

Complete Synthesis Method for D Flip-Flops with Set and Reset Inputs

David F. Cox
Research Associate Professor
dcox@mrc.unm.edu
(505) 272-7045

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

Abstract – A rigorous synthesis method is given that allows the design of asynchronous circuits using only D flip-flops. The two-state realization of the transition flip-flop which uses the clock and Data inputs is made use of in the state assignment step. Adding Set and Reset control to the two transition state variables is seen to be accomplished with some added logic. The Set and Reset inputs allow complete coverage of any asynchronous flow table that does not fall into the state constraints on the two-state realization of the Data and clock inputs. The resulting equations, however, can require iteration to solve.

1 Introduction

Asynchronous design was given a formal methodology by David Huffman in 1954 [1]. This paper will discuss extending his techniques to designs using flip-flops as the feedback elements instead of combinatorial circuits. Prior work [2] has been extended to show how the Set and Reset inputs on a D flip-flop can be included with the Data and Clock inputs in realizing asynchronous circuits. Asynchronous circuits are being developed for larger systems [3] and synthesis techniques have been lacking for the more complex systems. Work in Petri Nets [4] have improved the state diagram descriptions but actual hardware synthesis methods have not improved much since Huffman [5].

2 RS Characteristic Equation

The characteristic equation for the RS flip-flop has been shown [2] to be given by equation (1). Most RS flip-flop realizations do not fit equation (1) if both Set and Reset are asserted at the same time. To preclude this from happening equation (2) is given as a constraint on the Set and Reset inputs so that during the logic implementation only inputs that meet the constraint are allowed to occur.

$$Q = S + !R \bullet q \quad (1)$$

$$S \bullet R = 0 \quad (\text{constraint}) \quad (2)$$

3 D Flip-Flop

The logic diagram for a D transition flip-flop is shown in Figure 1. The flow table can be generated by using the RS characteristic maps in reverse to generate the next-state equations [2]. The characteristic equations for the D flip-flop are given in equations (3) and (4). The last term in each expression (in brackets) is a cover for hazards and does not need to be included in the synthesis characteristic equations since the feedback is now constrained within the D flip-flop and the design of the flip-flop itself is assumed to be hazard-free.

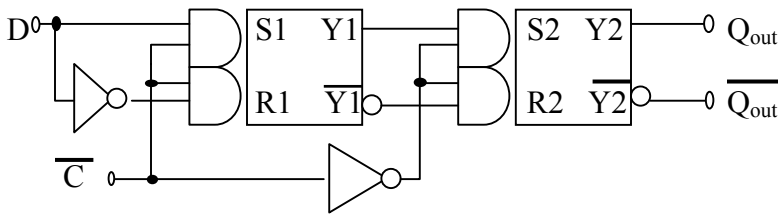


Figure 1. D flip-flop equivalent circuit.

$$Y1 = !C \bullet D + C \bullet y1 + [D \bullet y1] \quad (3)$$

$$Y2 = !C \bullet y2 + C \bullet y1 + [y \bullet 1y2] \quad (4)$$

The complete synthesis equation set for the D flip-flop can then be given. Note that the output of the flip-flop must be Y2. Y1 is internal to the flip-flop and may not be available as an input to some output function. Equations (5), (6) and (8) are used in the asynchronous synthesis procedure. Note that “Y” is the next state and “y” is the present state in a flow table.

$$\text{Out} = Y2 \quad (5)$$

$$Y1 = !C \bullet D + C \bullet y1 \quad (6)$$

$$Y2 = !C \bullet y2 + C \bullet y1 \quad (7)$$

$$\text{Constraint: } Y2 = 0 \text{ if } y1 = y2 = 0, \quad (8)$$

$$= 1 \text{ if } y1 = y2 = 1. \quad (9)$$

It is easy to see from equations (6) and (7) that Y2 must be equal to 0 if both y1 and y2 are 0, and that Y2 must be equal to 1 if both y1 and y2 are 1. The constraints are given in equations (8) and (9).

