

Challenges in SET Measurement – Working Draft

R. Shuler – April 3, 2009

Background and Importance of SET Measurement

Single Event Transients (SETs), the temporary response of a circuit to ionization radiation, are the ultimate cause of Single Event Upsets (SEUs), and thus errors. An SEU in a flip flop may result from an SET within the flip flop which makes it around the feedback loop to re-inforce itself, or an SEU may result from an SET in combinational logic near a clock edge being captured by a flip flop. Thus mitigation of Single Event Effects (SEE) boils down to understanding and mitigating SETs.

Mitigating SETs is aided by knowledge of their distribution in time (their length and arrival rates), in space (what nodes or combinations of nodes they hit), and how they vary in response to circuit and layout techniques, such as separation, interdigitation, and guard structures. For example, knowing length and arrival rates allow estimates of how many SETs will occur coincident with a clock edge, or provision for circuits to ignore pulses of typical SET lengths. Knowing distribution in space and the effect of layout techniques allow mitigation through sometime relatively simple changes in layout.

Early SET measurement consisted of “tuned flip flops” which measured one length at a time [need ref]. An improved scheme consisted of a series of “capture latches” [Narasimham] that freeze an SET as it propagates through a chain of fast latches which “freeze” the resulting pulse image when an SET is detected. It is then loaded into a shift register for output. This method captures the whole spectrum of SET lengths. Tradeoffs exist in the length of the chain, since short pulses will attenuate as they go through the chain. Resolution is also limited by the stage delay in the latch chain. An improvement to the capture latch system provides triggering on the trailing edge of the pulse, at the beginning of the chain, instead of the leading edge of the pulse at the end of the chain, and detects the trigger before entering the latch chain using logic gates which are faster than latches. This allows detection of shorter SETs which might otherwise be absorbed by the latch chain [Shuler]. Other efforts to measure SETs included storing the total charge of an SET on a capacitor, such as the gate of a FET [need reference]. This techniques requires analog to digital conversion and is more problematic to implement and calibrate in each new technology.

SET Measurement via Capture Latches

Difficulties Measuring Short SETs

The shortest pulse that could be captured in latches was 5 or 6 gate delays long. By “gate delay” here we mean a minimum delay such as a single-loaded inverter delay, i.e. the delay through one stage of a chain of inverters. Latch stage delays are usually a good deal longer.

Studies involving chips given a high total dose, which suppresses short pulses by unbalancing gates [Balasubramanian], showed that short pulses account a large percentage of SETs that translate into SEUs. Further improvement could be useful in studying the distribution of SETs, their origin in different types of logic, and the means of their mitigation.

Two problems arise in measuring short pulses. One is the collection of sufficient pulses to measure, since they do not propagate through long chains of logic typically used as “ion collectors” [Shuler, Narasimham]. This problem is partly alleviated by the fact that many more short SETs arise in circuits than long ones, so ion collectors do not have to be as large.

The second problem is that as long as the measurement uses the same technology as the circuits being measured, it will not be able to measure the limiting cases of that technology. Capture latches are essentially ordinary latches. The pulse measurement circuit requires a propagating pulse to trigger many capture latches, whereas in an application logic function, an SET need trigger only one latch (or flip flop) to cause an error.

When the problem is stated in terms of a technology examining itself, the solution obviously requires a different technology. When the problem is stated in terms of requiring a short pulse to propagate through a long chain of latches, the solution is to remove that requirement.

Dynamic Capture Latches for Measuring Short SETs

A faster technology than ordinary static CMOS exists and is easy to implement. It is called Dynamic Logic. One possibility is simply to replace the capture latches with dynamic latches. Many types of dynamic latches exist. We rule out the ones using pass gates on the ground that they are likely slower and might absorb the shortest pulses. Since we were familiar with the clocked (or current starved) inverter circuit from the many applications in which it appears [Loveless, Anith] and from its similarity to a commonly used RHBD circuit, the Guard Gate [Bhuva] or TAG [Shuler], we used it in a first design for a dynamic logic capture latch, Fig. 1.

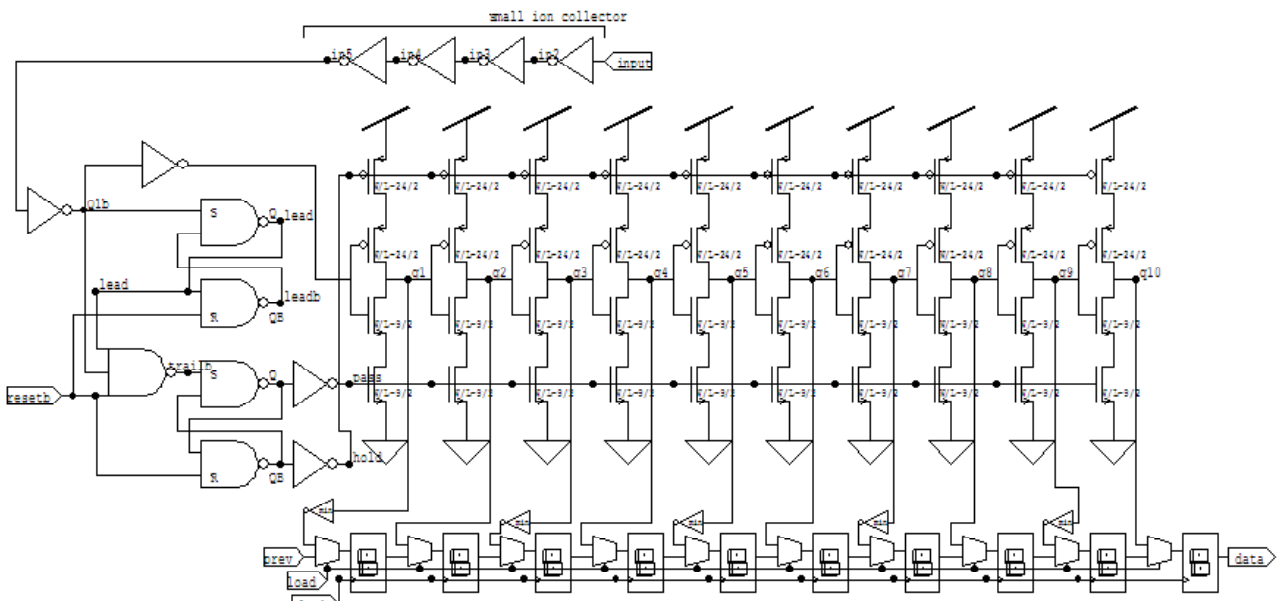


Fig. 1: Pulse capture circuit with dynamic latches

Only ten capture stages are shown for clarity. A short 4 stage ion collector is shown at the top, also for clarity as this is not a realistic length to collect many SETs. The shift register for reading out data shown at the bottom. DICE [need ref] flip flops are used to minimize errors in the readout circuit. Every other bit is inverted to make data interpretation easier (the capture latches invert every other bit). The trailing edge trigger technique is used for highest performance. So the circuit is a very conventional pulse capture circuit design, only differing in the details of the latch design, with data dynamically stored by the capacitance on each node in the capture chain.

This design provided a 25% improvement in interstage delay in the capture latches, over prior schemes with conventional latches. This translates into improved resolution of pulse lengths. Since the trigger

mechanism does not involve the latches, the minimum SET length that will trigger the circuit is unchanged. Overall the improvement seems only modest. A new problem is introduced because of the nature of the dynamic latches. If the trigger occurs while one is transitioning, and this will usually be the case, it will be frozen at an intermediate voltage. Because it is difficult to arrange for the inverted freeze signals “hold” and “pass” to be simultaneous, the latches may disagree with one another on the exact end of the pulse. The intermediate freeze state could be exploited by means of an analog to digital converter. This amounts to the same scheme as storing the SET on a capacitor, though perhaps with much greater resolution, since several stage delay “quanta” have been factored out of the SET width before the final stage where an analog value is stored on the stage’s output node.

