

Asynchronous Logic Design with Subcells with an Application for Space

David F. Cox

Research Associate Professor
NASA Institute of Advanced Microelectronics
Microelectronics Research Center
University of New Mexico
801 University Blvd. SE, Suite 206
Albuquerque, New Mexico 87106

Abstract

Asynchronous sequential logic design needs to be done carefully to avoid races and hazards. As a result, typical designs are done at a fairly low level using state tables, transition diagrams, timing diagrams and flow charts. The state assignment can help alleviate races but hazards need to be addressed at every level. A technique is given where any asynchronous transition table can be synthesized using RS flip-flops and eliminating simultaneous switching of both the Reset and Set inputs to the RS flip-flop. Space flight electronics such as satellites and space probes require electronic circuits that are

tolerant of radiation environments. Particle hits will upset logic nodes, altering data patterns as well as modifying stored data within registers and memory elements. The circuits should also be low power due to the limited weight of battery and solar generation systems that can be lifted into space. Asynchronous designs can lower the power in electronics used in space flight in areas that include processors and state sequencer/controllers. A method is given that allows asynchronous designs to be done simply, using a radiation tolerant RS flip-flop as the basic feedback storage element in the asynchronous state variable definition.

1. INTRODUCTION

Asynchronous synthesis typically uses combinatorial logic functions connected in a feedback arrangement. The input excitation and feedback connections can be derived from a karnaugh map that is itself generated through a state sequence definition [1]. Space electronics are subject to radiation. The radiation particles cause unwanted conduction paths in the electronics by creating hole-electron pairs in areas where they happen to impact. CMOS circuits will see nchannel or pchannel devices

turning on when they are not supposed to, causing erroneous data transitions in combinatorial logic or reversing stored data bits in memory devices. One method to inhibit the unwanted transitions is to minimize the error-prone circuitry. A method is given to constrain all feedback paths in asynchronous circuits to subcells called flip-flops. A flip-flop design is derived that will reject single-even-upset (SEU) particle hits.

2. RS FLIP-FLOP

The standard RS flip-flop has two level-inputs labeled Set (S) and Reset (R), and one output, labeled Q. The occurrence of a "1" or "hi" level on the S input will cause the Q output to go to a hi level (setting the flip-flop). The occurrence of a hi on the R will cause the Q output to go to a low (resetting the flip-flop). The only constraint is that the S and R inputs cannot both be at a hi level at the same time. This constraint can be stated as

$$S \bullet R = 0. \quad (1)$$

The characteristic equation for an RS flip flop can be generated from a state table synthesis [2] and is given by

$$Q = S + !R \bullet q. \quad (2)$$

Q is the output state that will be forced at the RS flip-flop output a time-delay after the inputs are applied. This equation (along with the constraint equation) can be used as a basis for creating an asynchronous circuit with all direct feedback connections being within the RS flip-flop (a direct feedback connection is defined as an output being connected back to its own input combinational network). The method may best

be shown through an example.

3. SYNTHESIS

As an example, the phase comparator of a digital phase lock loop will be used. The loop block diagram is shown in Figure 1. The input data (D) is assumed to be a serial data stream to

which the oscillator clock (C) needs to be synchronized. The outputs of the phase comparator are CU (Charge Up) and CD (Charge Down), which drive current sources which charge or discharge the filter.

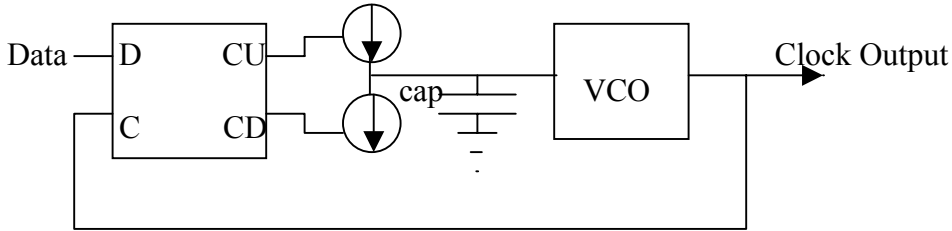


Figure 1. Phase Lock Loop Block Diagram.

The timing diagram for the phase compare operation is shown in Figure 2. If Data occurs before the Clock, the VCO (Voltage-Controlled-Oscillator) needs to speed up so CU is turned on to increase the capacitor (cap) voltage, increasing

the VCO's frequency (Clock rate). The Charge Down will turn on if the Clock rising-edge occurs first and will turn off when the Data rising-edge occurs.

