

# Sequential Circuit Design for Spaceborne and Critical Electronics

R. Katz<sup>1</sup>, R. Barto<sup>2</sup>, and H. Tiggeler<sup>3</sup>

<sup>1</sup>NASA Goddard Space Flight Center, Greenbelt, MD 20771

<sup>2</sup>Spacecraft Digital Electronics, El Paso, TX 79904

<sup>3</sup>Surrey Satellite Technology Ltd., Guildford, Surrey GU2 5XH

## *Abstract*

Circuits destined for use in critical applications such as spaceborne and military control circuits demand high-reliability and robust operation. Additionally, traditional considerations such as low power operation often apply. For implementation in FPGAs, limited resources and the device's architecture also have an impact on the circuit topologies used. Clock generation and distribution structures combined with storage element topology will be examined. Analysis of different finite state machine topologies when subjected to external disturbances such as electrical transients and single event upsets will be performed. The capabilities of various error detection and correction schemes for the various state machine classes will be analyzed.

## I. INTRODUCTION

This summary will provide an overview of the topics that will be addressed in the full paper. Details, circuit diagrams, and quantitative analysis will also be addressed in the full paper.

Sequential circuit design techniques for digital electronics are well known and are a standard topic in an elementary digital design text book. Shift registers, counters, finite state machines (FSMs), and other standard structures are all taught to the student in the first class. Logic synthesis technology has abstracted the design process, with machines generating the detailed design. It now becomes critical to not only understanding proper design and analysis but the algorithms embedded in these tools.

After examining many samples of logic design in spaceborne digital systems [1], it is apparent that designing reliable and robust sequential circuits is a topic that needs to be addressed. This paper will start from familiar basic principles and introduce and analyze structures and techniques used to construct sequential circuits. The analysis will cover basic circuit reliability under normal operating conditions such as variations in temperature, voltage, process and life time. Logic structures that affect reliability will be analyzed, including clock generation and distribution as well as decoding. Additionally, circuit behavior from anomalous conditions such as severe electric transients (power bus disruption, ESD, etc.) and radiation (SEU) will be analyzed quantitatively. Lastly, the effectiveness of various error detection and correction schemes will be covered.

The discussion and analysis will be focused on generic logic design. Since FPGAs are a very popular choice for the

implementation of logic circuits, the architectural and technological impacts of this technology will be addressed in detail.

## II. CLOCK GENERATION AND DISTRIBUTION

The simplest model for clock generation and distribution is to buffer an incoming signal and drive it to multiple loads. These loads can be either flip-flop inputs or the gates to latches. The characteristics of this signal will vary depending upon how the state machines are designed and the tools used to design the logic. For example, if edge-triggered flip-flops of a single polarity are clocked in parallel, then frequently, to meet the  $t_H$  specification, the skew on the clock signal should be minimized. Of course, there are certain designs that can deliberately introduce skew to maximize pipeline stage performance but that is outside the scope of this paper. At the other extreme, ripple counters are inherently insensitive to skew, using multiple clocks with one load per signal, so the limits on skew are not necessarily important. Other structures may be formed by using a single clock where sequentially adjacent flip-flops are clocked by opposite edges of the clock, trading off control of skew for duty cycle. For low to moderate speed circuits, analysis to show reliable operation is trivial. Another structure is to use latches to hold data. Here, a two-phase, non-overlapping clock structure provides a compact circuit design, particularly in technologies such as Act 1, where no hard-wired flip-flops are provided. The full paper will show methods for constructing these clocks and analyze circuit performance.

It turns out that even this simple model does not provide reliable clocking. First, as seen in [1], many designers use high-skew clocks for parallel clocking, instead of the provided low-skew, high fanout clock buffers. Secondly, buffers, both signal and clock buffers, are susceptible to single event transients (SETs) [Katz, Mavis]. Runt pulses can be distributed over the clocking network.

