} is -0.41174V. The threshold voltage and capacitance values as assigned above are used to generate the simulation model for on typical instance of the SRAM. Additional instances, reflecting typical process variations, can be created by randomly assigning a second set of parameter values from the same statistical distributions.

TABLE I. POWER UP CELL STATES FOR THE 1024 BIT SRAM UNDER THE FOUR POWER UP SCENARIOS.

No.of cells where all four scenarios have the same output	No.of cells where three scenarios have the same output	No.of cells where two scenarios have the same output
538	480	6

TABLE II. NUMBER OF MISMATCHES BETWEEN PAIRS OF POWER UP SCENARIOS.

Slow VDD Ramp Vs Slow VSS Ramp	Slow VDD Ramp Vs Fast VDD Ramp	Fast VDD Ramp Vs Fast VSS Ramp	Slow VSS Ramp Vs Fast VSS Ramp
0	250	470	219

A first set of experimental results are presented in Tables I and II. The simulation results in Table 1 show that only 538 cells out of the 1024 (1K) cells in the simulated SRAM consistently acquired the same logic values under all four power-up scenarios. Table II presents the same simulation results in a different way by comparing the number of mismatches between each pair of power up scenarios. It is interesting to note that there are no mismatches when the power supply is ramped slowly, whether the VDD rail is ramped up from VSS or if VSS is ramped down from VDD. However, approximately 25% of the 1024 cells acquire inconsistent values between fast and slow ramp rates for the same power rail. In case of fast VDD versus fast VSS ramps, fully 50% (half) of the cells acquire different states. While at this time we only highlight the fact that the power-up state is strongly dependent on the ramp rate and polarity at power up, following the analysis and observations in the later sections, it will be possible to explain why the percentage of cells in the four columns of Table II are, in fact, expected to be 0%, 25%, 50%, and 25% respectively.

IV. Simulation Experiments: Results and Discussion

In this section, we study and analyze several simulation experiments on how the power-up states of individual cells

are impacted by power supply ramp rate and direction. We particularly focus on mismatches in the threshold voltages in the pull up pair of transistors (PMOS), and the pull down pair (NMOS) in each cell (Figure 3), because these threshold voltage differences are known to mostly influence the power-up state. The power-on scenarios considered are once again the four discussed in the previous Section, although additional ramp rates are also studied.

In the first experiment we analyze the impact of NMOS and PMOS threshold voltage differentials on the power-up states of each of the 1K cells in the SRAM for a fast VDD ramp (VDD power rail ramp up over a $1ns$ period, with VSS held low). All circuit nodes were initialized to 0V at the start of the simulation. Results showing the relation between the power-up state and the NMOS and PMOS thresholds voltage differentials in each of the 1024 cells are displayed in the scatter plot of Figure 2(a). Here each red circle corresponds to a specific cell that powered up to a logic 1 state, while each blue triangle represents one that powered up to logic 0. The X-axis shows the NMOS threshold voltage differential $\Delta V_{thn} = V_{thn,R} - V_{thn,L}$. The Y-axis shows the PMOS threshold voltage differential $\Delta V_{thp} = V_{thp,R} - V_{thp,L}$. Notice in Figure 2(a) that all cells that power-up to the logic 0 state (red) lie above the X-axis in the scatter plot, and all cells that power-up to logic 1 (blue) are below the X-axis. This suggest that, for fast VDD ramp, the power-up state of a cell only depends on the PMOS threshold voltage differential in the cell, and is independent of the threshold voltages of the pull down NMOS transistors.