Latchless Dynamic Logic Measurement of SETs

The triggering mechanism of the above scheme still uses conventional CMOS, and so is no faster than the signals being measured. Only the latches are better. Further, the triggering mechanism uses SET-RESET (SR) flip flops, which are slower than individual logic gates. The fastest technology we have available in CMOS is dynamic logic *gates* [need ref]. It would be desirable to implement the trigger using largely dynamic logic. It is tempting also to use dynamic gates to implement the timing propagation mechanism.

With dynamic logic, there is no restoration to the initial state of a gate until a precharge cycle. The gate makes one transition and will not go back. Thus the SET pulse will not propagate in the normal sense (both leading and trailing transitions) through a chain of dynamic logic gates. Earlier we noted propagation of the pulse was one of the limitations on measuring short pulses. If the dynamic gates could be used for timing only, their speed and resolution could be applied directly to the measurement problem.

Using the leading edge trigger method, the trailing edge of the pulse would be lost. However, using the trailing edge triggering mechanism, if the propagation through the dynamic logic is “frozen” on the trailing edge, then the leading edge is retained in the frozen gate states. It can be compared to the trailing edge, whose position is known if the delay of the trigger circuit is known. This delay can be determined by simulation or by a test circuit.

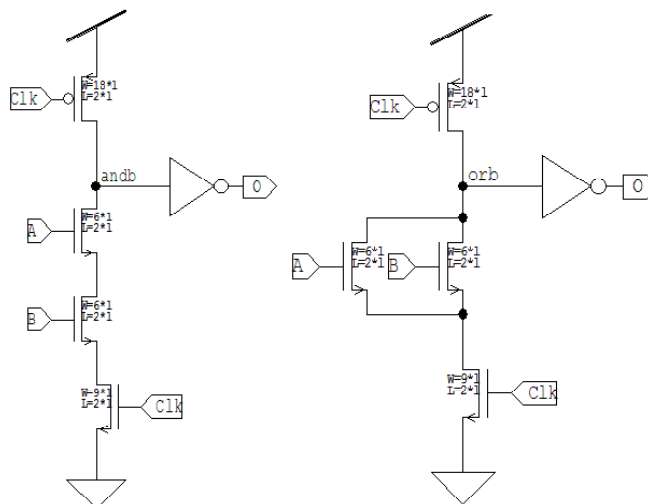


Figure 2: Domino Logic AND (left) and OR (right) gates

The NAND and NOR circuits shown in Figure 2 are dynamic gates, provided with a clock input. They are of the Domino variety [ref needed], with non-inverting outputs. Once pre-charged on a clock LOW cycle, they have an initial output of logic 0, and operate during the clock HIGH cycle. Once a gate has transitioned to 1, it will not transition back to 0 before the next clock cycle. In an SET capture application, precharge cycles could be relatively infrequent.

The symbol we adopt for these dynamic gates is the same as their static AND and OR counterparts, with the addition of a CLOCK input at the top of the symbol.

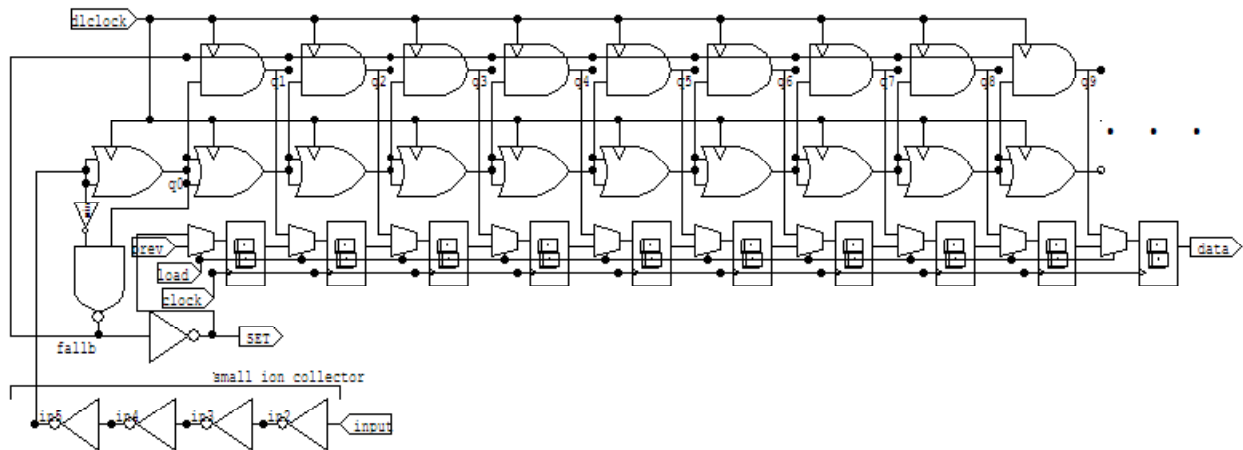


Figure 3: Dynamic Logic SET Measurement Circuit

An SET measurement circuit using dynamic (domino) logic (DL) is shown in Figure 3. It has a conventional static shift register for reading data out, and the same short ion collector used in the previous example. There are no capture latches, and the trigger circuit is considerably simplified. The operation of the circuit proceeds as follows:

- A pre-charge cycle on DLCLOCK clears the logic states of the domino logic. This repeats as necessary.
- An SET generated in the ion collector, or a bench test pulse, arrives at INPUT.
- The first OR DL gate triggers immediately and stays triggered. The trigger pulse travels down the chain of OR gates with near minimum DL gate delay, since the loading is approximately 2 gates at each stage.
- As the trigger pulse travels down the OR chain, it is copied at each stage to the input of an AND gate. These are independent gates and not chained. They serve as buffers so that the dynamic OR gates are not loaded by the output register. The second inputs of the AND gates are tied together and used as a freeze signal (active LOW). The AND gates which have already transitioned from 0 to 1 cannot be affected by the freeze signal (named FALLB in the diagram). But the AND gates connected to OR gates which have not yet transitioned will be prevented from doing so. So the AND gates will record the length of the SET, plus whatever time the 2-gate static logic freeze circuit requires to operate.
- Two gates of static logic look for the trailing edge, and issue a freeze command by taking FALLB low. FALLB refers to the “falling edge” of the input pulse, which will be the trailing edge since it is assumed to be a positive pulse. Negative going SETs can be detected by placing an extra inverter at the input of this circuit.

- The test control circuit, perhaps residing in external electronics, causes the shift register to be loaded, and the data to be serially retrieved. A chaining input is provided for connecting several experiments together.
- Since it is the sensitivity of DL to SETs that we wish to take advantage of, we also have to deal with spurious SETs originating in the DL gates. This is easy to do by monitoring the outputs of the AND and OR chains, and if they go HIGH without a trigger (signal SET), a precharge cycle can be used to clear the DL gates of any SET induced within them.

This design provides an additional 25% improvement in resolution (or interstage delay), and approximately 60% improvement in the shortest pulses measured. We look at performance details below.

Performance of SET Measurement Circuits

To give a process independent idea of the relative performance of various SET measurement architectures, we will express performance timing in terms of multiples of the basic gate delay of single-loaded inverters. For example, if the inverter delay is 80ps, and the minimum pulse width that will trigger a pulse capture is 200ps, then we will say the trigger sensitivity is $200/80 = 2.5$ gate delays (gd).

In general, we take trigger sensitivity to be the length of the shortest pulse that can be measured. The circuits considered vary in that some may record one latch for any trigger, and some may record a latch bit only for longer pulses. This difference is generally moot as long as the trigger signal is separately available to the test control circuitry. If not, it should be included in the output data readout, as was done with the dynamic logic circuit above.