3 Complete D Expression

Incorporating the Set and Reset inputs need to be done in a way that sets and resets output Y2 with an asynchronous Set or Reset input, respectively. Simultaneous Set and Reset occurrences are avoided during the filling in of the final Karnaugh maps. The complete set of characteristic equations is given by combining the previous equations for the RS and D flip-flops. The logic diagram that gives a minimal realization for the DRS flip-flop is shown in Figure 2. The state variable equations can be written from inspection and reduced to equations (10) and (11).

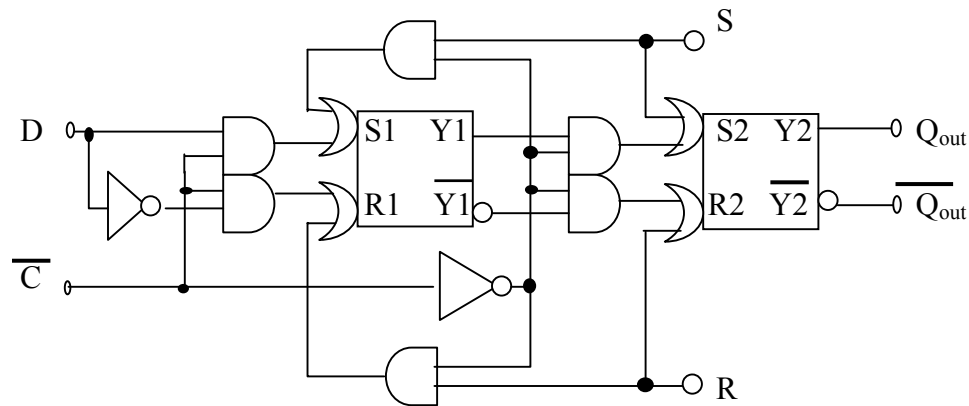


Figure 2. DRS (Data-Reset-Set) flip-flop equivalent.

$$\begin{aligned}
 Y1 &= !C \bullet D + C \bullet S + !(C \bullet !D + R \bullet C) \bullet y1 \\
 &= !C \bullet D + C \bullet S + (C \bullet !R + !C \bullet D) \bullet y1 \quad (10)
 \end{aligned}$$

$$\begin{aligned}
 Y2 &= C \bullet y1 + S + !(R + C \bullet !y1) \bullet y2 \\
 &= S + C \bullet y1 + !R(!C + y1) \bullet y2 \quad (11)
 \end{aligned}$$

The constraint equations can be derived from the SR constraints from the prior section. The Set and Reset map-covering constraint is given in (2). The D flip-flop characteristic equation coefficients have been modified by the Set and Reset inputs but can still be treated in the same fashion. The constraints on the state variables are no longer valid, however, with the Set and Reset inputs allowed to occur. The coverings are then expanded by replacing the originals with the coefficients in (10) and (11). An example will illustrate the procedure.

4 Synthesis Example

Handshake operations usually require an asynchronous communication between systems using different (asynchronous) clocks. A clocked input sampling may be desired, along with an instantaneous reset when the operation is complete. The timing diagram [6] shown in Figure 3 may describe such an operation with an input request (X) being sampled by a local acknowledge signal (Y) that starts a local operation (Q) that can be asynchronously reset when the request has been completed.

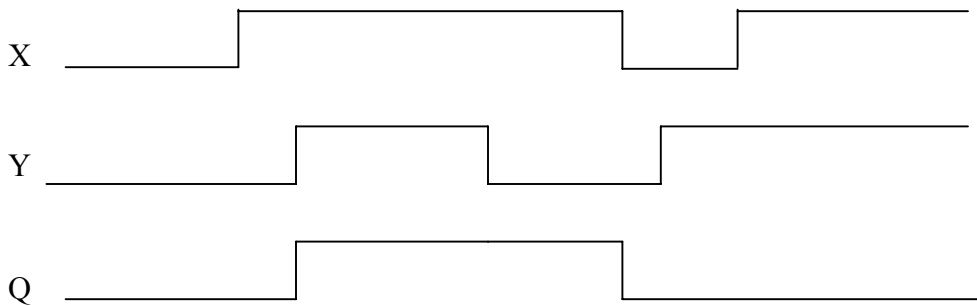


Figure 3. Example timing diagram.