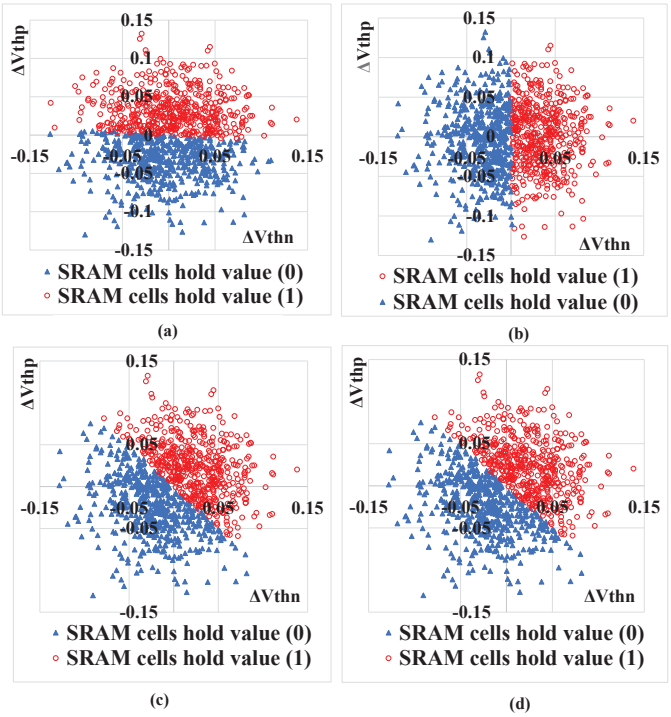

Fig. 2. Scatter diagram for 1K cells. (a) Fast ramp up VDD. (b) Fast ramp down VSS. (c) Slow ramp up VDD. (d) Slow ramp down VSS.

The above result can be predicted by simple analysis. Observe from Figure 3 that for a very rapid VDD rise at power-on, the increased voltage will initially all drop across the source and drain of the pull up PMOS transistors because the capacitances C1 and C2 will take time to charge. Therefore, with the gates of the two PMOS transistors initially held low because of this inertia of the two capacitances, both the PMOS transistors will turn-on strongly, and start charging their respective output capacitances. The PMOS with the smaller threshold voltage (in magnitude) will have the larger current drive, and will raise the voltage at its output faster. This will result in that output latching high when the bistable cell stabilizes, capturing a logic 1. Note that PMOS threshold voltages are negative. The side with the less negative (smaller in magnitude) threshold voltage has a positive threshold voltage differential ΔV_{thp}, with respect to the other PMOS. Therefore, a positive PMOS threshold voltage differential implies a logic 1 state at power-up, consistent with the observation from Figure 2 (a).

The scatter-plot in Figure 2 (b) shows the relation between the power-up state and the NMOS and PMOS threshold voltage differentials in each of the 1024 cells for a fast VSS ramp. Recall that here the SRAM is initialized with all nodes initially at the the VDD voltage; the VSS line is then ramped low over a short $1ns$ period. Figure 2 (b) shows that in this case the power-up state depends only on the NMOS threshold voltage differential and not on the PMOS threshold voltages. Again, this is to be expected based on the the discussion in the preceding paragraph.

Figures 2 (c) and (d) are scatter plots for slow VDD ramp and slow VSS ramp respectively. Notice that the plots are identical, indicating that identical power-up states are observed at the same memory cell locations for the two cases. This result was also presented earlier in Table II. Notice from the figures that the power-up state now is a function of both the PMOS and NMOS threshold voltage differentials -the line separating the logic 0 and logic 1 states is diagonal and not vertical or horizontal. In fact, the unity slope of this line suggests that the PMOS and NMOS threshold voltage differentials both have equal weights in deciding the power-up state. If both threshold voltage differentials bias the cell towards the same power-up state, then the cell will obviously acquire that state. This would be the case, for example, if the left PMOS had a smaller threshold voltage (in magnitude) compared to the right PMOS, biasing the left cell output high, and the left NMOS had a higher threshold voltage, re-enforcing the same bias. However, if the two threshold differentials conflict in biasing the cell, then the transistor pair with the bigger threshold voltage differential (in magnitude) decides the outcome. Notice that along the dividing line between the logic 0 an 1 states in Figures 2 (c) and (d), the magnitudes of ΔV_{thp} and ΔV_{thn} are exactly the same, although the signs are opposite, making the net $\Delta V_{thp} + \Delta V_{thn}$ equal to zero.

To understand why a slow VDD ramp and a slow VSS ramp, yields exactly the same power-up states, it is important to recognize that transient effects play no role whatsoever in determining the final logic levels in these power-up scenarios. When the power supply ramp is excessively slow, the circuit nodes are at equilibrium at all instants during the power-up. Virtually no current flows into the capacitances C1 and C2 (see Figure 3). Node voltages at all instances are defined only by steady state currents in the transistors, consequently circuit dynamics are not significant and it does not matter is the power supply is being ramped up or down. For perfectly matched PMOS and NMOS transistor pairs, and in the absence of any noise, in theory the voltages across the two capacitances would be always be equal during the entire duration of the slow power ramp. Any deviation is entirely due to differences among the transistor pairs. Note that for the purposes of our simulations, we have assumed that all manufacturing variations in the cells manifest as random threshold voltage variations, observing that to a variability in other features such as channel lengths etc.

also influence threshold voltages.