The chart below shows the relative performance of the original leading edge triggered capture latch circuit, the improved trailing edge triggered circuit, and the two dynamic circuits discussed in this paper.

SET Capture Circuit Performance

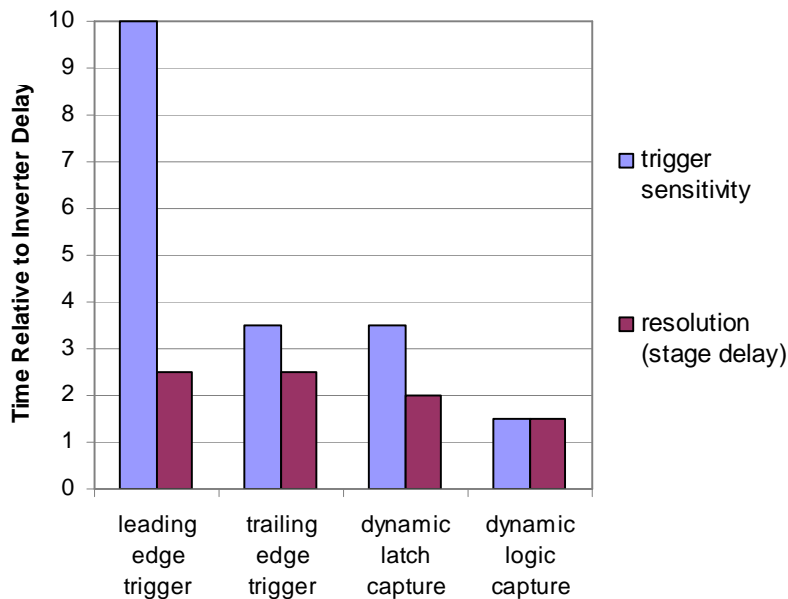


Figure 4: Comparative Performance of SET Capture Techniques

Another way of comparing SET capture circuits is by the layout area they consume. In some cases, many capture circuits may be needed, and their area may exceed the circuits being measured, so this area can become important. The conventional latch circuits take modestly more area per stage than the dynamic circuits. The dynamic latch is considerably smaller than a conventional latch, and so it has an area advantage, although the readout flip flops will be unchanged and will dominate the area. The dynamic logic capture will take a similar area per stage, but due to its fine resolution, it will take many more stages to capture the longest pulses. If interest is only in short pulses, the long pulses can be sacrificed. In the next section, we will address another solution to this issue.

A Compact Wide-Range SET Capture Technique

When measurements are made over a wide range of pulse lengths, it is often not necessary or even desirable to maintain the same absolute resolution over the entire range. A logarithmic or approximately logarithmic scale would be preferable. As long as pulses are being propagated down the capture latch chain, this approach is impractical because it would result in absorption of shorter pulses. But with the dynamic logic capture circuit, the pulse does not propagate, only a timing signal initiated by the SET, so we are free to vary the interstage delay and adopt a more practical pulse length scale.

In principle this is as simple as inserting extra delay stages or adding node capacitance. We want a compact scheme, but also one that does not require creation of many special layout cells, and preferably one that does not require much tuning for a new process. For those reasons the circuit of Figure 5 uses the inputs of standard logic gates to add progressively more capacitance, and thus delay, to the nodes in the timing chain of dynamic OR gates.

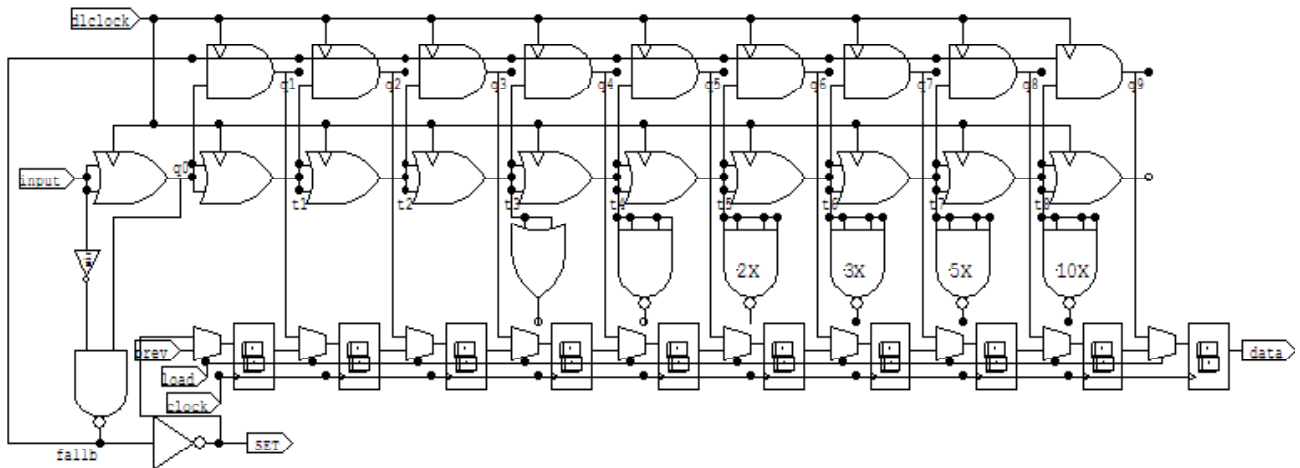


Figure 5: Compact Wide-Range SET Capture Circuit

This circuit differentiates pulses between zero and about 40 gate delays into 9 categories or “bins” of non-uniform size, preserving high resolution for shorter pulses and using progressively larger bins for longer pulses. Pulses over 40 gate delays are collected in a tenth bin.

Thirty logic gates are used as dummy loads to adjust stage delay timing. This is not as compact as using 6 custom tailored capacitors for each process to be evaluated, but makes considerably better use of the experimenter’s time. To cover the same range of pulse widths, using conventional uniform stage delays, would require at least double the number of stages shown. That would require the equivalent of 90 additional gates. So using dummy logic loads is a good informal optimization of area and the experimenter’s time. Depending on the cells present in a library, even more effective ones than those shown might be found.

As process geometry decreases, gate delay, which is proportional to node capacitance, decreases approximately as the inverse square of the geometry, while maximum SET length seems to decline at best linearly [need reference]. Thus the range of pulse widths which are important increases relative to gate delay, and the twin problems of measuring short pulses and constructing compact capture circuits become worse. So techniques for addressing those problems will become more important.

Figure 6 shows the stage delays for the circuit of Figure 4 as simulated in a 250nm process. The leftmost node state “q0” (light green) is always triggered if an SET is detected at all. The next node state “q1” (red) is a bit further away than we’d like because of the loading at q1 by the trigger circuit (more about this in a moment). Q2 and q3 occur at minimum intervals, and with q4 the intervals begin to gradually increase.