X is clocked by the positive edge of Y to turn Q on. Q will reset asynchronously with X returning to zero.

A flow table describing the timing diagram and the resultant state map are shown in Figure 4. Realization maps for the C, D, S and R inputs to a D flip-flop are derived as shown in Figures 5 and 6.

state	XY				Q
	00	01	11	10	
1	①	①	①	4	0
2	1	1	-	-	1
3	1	2	③	3	1
4	1	-	3	④	0

q1q2	XY				Q = q2 (output)
	00	01	11	10	
00	00	00	00	10	0
01	00	00	-	-	1
11	00	01	11	11	1
10	00	-	11	10	0

Q1 Q2

Figure 4. Flow table and next-state map for timing.

Using equations (10) and (11) we can now find the input maps by extracting the individual state variables from the next-state map [2,7,8]. The only constraint that has to be met is $S \bullet R = 0$. Note that $!C \bullet D + C \bullet S$ must equal "1" in only two locations in the top half of the map since $q1 = 0$ there, and the coefficient of $q1$ will not be able to produce a "1" for the equation. Each coefficient's map can be generated from the maps as shown and are produced just below the map for state variable "Q1" in Figure 5. S and R are set to "0" unless they are absolutely needed in

the realization. Figure 6 shows the maps for the coefficients of equation (11), which is the state variable Q2's equation.

$$\begin{array}{c|cccc}
 & \text{XY} & & & \\
 \text{q1q2} & 00 & 01 & 11 & 10 \\
 \hline
 00 & 0 & 0 & 0 & 1 \\
 01 & 0 & 0 & - & - \\
 11 & 0 & 0 & 1 & 1 \\
 10 & 0 & - & 1 & 1 \\
 \hline
 & \text{Q1} & & &
 \end{array}
 = !C \bullet D + C \bullet S + (C \bullet !R + !C \bullet D) \bullet
 \begin{array}{c|cccc}
 & & & & \\
 \hline
 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 \\
 & 1 & 1 & 1 & 1 \\
 & 1 & 1 & 1 & 1 \\
 \hline
 & \text{q1} & & &
 \end{array}$$

$$!C \bullet D + C \bullet S = \begin{array}{c|cccc}
 & & & & \\
 \hline
 & 0 & 0 & 0 & 1 \\
 & 0 & 0 & - & - \\
 & - & 0 & - & - \\
 & 0 & - & - & - \\
 \hline
 \end{array} ; \quad (C \bullet !R + !C \bullet D) = \begin{array}{c|cccc}
 & & & & \\
 \hline
 & - & - & - & - \\
 & - & - & - & - \\
 & 1 & 0 & 1 & 1 \\
 & 0 & - & 1 & 1 \\
 \hline
 \end{array}$$

Figure 5. State variable Q1's map equations.

$$\begin{array}{c|cccc}
 & \text{XY} & & & \\
 \text{q1q2} & 00 & 01 & 11 & 10 \\
 \hline
 00 & 0 & 0 & 0 & 0 \\
 01 & 0 & 0 & - & - \\
 11 & 0 & 1 & 1 & 1 \\
 10 & 0 & - & 1 & 0 \\
 \hline
 & \text{Q2} & & &
 \end{array}
 = S + C \bullet \begin{array}{c|cccc}
 & & & & \\
 \hline
 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 \\
 & 1 & 1 & 1 & 1 \\
 & 1 & 1 & 1 & 1 \\
 \hline
 & \text{q1} & & &
 \end{array}
 + !R(!C + q1) \bullet \begin{array}{c|cccc}
 & & & & \\
 \hline
 & 0 & 0 & 0 & 0 \\
 & 1 & 1 & 1 & 1 \\
 & 1 & 1 & 1 & 1 \\
 & 0 & 0 & 0 & 0 \\
 \hline
 & \text{q2} & & &
 \end{array}$$

(Assume S = 0 to see if the other maps can be completed.)

$$C = \begin{array}{c|cccc}
 & & & & \\
 \hline
 & - & - & - & - \\
 & - & - & - & - \\
 & 0 & - & - & - \\
 & 0 & - & 1 & 0 \\
 \hline
 \end{array} ; \quad \text{and, } !R(!C + q1) = \begin{array}{c|cccc}
 & & & & \\
 \hline
 & - & - & - & - \\
 & 0 & 0 & - & - \\
 & 0 & 1 & 1 & 1 \\
 & - & - & - & - \\
 \hline
 \end{array}$$

Figure 6. State variable Q2's map equations.