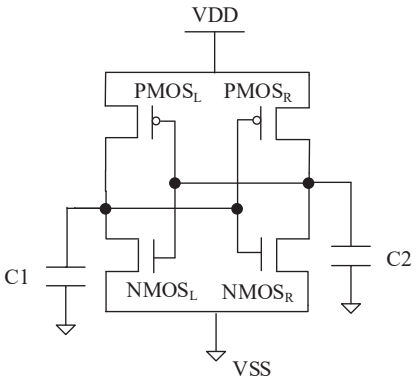

Fig. 3. The SRAM cell model used for simulation.

In the next experiment, we investigate the impact of varying power supply ramp rates between the fast and slow extremes that we have discussed thus far. It was observed in Table II that 250 (of the 1K) SRAM cells powered up to a different state for a slow $1s$ ramp when compared to the fast $1ns$ ramp. In Table III we also present results for ramp rates in between these two extremes. Observe that the largest number of power-up state changes are observed in the $10ns$-$10us$ range. No additional changes are observed as the power-up ramp is slowed down beyond $100us$, suggesting that our $1s$ window for the slow ramp is excessively conservative. Similar results are seen in Table III for VSS ramp. Note that while simulation indicates that a very few additional cell can change state in theory if even faster VDD/VSS ramps are employed, we have limited the fastest ramp period to $1ns$ because ramping VDD over picosecond windows is clearly impractical in practice.

TABLE III. NUMBER OF POWER-UP STATE MISMATCHES AT SLOWER RAMP RATES WHEN COMPARED TO FAST $1ns$ VDD AND VSS RAMPS.

Ramp Period	VDD ramp	VSS ramp
$1ns$	0	0
$10ns$	10	7
$100ns$	91	44
$1us$	200	148
$10us$	243	210
$100us$	250	219
$1ms$	250	219

Figure 4 explores the impact of variations in the design parameters for the SRAM cells. As transistor sizing in the cells is varied, or the load capacitance at the output is changed, the results for the power-up states are not significantly changed.

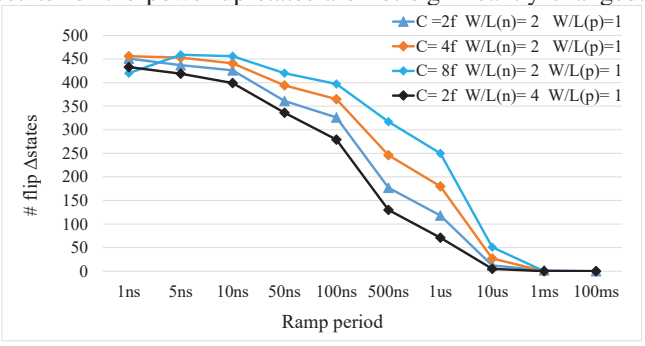

Fig. 4. Number of different power-on states between fast VDD ramp and fast VSS ramp.

V. PROPOSED METHOD TO IDENTIFY AND CALIBRATE STRONG SRAM CELLS

The goal of this study is to develop a reliable methodology for identifying the strongest cells in SRAMs for use in constructing PUFs. Given the strong dependence of SRAM cell power-up states to power supply ramp rates shown in the earlier sections, it is clear that any PUF application must employ a relatively slow power supply ramp to ensure a consistent response in the presence of variability and noise in the power-on circuitry. Using a VDD or VSS ramp period that is, say, one to two orders of magnitude larger than the minimum period where the SRAM response stabilizes and matches that from an arbitrarily slow ramp, can ensure that the PUF response will not be impacted by noise and aging degradation in the on-chip power-on circuitry associated with the PUF. For the sample simulations presented in Table III and Figure 4, an appropriate choice for the power-on period could be 5-10 ms. Using such a slow ramp for reading the PUF has other advantages. It can allow the use of a relatively small transistor switch with limited drive strength to power-up an on-chip PUF. Faster ramp rates require much larger switches (with low ON-resistances) to quickly charge the large power rail capacitances. Moreover, aging degradation in the switch resistance can alter power-up ramp rates and cause a change in the response as observed in Figure 4. Note that the SRAM response is particularly sensitive to ramp rates in the $50ns-1\mu s$ range, which may easily be experienced in practical situations if ramp rates and turn-on drive strengths are not carefully considered during design. This can increase errors in currently proposed PUF designs that completely ignore power-on .