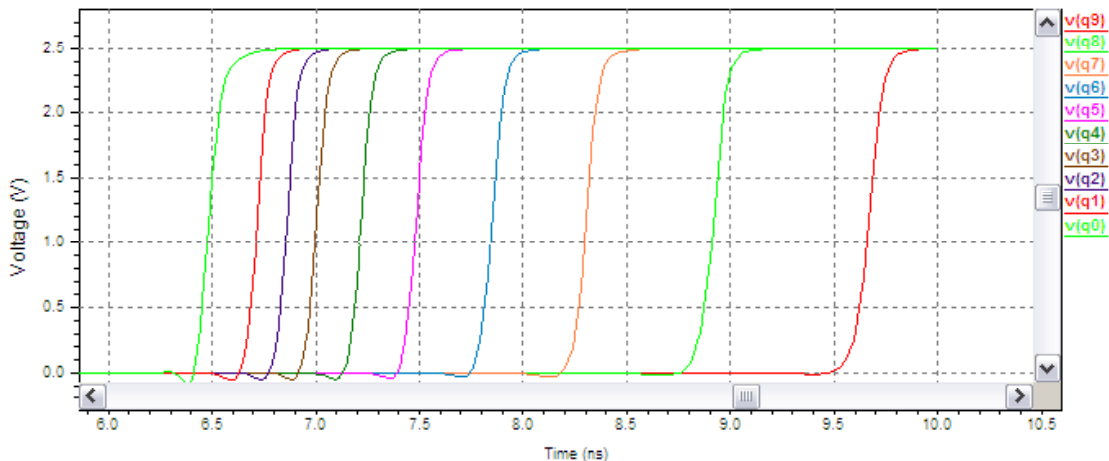


Figure 6: Stage Delays for Wide-Range Pulse Capture Circuit

Calibration of Pulse Capture Circuits

The higher performance capture latch circuits we have seen have non-uniform stage delays due to placement of trigger circuits. This could be eliminated if we sacrifice resolution, by adding extra load to the fastest stages. Non-uniform delays may also exist due to device and parameter variation, and routing variability. Nor can one consider the capture circuit alone, because all pulses do not originate exactly at its input. Instead, most pulses originate elsewhere and are propagated to the input of the capture circuit either by a chain of gates, or by a merge tree (covered in later sections). The amount of pulse width distortion in the feed circuit, which we call an “ion collector,” varies with path length and character.

Calibration is essential to place bounds on the amounts of these various pulse length distortions. Without it, the data may be entirely misleading. One way to calibrate is to arrange the circuit layout so that a laser pulse of known energy can be injected at various positions in the ion collector, or directly at the input of the pulse capture circuit. What we will consider here is a bench test input that is attached to the ion collector at some point.

We have used several methods of generating test pulses, none of which is perfect:

- An external pulse, which is usually not short enough for full calibration.
- An external pulse modified by passive components to barely touch the pad switching threshold and thus be very short, but whose length inside the chip is only approximately known.
- A series of 10 capture latches plus some pulse logic, which obviously generates only pulses of one fixed length.
- An analog variable length internal pulse generator [Anitha], whose length is calibrated by examining a ring oscillator made of the pulse generator circuits (the most complex, but most flexible).
- An inverter chain with some logic to extract pulses of selectable length (our latest method, quite simple, still in fab and not yet evaluated)

An example will show how badly data can be skewed if calibration is not adequate, or if the ion collector circuit introduces too much path dependent distortion in pulse width. The two data sets of Figure 7 use identical pulse capture circuits (two instantiations of the same layout block) of the conventional latch, trailing edge trigger type. So there is no difference in the data owing to the pulse

capture circuit. In this case the bench test pulse passes through an ion collector which is a simple chain of 240 inverters, a modest length compared to what some investigators have used. The ion collector labeled “A cells” has ordinary guard rings (substrate contacts) around the P and N regions of each inverter. The “G cells” have guard *drains* [reference Balaji’s paper on Jody’s guard drains] laterally between inverters, and substrate contacts above and below each P or N region. The experiment was intended to measure the effect of the different guard structures on the collected charge and thus SET length. For each of the 8 bench tests in the figure a different test pulse length was used, and the same pulse was routed to both ion collectors (internal to the chip). The output pulse lengths, measured as number of capture latches triggered, are quite different.

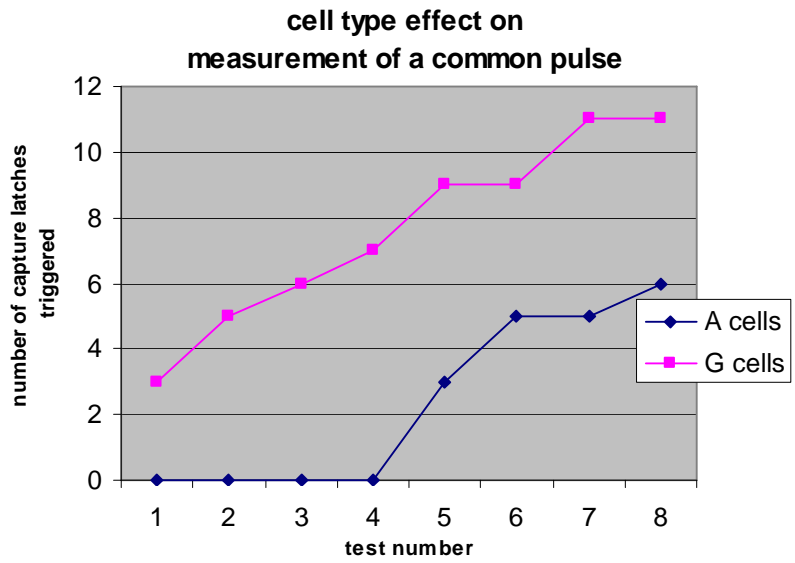


Figure 7: Effect of cell layout details on pulse propagation in a 240 inverter string

Are the A cells shortening pulses, or are the G cells stretching them? There is not sufficient trusted independent calibration in this setup to make the determination.

We will use more data from this same chip (a 180nm bulk process) to illustrate other points about ion collectors later. The point we want to make here is that calibration is essential to detect unexpected problems.

Measurement of SETs from Charge Sharing

It is widely known that a paraticle of ionizing radiation may generate SETs at several nearby nodes. In the case of extreme angle strikes, the particle may actually pass through many nodes. More commonly, charge from an ion strike at a particular location may be shared through the substrate [Amusan]. By charge sharing we do not mean electrical propagation of SETs. TCAD modeling or direct experiment is needed to evaluate charge sharing. Direct measurement of charge sharing under ion strike conditions, in order to validate models, is a current issue in SET measurement.

Simultaneous SETs in interleaved ion collectors.

One method that has been suggested to directly measure charge sharing is to use interleaved strings of logic gates, usually inverters [Narasimham, Bhuva]. If SETs are detected simultaneously in two or more of the interleaved strings, they are presumed to be the result of shared charge from a single ion strike.

Each string in such an interleaved ion collector must be separately monitored by a pulse capture circuit. In our experiment, when any one of them is triggered, all three are read out in sequential daisy chain fashion, and then the entire experiment reset for the next capture.

In order to efficiently construct triple interleaved strings of inverters, a layout block with 15 inverters wired as 3 interleaved strings of 5 inverters each was constructed (Fig. 8), and 48 of these were placed by an auto router, resulting in 3 interleaved strings of 240 inverters each (Fig. 9).