The maps of Figures 5 and 6 need to be solved simultaneously, which is tedious and needs to be done by trial and error. The results are given in equations (12), (13) and (14). Note that S = 0.

$$D = X \quad (12)$$

$$C = Y \quad (13)$$

$$\& \quad R = !X \quad (14)$$

The D flip-flop connections are shown in Figure 7. A simpler procedure could be used by assuming a realization with only RS flip-flops [8]. The D flip-flop consists of only RS flip-flops but the Reset and Set inputs are constrained in the D, whereas they are free to be assigned in the RS implementation. For this reason it is generally preferred to implement an asynchronous map with RS cells rather than D cells, unless the D flip-flop is the only one available. An integrated circuit has the flexibility of using any cell desired whereas a discrete design may be forced into using D flip-flops.

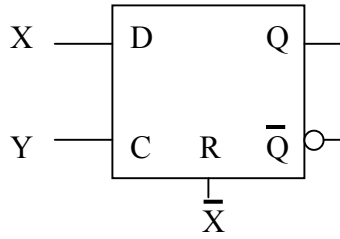


Figure 7. D flip-flop connections for example in text.

An asynchronous implementation using only RS flip-flops would have the Set and Reset inputs for Q1 and Q2 given by equations (15) through (18). The implementation was done using techniques outlined in [2]. The inputs are seen to be simpler and the synthesis was much less tedious in implementing.

$$S1 = X \bullet !Y \quad (15)$$

$$R1 = !X \quad (16)$$

$$S2 = Y + q1 \quad (17)$$

$$R2 = !X \bullet !Y + !X \bullet !q1 \quad (18)$$

5 Conclusions

A synthesis method has been given for implementing asynchronous sequential circuits using only D flip-flops and interconnect logic. The procedure is seen to be fairly complex but rigorous. Some design libraries may include only the D flip-flop, in which case the more complex synthesis procedure may be required. Design techniques have been given for flip-flops that can be used in a radiation environments [9,10]. These techniques may be used in designing lower-power and higher-speed digital systems for use in space systems where radiation is also a problem [9]. Known radiation-tolerant design techniques [10] can be used in the design of the flip-flops to insure robust operation in a space environment.

References

- [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] D.F. Cox, "Further Techniques in Asynchronous Design with One- and Two-State Storage Cells", 8th Annual NASA Symposium on VLSI Design, University of New Mexico, Albuquerque, New Mexico, October, 1999, pp. 2.1.1 – 2.1.9.
- [3] Christoph Heer, "Entwurfsmethode für selbstgetaktete VLSI-Datenpfadarchitekturen", Doktor-Ingenieurs (Dr.-Ing.) Dissertation, University of Ulm, Ulm, Germany, 1995.
- [4] A. Yakovlev, L. Gomes, L. Lavagno, "Hardware Design and Petri Nets", Kluwer Academic Publishers, 2000.
- [5] M. Morris Mano, "Digital Design", Second Edition, Prentice-Hall, 1991.
- [6] Stephen H. Unger, "Asynchronous Sequential Switching Circuits", John Wiley & Sons, Inc., 1969.
- [7] D.F. Cox, "Asynchronous Logic Design with Transition Flip-Flops", IEEE Midwest Symposium on Circuits and Systems, August, 1999, Los Cruces, New Mexico.
- [8] D. F. Cox, "Asynchronous Logic Design with Subcells", 6th IEEE International Conference on Electronics, Circuits and Systems, September, 1999, Pafos, Cyprus.
- [9] T. P. Ma and Paul V. Dressendorfer, "Ionizing Radiation Effects in MOS Devices and Circuits", John Wiley & Sons, 1989.
- [10] 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.