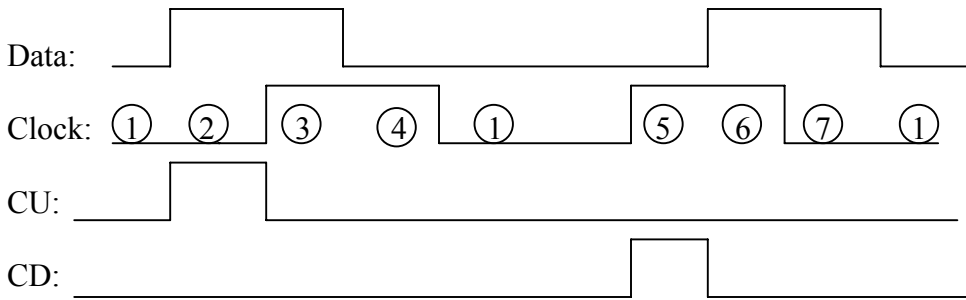


Figure 2. Timing Diagram for Phase Comparator.

DC	00	01	11	10	CU	CD	DC	00	01	11	10	CU	CD
State 1	①	5	-	2	0	0	State 1	①	5	-	2	0	0
State 2	-	-	3	②	1	0	State 2	-	-	3	②	1	0
State 3	-	4	③	-	0	0	State 3	1	③	③	③	0	0
State 4	1	④	-	-	0	0	State 5	-	⑤	3	-	0	1
State 5	-	⑤	6	-	0	1							
State 6	-	-	⑥	7	0	0							
State 7	1	-	-	⑦	0	0							

Figure 3. State Table with Reduction.

The numbers represent the states in the sequence that can be uniquely described from the inputs and outputs. The State Table is shown in Figure 3, along with its reduction. The state assignment

follows directly and is given in Figure 4. States were assigned to give adjacent transitions with the exception of input "00".

		DC	00	01	11	10	CU	CD
q1	q2	00	00	10	-	01	0	0
		01	01	-	11	01	1	0
		11	00	11	11	11	0	0
		10	10	10	11	-	0	1

Q1 Q2

Figure 4. Next State Karnaugh Map.

4. ALGORITHM

The Next State Map is broken into two maps, one for each of the states. The normal procedure would then be to generate the combinatorial functions required to generate each state and the design would be complete. Instead, relook at the SR flip-flop's characteristic equation and use Karnaugh maps as the variables, since a Karnaugh map is itself a function. Use the maps for each

state variable as the next-state variable as shown in Figure 5. Using equation (2),

$$Q_n = S_n + !R_n \cdot q_n \quad (3)$$

Where Q_n is the nth next state variable, S_n is the Set input for the nth state variable, R_n is the Reset input for the nth state variable and q_n is the present state for the nth state variable. The equation is valid on a cell-by-cell basis for each map. This is easy to show by taking the minterm expansion of each map in the equation. Thus, for Q1:

$$\Sigma \text{minterm}(Q1) = \Sigma \text{minterm}(S1) + \Sigma \text{minterm}(!R1) \cdot \Sigma \text{minterm}(q1) \quad (4)$$

Each minterm consists of a product of D, C, q1, q2 and the map entry. A specific value for D, C, q1 and q2 point to only one entry in each of the maps, which is in the same location for each map. The equation will then hold for each map cell taken individually. The "AND" function will AND one cell of !R1 with the sum of all of the cells of q1 ("OR" function) but only the cells with the same values of D, C, q1 and q2 can give

a non-zero result. ANDing two non-coincident cells would contain products of the form $(D \cdot C \cdot q1 \cdot q2)_{!R1} \cdot (D \cdot C \cdot q1 \cdot q2)_{q1}$ (5) At least two of the input variables will have different values for non-coincident cells, producing a "zero" result. Hence, we can eliminate the "Σ's" in the equation and use equation (3) across one cell at a time.

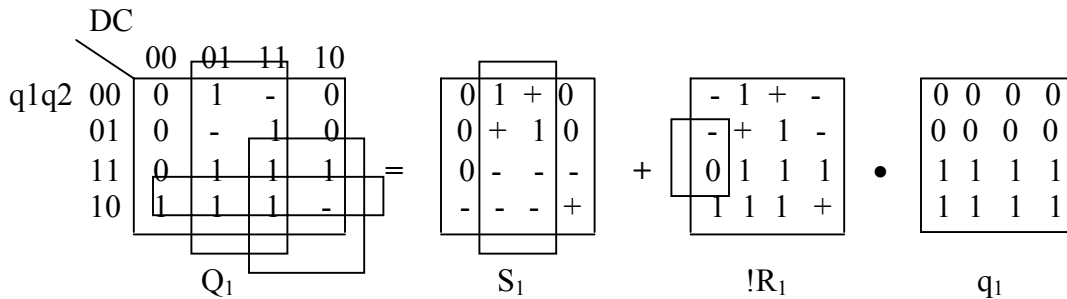


Figure 5. Karnaugh Equation for Next State Variable Q1.