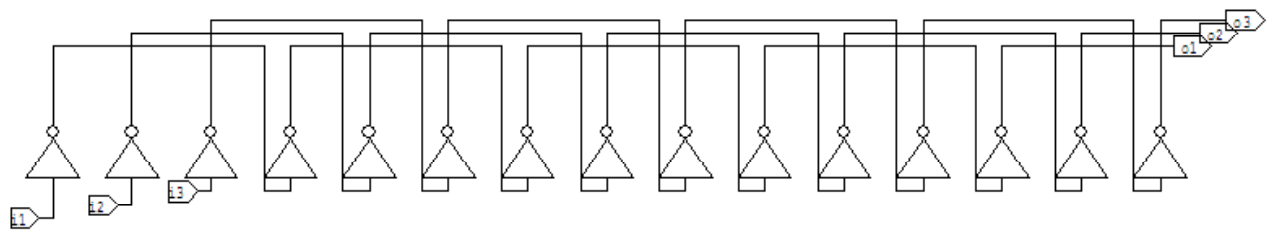


Figure 8: Layout Block (quasi-schematic) for Triple Interleaved Ion Collector

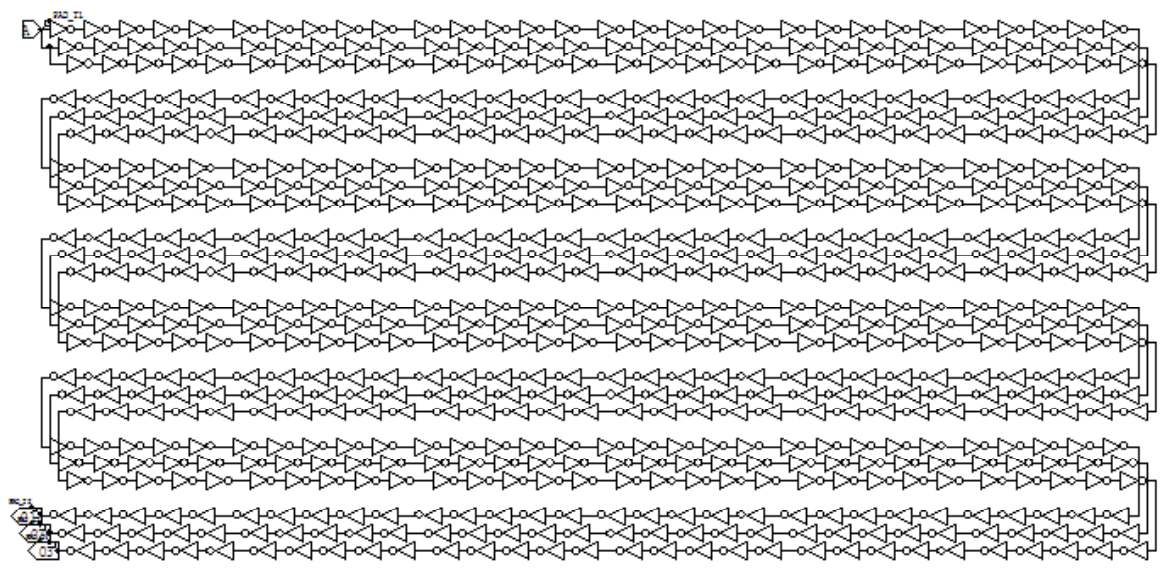


Figure 9: Triple Interleaved Ion Collector

Bench test results revealed unexpect pulse width distortions (discussed below) which undermined the ability to confidently use data from these experiments. Attempts to improve the experiment are taking two directions. One party to the original experiment is fabricating essentially the same design in a 45nm Silicon On Insulator (SOI) process, under the assumption that whatever coupling effects destroyed the original experiment will not be in play in a SOI process. Those of us who wish to measure charge sharing in bulk processes have attempted to understand and model the effects, and to construct a revised experiment which will be free of the effect, and can thus obtain valid charge sharing data.

The capture latch data from bench tests of the original experiment, done in 180nm bulk, based on the simultaneous input of the same pulse to all three strings in an ion collector, is shown in Table 1. The “test pulse” column gives the width of the test pulse as inferred from a ring oscillator of test pulse generators. Next there is a column for each of two cell types which gives the raw capture latch counts of all three strings. Sometimes a second set of latch counts is given if there was wide variation. In

parenthesis is given the capture pulse width implied by multiplying the largest number of capture latches times 0.16ns, which is the capture latch stage delay given by a ring oscillator of capture latches. There appears to be some pulse broadening by propagation through the inverter strings, which is expected [Massengill].

test pulse	normal cells	guard ring cells
4.18 ns	28-28-28 (4.5ns)	
3.65 ns	25-24-25 (4 ns)	
2.15 ns	21-21-21 (3.36ns)	24-23-25 (4 ns)
1.97 ns	19-19-19 (3 ns)	22-22-23 (3.5ns)
1.33 ns	11-11-11 (1.76ns)	16-15-16 (2.56ns)
1.13 ns	9-9-9 (1.44ns)	11-11-12 / 11-7-11 (1.76ns)
1.05 ns	7-9-7 (1.44ns)	10-10-10 (1.6ns)
.95 ns	7-8-7 (1.28ns)	7-7-8 (1.28ns)
.89 ns	4-7-4 (1.12ns)	7-8-8 / 8-0-8 (1.28ns)
.86 ns	0-6-0 (0.96ns)	
.75 ns	0-5-0 (0.8ns)	
.69 ns	0-3-0 (0.48ns)	
.665 ns	0-0-0	

Table 1: Bench Test Data from Triple Interleaved Ion Collectors

Notice that in the outer strings of the “normal cells” (those having no particular structure separating adjacent cells), pulses below about 1 ns are dramatically absorbed. This was *not* expected. The guard ring cells, having substrate contacts completely surrounding the P and N regions of each inverter, seem largely free of this problem, although they erratically and unpredictably may absorb the pulse in the center string for pulses near or below 1 ns.

It was hypothesized that some sort of coupling between adjacent inverters in the “normal” layout might explain the unusual behavior. For example, such coupling might act through drain sidewall capacitance to link adjacent drains, and would be reduced by the substrate contacts between cells for the “guard ring” case.

The total drain capacitance for our 180nm inverter, including both P and N FETs, area and fringe capacitance, is 0.003pF. We tried allocating some of this as adjacent node coupling in the ion collector Spice model. While not able to match the actual data exactly, somewhat similar effects could be produced. Figure 10 shows the Spice result when 0.002pF node coupling is used.

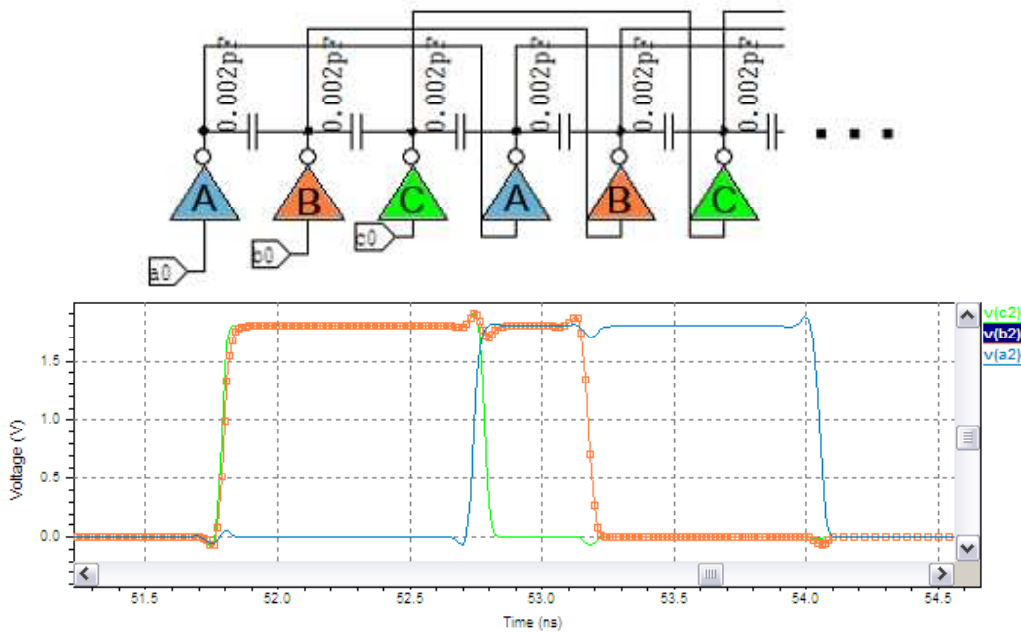


Figure 10: Coupling model and Spice result for 1 ns input, 240 stages, and “normal” cells

Highlighted is the center string pulse (orange) which has been stretched to about 1.3ns, consistent with Table 1. The two outside pulses, shown in green and blue, have become non-overlapping. One of them is stretched a similar amount, the more delayed one, whereas the early one (green) is still 1 ns.