To identify the stable cells in the SRAM and calibrate their relative strength, we propose running three different tests on the SRAM. The first two are fast VDD and VSS ramp tests. As shown earlier, the power-up states in these tests are primarily determined by the NMOS and PMOS threshold voltage differentials, respectively. By selecting cells that power-on in the same state for both these tests, we identify a set of cells where both the NMOS transistor pair, and the PMOS transistor pair, individually bias the cell to the same power-on state. Assuming random threshold voltage variations, this should be about 50% of the 1K SRAM cells, as was observed in the Fast VDD ramp versus Fast VSS ramp column in Table II. (The other entries showing mismatches between fast ramps and slow ramps in Table II are approximately 25% because in addition to the 50% of cells that are biased to the same state by both the PMOS and NMOS transistors, half of the remaining (50%) cells with conflicting biases also randomly power-on to the same state, leaving 25% in conflict.)

Our next challenge is to calibrate the individual strength of the candidate cells (about 50% of the total in a nominally unbiased design) that have a consistent bias from both the NMOS and PMOS transistors. For this we perform the following experiment. The SRAM is powered up and a logic 1 is written to all candidate cell locations. VDD is now slowly ramped down to a small voltage, e.g., 0.14V in the data in Table IV. Then VDD is slowly ramped up back again to its nominal value. While many cells will recover and retain their original logic 1 value when powered back on, some cells will flip to a logic 0. Clearly these cells have an inherent bias towards powering up in the opposite logic 0 state. Next the experiment is repeated again, ramping down slowly to a little higher voltage this time, e.g., 0.15V in Table IV, before ramping up again to the nominal voltage. In this instance, not all the cells that flipped at the 0.14V minimum will flip again, but a smaller subset comprising the more internally biased cells will still flip to a logic 0. This is because the externally programmed logic 1 bias is a little larger at the 0.15V minimum than at 0.14V. This set of (fewer) 0 bias cells

are clearly stronger in its inherent than those that flipped at the 0.14V minimum. In this manner, by repeating the experiment for multiple small increments in minimum VDD values, the the 0 biased cells can be calibrated into bins of approximately equal strengths. In Table IV, the most strongly biased cells are those that still flip when VDD was ramped down to only 0.22V. Similarly, cells inherently biased to logic 1 can be calibrated by initially writing logic 0's into all the cells.

TABLE IV. SIMULATION RESULTS FOR STRONG CELL CALIBRATION AND BINNING

VDD Minimum	Ave $\Delta NMOS$	Ave $\Delta PMOS$	$\Delta NMOS + \Delta PMOS$	No. of selected cells
0.14	0.044144	0.042454	0.0869831	356
0.15	0.04819	0.046113	0.094303438	253
0.16	0.051977	0.04992	0.1018969	198
0.17	0.054246	0.055269	0.109515104	139
0.18	0.059512	0.060875	0.12038708	93
0.19	0.066556	0.06874	0.135296	51
0.2	0.066324	0.07552	0.1418435	34
0.21	0.074835	0.083071	0.15790579	17
0.22	0.079643	0.083429	0.1630713	7

Table IV shows the number of cells in each bin that display varying bias strengths. Additionally, the magnitude of average mismatch in the NMOS threshold voltages $\Delta NMOS$ and PMOS threshold voltages $\Delta PMOS$ for all the cells in the bin, and their sum, is also shown. Recall that both the NMOS and PMOS transistors in all our candidate strong cells are biased in the same direction. Also, earlier the simulation results in Figure 2 showed that the NMOS and PMOS transistor differentials are equally weighted in determining the power-on state for slow ramp. Therefore it is not surprising that the bin strengths in Table IV correlate well with the sum of the average NMOS and PMOS threshold voltage differentials. The number of cells in the strongest bins is clearly a function of the threshold voltage distribution due to process variations. As the standard deviation of this variation goes down, there will be fewer strong cells in the tail of the distribution with a large mismatch in threshold voltages. While correlating the bias strength of cells in the bins in Table IV against actual noise and aging measures remains the subject of future work, our methodology can readily allow the selection of any desired number of the strongest cells in an SRAM for use in PUFs.