The Data, Clock and present state input labels are not shown on the maps to the right and are the same as the next-state map on the left. The maps on the right of the equation symbol are filled in the following manner:

1.2 The Set map (S1) cells are filled in as:

4.1 The present-state map, q1, is an identity and its cells are filled in according to their value as defined on the appropriate row (q1 is 0 for the first two rows and 1 for the next two rows as shown on the Q1 map).

a. A 0 in the next-state map cell is filled as a 0 in the corresponding Set map cell.

- b. A 1 in the next-state map cell is filled as a 1 if the present state (q_1) is a 0 for that cell row, or as a “-“ don’t-care if the present state (q_1) is a 1 for that cell row.
- c. A don’t-care (-) in the next-state map cell is filled as a “+” in the corresponding Set map cell. The +’s are a special don’t care situation.

4.3 The Reset map (!R1) cells are filled in as:

- a. A don’t-care (-) if the corresponding Set cell is a zero and the present-state row is 0.
- b. A 1 if the corresponding Set cell is a 1 and the present-state row is either a 0 or a 1 (to maintain $S \bullet R = 0$).
- c. A “+” if the corresponding next-state cell is a don’t-care (-).
- d. A 0 if the corresponding next-state cell is a 0 and the present-state is a 1.

Note that the constraint equation (1) is met as long as the corresponding Set and !Reset cells are not 1 and 0 respectively. This is the

controlling criteria for item 4.3-b. The +’s are handled as don’t-cares and can be circled in each map as long as the constraint condition is met. For example the Set map could circle the middle two columns, giving $S_1 = D$. Since the !Reset map is the inverse of Reset the zeroes will be circled, giving $R_1 = !D \bullet !C \bullet q_2$. If the +’s in the Set map are 1’s, they must also be 1’s in the Reset map coverage (since the Reset then would be 0’s and the constraint of equation (1) would be met).

The RS flip-flop implementation will generally give a simpler realization due to the number of don’t-cares in the Set and Reset maps. Hazards in each map would be handled in the standard way of including all adjacencies in the coverage [3]. The RS flip-flop for space use would be designed to take particle hits without upsetting and the combinational circuitry can be standard cells in two redundant paths. The realization for Q1 using feedback and using an RS flip-flop is shown in Figure 6.

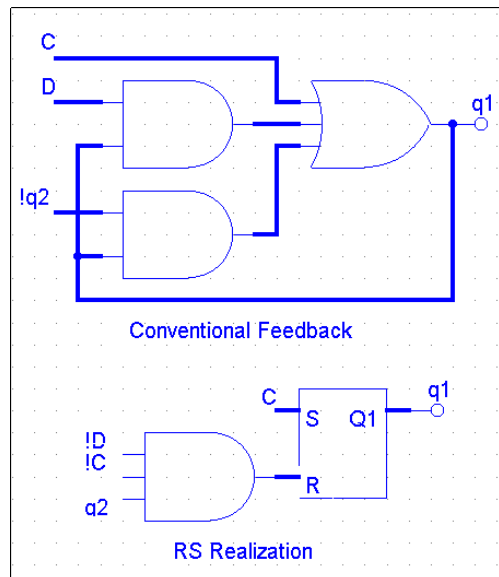


Figure 6. Logic realizations for Q1.

5. RADIATION-TOLERANT RS FLIP-FLOP.

The standard CMOS RS flip-flop is shown in Figure 7. A redrawn, but equivalent, circuit is also given. A radiation particle hit can be modeled by the current source on U7. If a high-energy particle hits the channel area of U7 (or any other device) it will create hole-electron

pairs that can short the device [4], causing node q to go high if it was not already in a high state. This will set the flip-flop. Data will be corrupted and space flight missions could fail. A radiation hit to the nchannel devices would cause a current pulse to ground, shorting the device and pulling the node low. The current pulses are typically around 1ns in duration.