More recent technologies provide digital delay lock loops (DLLs). These control the delay of the clock such that the output of the clock distribution system has zero delay with respect to the input signal, maximizing system level performance. Additionally, some implementations will support the generation of multiple phases of the clock as well as other features. The full paper will address the construction of these circuits and then analyze these structures for SEU performance.

### III. FINITE STATE MACHINE TOPOLOGIES

There are a number of different state machine topologies, with the most common being the parallel clocked, single edge structure. The increasing use of hardware description languages such as VHDL are geared to this implementation, with other structures possible in that domain, but awkward. Schematic capture is much more amenable to other logic structures such as ripple counters, latches clocked by two-phase, non-overlapping clocks, and opposite edge clocking, for example. The full paper will show schematically (and textually when appropriate) the different structures and critical paths.

For parallel machines (others to be included in the full paper), popular implementation strategies include straight binary, gray coded, one-hot, and modified-one hot. Another little-used implementation strategy includes the Johnson twisted ring counter. For a machine with  $m$  flip-flops, these structures have the capability of representing from  $m$  states to  $2^m$  states. Obviously, the implementation will drive the probability of an error from sources such as cosmic rays and protons, which generate SEUs. Each structure will be quantitatively analyzed in the full paper for relative probability of failure.

### IV. STATE DECODING

State decoding to generate outputs is frequently a problem [1]. Many designers do not understand that certain structures can generate glitches. Examples seen in flight hardware include decoding parallel or ripple counters directly or using the TCNT output of counter IP, which is equivalent.

The one-hot structure enables glitch-free decoding for single states, with the output of the flip-flop driving out of the module. However, an enable, say for a write pulse, generated from two consecutive states may have a static hazard. Static hazards and methods of eliminating them will be reviewed in the full paper.

An interesting circuit structure which permits economical and glitch-free decoding, if done correctly, is the Johnson twisted ring counter. Single states as well as multiple states can be asynchronously decoded reliably.

### V. COUNTERS

A thorough discussion of different counter structures will be given. Applying TMR blindly to make reliably counters, for example, will not work [3].

### VI. ERROR DETECTION

For many applications, detecting the error and generating a signal to initiate correct action may meet the system requirements. Traditionally, error detection has meant the use of redundancy, such as a parity bit and a generation/checking mechanism. For SEU detection, a single bit (assuming no MBUs) is adequate; for a more robust performance level, more protection is needed.

Obviously, for a binary encoded state machine, this additional information is needed, with the extra bits increasing the error rate. Without the bits, then there may be no illegal states and detection of an error is difficult. Other topologies, such as the Johnson twisted ring counter and on-hot machines, have illegal states. An interesting property of the one-hot state machine is that all legal states differ by at least two changes in bits; any single bit error from an SEU will result in an illegal state. This can be detected in a number of different ways, one of which is a simple  $n$ -bit LUT for a state machine with  $n$  states. The full paper will analyze all of the state machine topologies and the hardware required for different error correction schemes as a function of the size of the state machine. The impact of testability on the error detection mechanism will be discussed.

### VII. ERROR CORRECTION

There are several methods for error correction for a sequential state machine. One involved redundancy at the flip-flop level with a voter which can be automatically built by an HDL synthesizer. Schemes such as Hamming codes, which are used in memories, will be discussed, along with their limitations. Since SEUs are not synchronous with the clock, a single bit may result in multiple single changes, complicating schemes such as this. The impact of testability on the error detection mechanism will be discussed.

### REFERENCES

1. R. Katz, R. Barto, and K. Erickson, "Logic Design Pathology and Space Flight Electronics," MAPLD 1999, Laurel, MD.
2. R.B. Katz, R. Barto, P. McKerracher, B. Carkhuff, and R. Koga, "SEU Hardening of FPGAs for Space Applications and Device Characterization," *IEEE Transactions on Nuclear Science*, **NS-41**, pp. 2179-2186 (1994).
3. R. Barto, and K. Erickson, "SEU Induced Anomalous Behavior of Voted Ripple Clocks," MAPLD 1999, Laurel, MD.