Why does any string respond differently than any other? They are physically identical, with identical surrounding structures. But the events in time are different. Early in the chain here is what each inverter sees:

- CENTER (B) – both neighbors transition at same moment the center inverter transitions.
- LEFT (A) – one neighbor has already transitioned.
- RIGHT (C) – one neighbor has not yet transitioned.

Under these conditions the center (B) transitions faster, because with the voltage moving in the same direction on each side of its coupling capacitance, there is no parasitic current. (A) has just been pushed in the wrong direction by the transition on the preceding (C) inverter, so the (A) transition takes longer. The three strings drift apart in timing due to asymmetric mutual influences. The left side (A) is retarded and diminished by its predecessor neighbor moving in the opposite direction. The center (B) is boosted by same-direction transitions in its neighbors. The right side (C) is only slightly retarded by its successor neighbor which is not yet transitioning. Spice probably does not provide a high accuracy simulation of the coupling capacitance because it does not model substrate diodes.

If this hypothesis has merit (which is not certain, but suggested), then the problem of interleaved string pulse width distortion may be confined to bulk CMOS. SOI would not have as much coupling of this kind. However, if there is coupling through the power supply node, it could apply to both technologies. The best way to find out is to run comparable well calibrated experiments in an SOI process.

Given the lack of consistent string differences in the guard ring isolated cells, a re-design of the interleaved ion collector to reduce coupling seems warranted. The length should be reduced as well.

But how to design a detector for charge sharing, inherently a form of coupling, without the coupling that is ruining this experiment?

Non-repetitive, merged, interleaved ion collectors for charge sharing measurement.

The first ion collectors attached to pulse capture circuits [Shuler 2006] were not single long inverter chains, but shorter chains merged with INVERT/NAND (i.e. OR) trees. We return to that architecture to reduce length effects. But the coupling effects appear even stronger than length effects. How do we eliminate coupling without too small an ion collector, or so many merges that pulses are absorbed?

The coupling effect that ruins the interleaved inverters, according to our hypothesis, depends on pulses propagating through cells that are adjacent in the same relative positions at each stage, so that the edges of the pulse push and pull on one another to stretch or shrink pulses, or shift them in time. Detection of charge sharing only requires that cells be adjacent in large numbers. The pattern of adjacency is unimportant. Figure 12 shows 6 interleaved strings of inverters connected such that any given two strings are adjacent only in every 3rd stage!

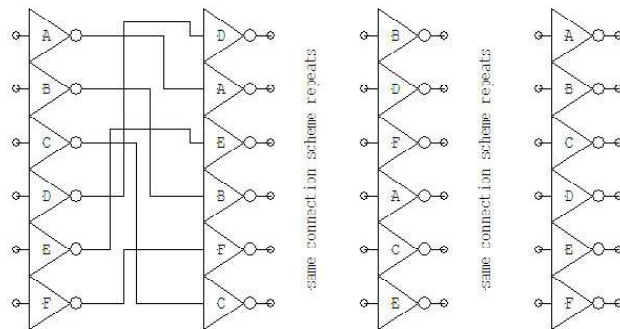


Figure 12: Interleaved inverter strings A through F, without sequential adjacency

The connection pattern between each stage is identical. Used with a 4 to 1 merge pattern as shown in Figure 13, six interleaved strings with a total of 408 inverters can be created with no more than 6 points at which any shared pulse finds itself again adjacent to the other string carrying its partner.

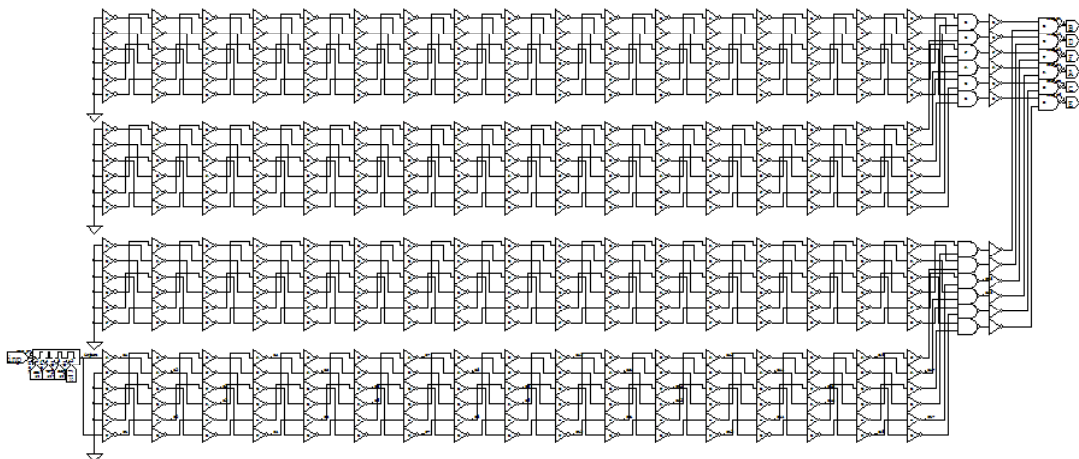


Figure 13: Non-Repetitive, Merged, Interleaved Ion Collector

The module at the lower left is the selectable pulse generator mentioned earlier, with 4 control inputs. It can provide bench test and calibration. It is attached to only two of the strings so as to better model propagation of dual SETs from a charge sharing event.

One further improvement was made to the interleaved ion collector. In real circuit layout, all transistors drains are not equally spaced. Some drains are at minimum spacing, which makes charge sharing more likely. To mimic this condition, we flipped every other inverter in the manner shown in Figure 14, so that every drain is at minimum distance from one neighbor.

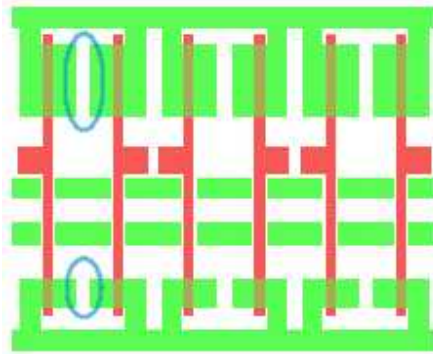


Figure 14: Every other inverter flipped for drains at minimum spacing (diffusion and poly shown)

In summary, anything one does to amplify a small or infrequent effect such as charge sharing, in order to measure it, is also likely to also amplify something that you don't want to measure, such as the coupling phenomenon that we just described. The unwanted factors must be identified and eliminated.

Measuring SETs from Logic Circuits

An SET can appear anywhere in a circuit, whenever a node voltage differs from the substrate or well. It is as likely to appear in the test circuit as the circuit under test. Experimenters have resorted to monolithic arrays of chained circuits to gather SETs and funnel them to a test circuit for measurement. But aside from memory, such arrays are not typical of useful circuits. Chains of inverters are typical of nothing except perhaps the occasional ring oscillator. So investigators are beginning to ask what kind of circuits are typical, and how do we get SETs from them?

[need to identify and discuss history of measurement of SETs from logic, including studies of effect of clock speed on the capture of SETs by FFs]

Multiplexors

Two kinds of circuits account for much of digital logic. The most common circuit is the multiplexor. Multiplexors select and route data, and implement logic functions. The ways of implementing multiplexors are as varied as their application. Common examples are shown in Figure 15. Each of them has qualities that suit one application or another. Each has unique characteristics for the generation and propagation of SETs. Logic muxes are faster but usually larger. Tristate muxes used to be popular for putting data on busses. Passgate muxes are smaller and typically used in routing networks, but are slower.