VI. CONCLUSION

While SRAM arrays are particularly attractive for use as PUFs, errors in the PUF response due to instability caused by voltage, temperature, environmental noise, and degradation due to aging is a challenge. Recent proposals aim at identifying a subset of all the cells in an SRAM, the strongest, most robust cells, and using only these to construct a PUF. However, we show here that the SRAM power-up state is also strongly dependent on the power supply ramp rate and direction. Very different SRAM states are obtained under different power-up scenarios; these must therefore also be critically considered in PUF construction. We also show that analyzing the power-up states under different ramp rates and directions can provide considerable insight into which transistor pairs in each individual cell are mismatched, and even the extent of the mismatch. Since threshold voltage mismatch is key to cell power-on bias, we show how such experiments can be exploited to reliably identify the most robust strong cells in SRAMs for use in PUFs. These cells can be expected to generating reliable keys for cryptographic operations across a wide range of operating conditions, noise and device degradation.

VII. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under Grant No. CCF-1527049.

REFERENCES

[1] G. Clarke, D. Van Dijk, and S. Devadas, "Controlled physical random functions," *Proceedings. 18th Annual*, pp. 149–160, 2002.

[2] U. Ruhrmair and D. E. Holcomb, "Pufs at a glance," in *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*. IEEE, 2014, pp. 1–6.

[3] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Helper data algorithms for puf-based key generation: Overview and analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 6, pp. 889–902, 2015.

[4] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[5] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 2002, pp. 148–160.

[6] G. Suh and S. Devadas, "Physical Unclonable Functions for device authentication and secret key generation," in *Proc. of ACM/IEEE on Design Automation Conference*, 2007, pp. 9–14.

[7] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "Physical unclonable functions and public-key crypto for fpga ip protection," in *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*. IEEE, 2007, pp. 189–195.

[8] C.-E. Yin and G. Qu, "Temperature-aware cooperative ring oscillator puf," in *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on*. IEEE, 2009, pp. 36–42.

[9] K. Xiao, M. T. Rahman, D. Forte, Y. Huang, M. Su, and M. Tehranipoor, "Bit selection algorithm suitable for high-volume production of sram-puf," in *Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 101–106.

[10] E. I. Vatajelu, G. Di Natale, and P. Prinetto, "Towards a highly reliable sram-based pufs," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016*. IEEE, 2016, pp. 273–276.

[11] M. Liu, C. Zhou, Q. Tang, K. K. Parhi, and C. H. Kim, "A data remanence based approach to generate 100% stable keys from an sram physical unclonable function," in *Low Power Electronics and Design (ISLPED, 2017 IEEE/ACM International Symposium on*. IEEE, 2017, pp. 1–6.

[12] C. Bösch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient helper data key extractor on fpgas," *Cryptographic Hardware and Embedded Systems–CHES 2008*, pp. 181–197, 2008.

[13] R. Maes, P. Tuyls, and I. Verbauwhede, "Low-overhead implementation of a soft decision helper data algorithm for sram pufs." in *CHES*, vol. 9. Springer, 2009, pp. 332–347.

[14] R. Maes, P. Tuyls, and I. Verbauwhede, "A soft decision helper data algorithm for sram pufs," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*. IEEE, 2009, pp. 2101–2105.

[15] S. Pandey, S. Deyati, A. Singh, and A. Chatterjee, "Noise-resilient sram physically unclonable function design for security," in *Asian Test Symposium (ATS), 2016 IEEE 25th*. IEEE, 2016, pp. 55–60.

[16] Z. Paral and S. Devadas, "Reliable and efficient puf-based key generation using pattern matching," in *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 128–133.

[17] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "The butterfly puf protecting ip on every fpga," in *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*. IEEE, 2008, pp. 67–70.

[18] J. H. Anderson, "A puf design for secure fpga-based embedded systems," in *Proceedings of the 2010 Asia and South Pacific Design Automation Conference*. IEEE Press, 2010, pp. 1–6.

[19] S. Eiroa, J. Castro, M. C. Martínez-Rodríguez, E. Tena, P. Brox, and I. Baturone, "Reducing bit flipping problems in sram physical unclonable functions for chip identification," in *Electronics, Circuits and Systems (ICECS), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 392–395.