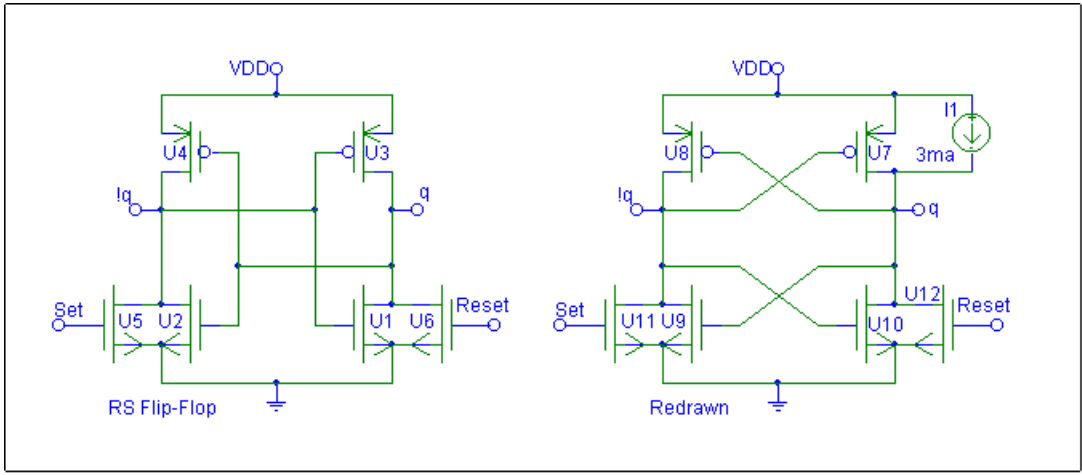


Figure 7. CMOS RS Flip-Flop in Standard and Redrawn Configuration.

One fix for the flip-flop shown in Figure 7 would be to add pass devices as shown in Figure 8. This modification will require two consecutive radiation hits in two different devices before an upset will occur [5]. Using such an RS flip-flop for radiation tolerant designs will allow much more reliable operation in space without having to go to radiation hardened processes. Indeed, the radiation hardened process that uses the

standard RS flip-flop may still be susceptible to SEU's in a radiation environment. The radiation-tolerant circuit of Figure 9 can be manufactured in standard CMOS foundries, providing less expensive space flight hardware. The Set and Reset now drive two nodes each and the !Reset is an inverted signal. The output, q, uses a buffer to restore the high and low signals.

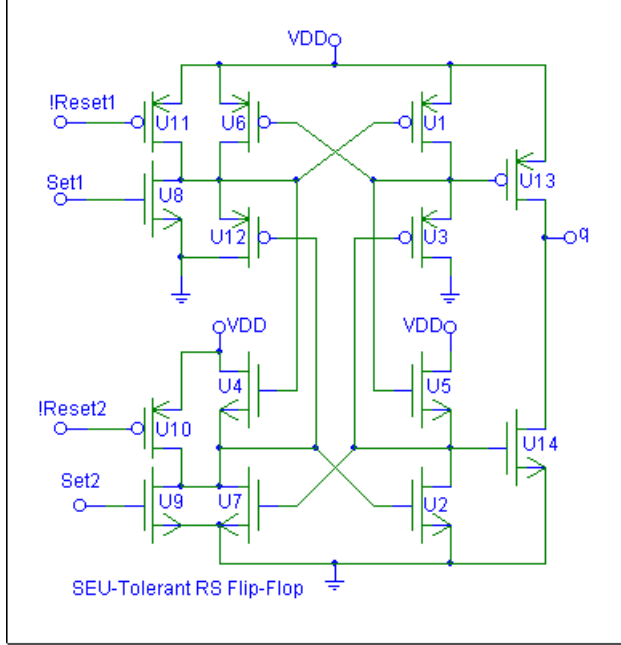


Figure 8. A Radiation Tolerant Set-Reset Flip-Flop.

6. CONCLUSIONS

A synthesis method for creating asynchronous circuits using RS flip-flops for the sequence memory has been given. This method is used to create asynchronous sequential circuits easily by using an RS flip-flop.

BIBLIOGRAPHY

- [1] David A. Huffman, "The Synthesis of Sequential Switching Circuits", *Journal of the Franklin Institute* 257 (3), pp. 161-190, 257 (4), pp. 275-303, 1954.
- [2] M. Morris Mano, "Digital Design", Second Edition, Prentice-Hall, 1991.
- [3] Stephen H. Unger, "Asynchronous Sequential Switching Circuits", John Wiley & Sons, Inc., 1969.
- [4] T. P. Ma and Paul V. Dressendorfer, "Ionizing Radiation Effects in MOS Devices and Circuits", John Wiley & Sons, 1989.
- [5] J. Canaris and S. Whitaker, "Circuit Techniques for the Radiation Environment of Space", *IEEE 1995 Custom Integrated Circuits Conference*, pp. 5.4.1 – 5.4.4, 1995.