Though we emphasize the differences in these circuits, they all have a common feature in that they have some transistors in series, and turning these transistors on propagates the selected signal. In both NAND gates and NFET switches, it is series NFETs that do the selection. In other cases both PFETs and NFETs are in series. On the inside, the gated logic mux and the tristate mux are basically the same

circuit, one is just made into modular components and the other tightly laid out. The full passgate mux basically removes input and output buffer transistors from other versions of mux circuits. All these differences, however small they are conceptually, affect the generation and propagation of SETs.

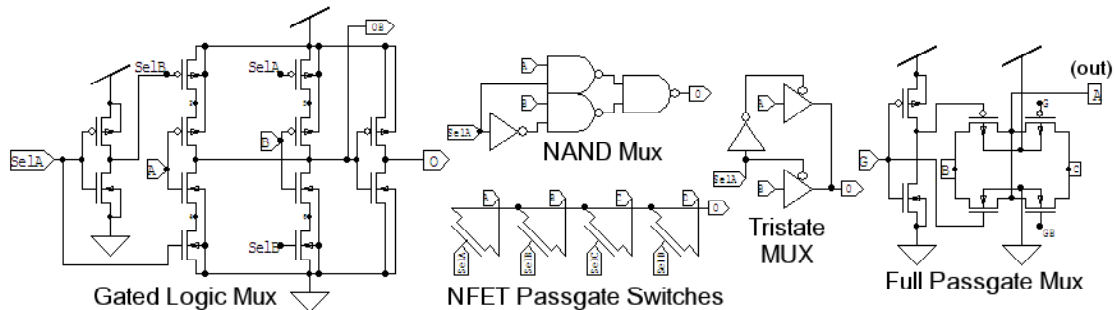


Figure 15: Common Multiplexor Circuits

In many kinds of busses or passgate routing networks, an SET that erroneously places data on the bus can affect dozens of other signals. But because of higher capacitance, it may take a larger SET to disturb a routing network. In fast logic muxes, the only disturbed signal may be the output, but even a tiny SET may be fully propagated.

Memory addressing networks are basically muxes. Routing networks of course are muxes. Surprisingly, Combinational Logic Blocks (CLBs) in FPGAs are also muxes, combined with Look-Up Table (LUT) memory. And of course, even in an Arithmetic and Logic Unit (ALU), selection of functions and routing of data is handled by muxes. If muxes and memory elements were the only circuits we had, we could still do virtually all types of computing. To implement any logic function, just construct a memory of its truth table, and use the input signals to select the answer from the truth table. That is how a LUT works.

Notice that the “gated logic mux” is also the same circuit as the common latch. To make it a latch, we just connect the output to either of the inputs. The select input becomes the clock, and selects whether the latch accepts new data or retains old data. So this type of mux has been studied extensively in its incarnation as a latch.

Arithmetic and other logic functions

When speed, power and size are more important than generality, muxes give way to combinational logic. In theory all combinational logic could be implemented as a sum of terms (NAND/NAND or AND/OR tree), which has fairly benign SET characteristics, but in practice it is more economical to use ad hoc logic, which has intermediate and re-combined terms along paths of different lengths.

The circuit of Figure 16 adds two bits, A and B, plus CARRY_IN, producing a SUM bit and CARRY_OUT. It consists of two half adders, and a NAND gate to merge the carry from each half adder. The first half adder adds A and B, and the second adds CARRY_IN to the sum of the first.

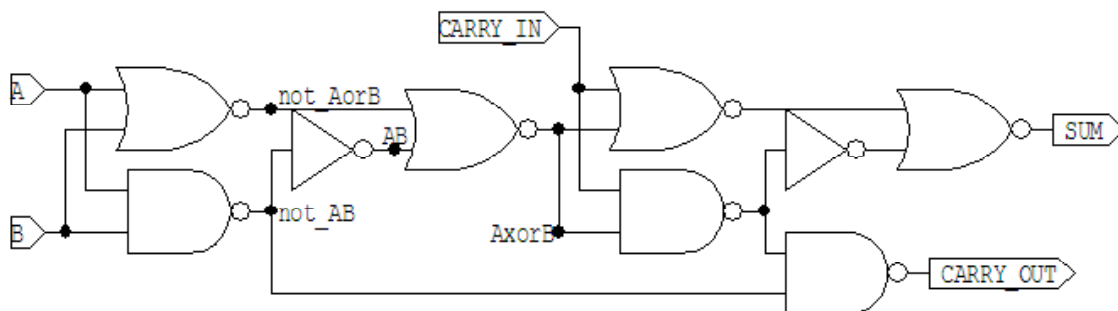


Figure 16: Full Adder with Ripple Carry

Notice that a half adder performs an XOR (exclusive OR) function, another commonly used operation. If you look inside a commercial XOR layout cell, you see basically the circuit of a half adder, tightly laid out, but present in all details. So the SET characteristics of half adders will apply to XOR gates as well.

Logic Clouds

For purposes of including logic in SET or SEU test circuits, we use the concept of a logic cloud [get reference from Jody]. This is just an arbitrary collection of logic that fits within a test circuit. The internals of it can be changed to whatever type of logic we'd like to evaluate. Preferably, the function would always be the same, so that the test circuit does not have to be re-designed to work with different logic clouds. Usually this is the identity function, i.e. the logic cloud output is the same as its input. Care is taken to assure the output is sensitive to SETs in logic paths we wish to test.

For an example, let's construct a logic cloud containing both arithmetic and mux circuits. We'll call this "COMBO." It's symbol and logic is shown in Figure 17.

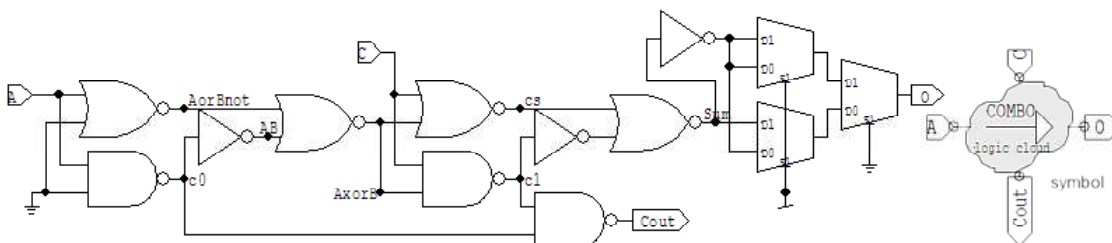


Figure 17: Combination Logic Cloud (symbol at right)

This logic cloud is different from most in that it has two inputs. We want to see the effect of SETs propagating along the carry path. For a carry in (C) of 0, it has the identity function $O \leq A$. An SET propagating on the carry path will always disrupt the output. For $A=0$ many sources of SET will alter the output O, and some will start an erroneously propagating carry as well. The output 4 to 1 mux is wired so that the first stage is in the select D1 state and the second stage is in the select D0 state, so there is variety in the mux state, which might result in different SET sensitivities. If we wanted to more closely evaluate the impact of SETs on the select inputs, we could attach logic or latches to its select inputs, instead of hard wired lines. The unused inputs of the mux are connected to the inverted state of the expected output, so that if there is an error in the mux select, it will in fact show up. Otherwise the error would be masked.

Detecting SETs from Logic Clouds

Most logic clouds, including the one above, do not have very good properties for chaining end to end. The full adder actually shortens (and eventually absorbs) many input disturbances because of its particular AND type recombination paths. Some circuits will lengthen pulses considerably because of OR type recombination paths. Most logic circuits, because of multi-input gate inefficiencies, will not pass the short pulses that inverters will pass. Yet in real logic circuits the short pulses cause problems. The difficulty here is that we cannot rely on chaining to collect and route them to a measuring circuit.

One approach is to use a merge circuit, similar to the one discussed in connection with triple interleaved ion collectors. However, since the logic cloud is not always in the same state, i.e. SETs are not always of the same polarity, we must take an extra step to detect them and convert them to pulses of a uniform polarity. Figure 18 shows one solution to this.

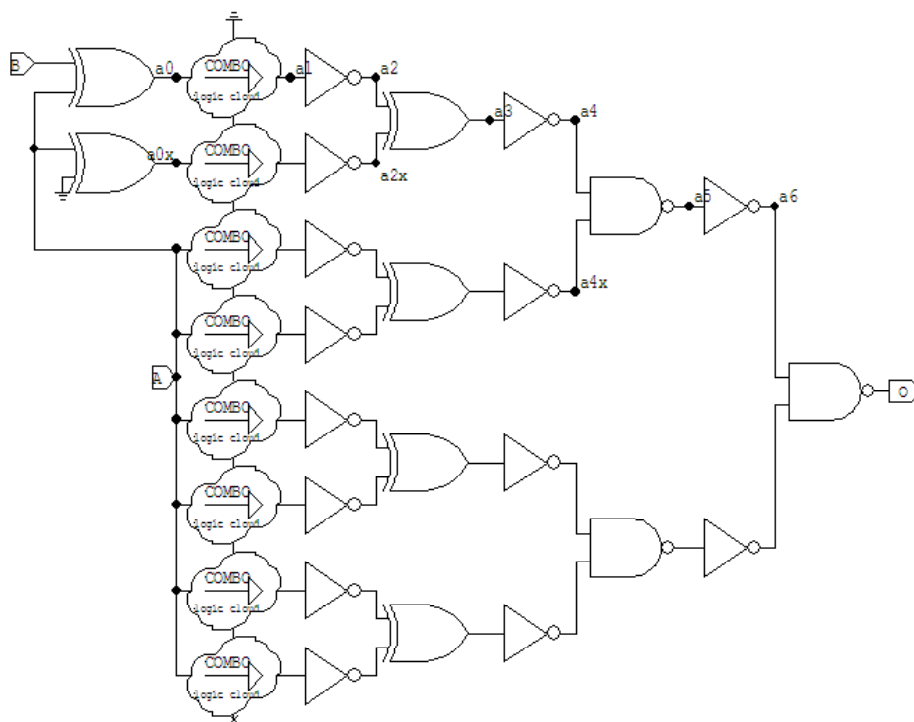


Figure 18: Detection and merger of SETs from logic clouds

The inverters following each cloud provide buffering and pulse shaping, since the output of the cloud might be driven by a weak multi-input gate. All logic clouds are driven with the same input A, except for a special test circuit we'll describe in a moment. So they should have the same output at any given time. By XOR'ing pairs of them, any SET is detected and becomes a positive going pulse. From there, an INVERT/NAND merge circuit follows as before.

A test input is necessary to make sure the circuit is working, and to exercise and debug the test rig before going to a heavy ion or other test facility. In this case, since the input A is unknown and may change dynamically during testing, we arrange for a test input B to invert the input to the first logic cloud by means of an XOR gate. A second XOR gate is used at the input of the second logic cloud to balance the timing of the first two logic clouds, so that in normal operation the SET detect XOR at their output will be happy and not mistake unequal timing paths for a short SET.

The circuit of Figure 18 gives us 8 instances of the logic cloud from which to collect SETs. This might not be enough. However, it is easy to repeat this whole process, creating an upper level module instancing 8 of the Figure 18 circuits, giving 64 logic clouds. At that point, the area of logic clouds exceeds the area of most inverter string ion collectors. The overhead of the merge circuitry is relatively low because of its hierarchical nature. And because of its careful buffering and balancing, it usually will not have much effect on propagation of the SETs, probably less than the last few stages of most logic clouds.

By this means, SETs can be gathered from many different types of circuits and fed to a pulse capture and measurement circuit.

Comparison of SET and SEU data

SET measurements produce a volume of data, including histograms of SET width for each different ion and beam angle and type of circuit. While this data can be useful to a designer wanting to know how long the SETs are which must be tolerated, it is not very useful in predicting the error rate performance of actual circuits in a particular environment. By contrast, an elaborate science has been made out of predicting in situ error rates from SEU data.

For this reason, it is a good idea to include some SEU experiments, involving flip flops and the same type of logic circuits on which SETs are being measured. The logic cloud, with its identity function, can be easily inserted into most types of flip flop SEU experiments. The ideal circumstance is where space and pinouts permit various control experiments, such as an SEU experiment with no logic, and one with the same logic cloud as used in the SET experiments. But if several logic clouds are to be measured, the number of control experiments can grow rapidly. Another alternative is to use a slow clock speed test run as the non-logic control, if test facility time permits.

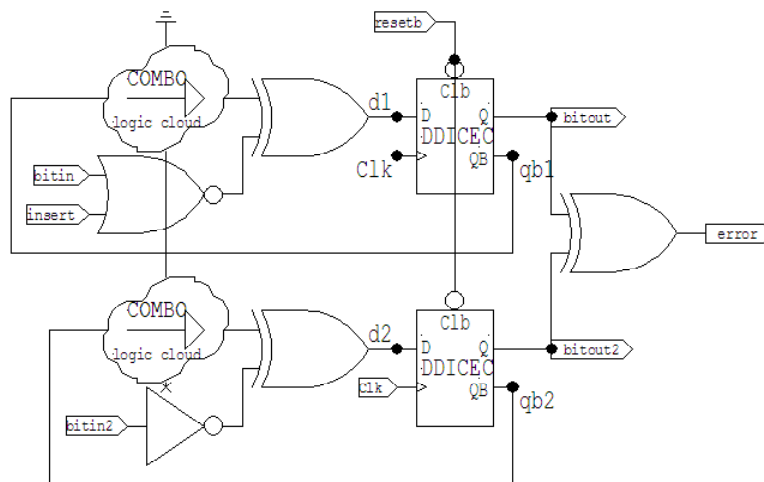


Figure 19: Logic clouds in an SEU test cell

Figure 19 shows how the COMBO logic cloud with its carry circuit can be incorporated into an SEU test cell which we have often used [Shuler 2005, 2006, 2008]. The two flip flops are part of parallel, identical test circuits connected through the BITIN/BITOUT and BITIN2/BITOUT2 signals. If they disagree, an SEU is reported by signal ERROR. A test circuit with signal INSERT is provided to check the functionality of the circuit and the test rig. The use of the logic clouds does not change the operation of the circuit, except in the generation of SETs, which will become SEUs if they arrive at the flip flops near a clock edge.

Summary and Conclusion

We have seen how pulse capture circuits, used to measure SETs, can be designed to measure shorter pulses, and can be made smaller while measuring pulses over a wide range of widths. Difficult problems in early attempts to directly measure SETs arising from charge sharing were analyzed, and workarounds proposed. Finally, rationale and methods for measuring SETs from various types of logic were explained.

SET measurement is a developing field, with many opportunities to ask new questions and make new discoveries. We hope this presentation will not just inform, but inspire investigators to adopt and improve the latest techniques. Large realistic designs, such as FPGAs, and high performance processors, are needed for future space missions. These remain to be investigated in detail below the “black box” level. Only detailed investigations will lead to design improvements. Black box investigations, though very valuable, lead mostly to workaround designs involving application or hardware redundancy. Interest in SETs and SEUs arising in terrestrial applications continues to increase, and the potential exists to explore technologies and designs that might not be considered for traditional extreme environment applications.

In closing we reflect that the challenges of SETs are like the old Zen swordmaster who took on a new student. At first nothing happened, and the student complained that he was not learning anything. So the old master took to attacking the student at random moments when he was cooking or sleeping [Herrigel]. And so we strive to be ever watchful for whatever may happen when we place a new circuit or technology under nature’s random attack.