The Use of A Simple Methodology for Flip Flop Conversion as an Aid in Teaching Synchronous Sequential Circuits in a Digital Systems Design Course

Letha Hughes Etzkorn Computer Science Department The University of Alabama in Huntsville

Abstract

Most digital systems textbooks treat the topic of converting one flip flop to another by simply giving the student certain simple conversion circuits, such as the use of an inverter between the R and S inputs of an RS flip flop to form a D flip flop, or tying together the inputs of a JK flip flop to make a T flip flop. However, a more general, but very simple, methodology for flip flop conversion has advantages when used to teach synchronous sequential circuits to students in a digital systems course. The use of this methodology removes a source of student confusion (how did they come up with that circuit in the first place?) and allows the student to practice certain standard flip flop techniques before the student is required to use those techniques in a more general sequential circuit analysis. This paper describes this methodology.

1.0 Introduction

A digital systems design course is required in both electrical/computer engineering and computer science curricula: the ACM curriculum 79 course CS4, and the IEEE Computer Society's model program course SA6 [4]. Synchronous sequential circuit analysis and design are standard topics in such a digital systems design course. A sequential circuit is a circuit in which the output at any time *t* is a function of the external inputs to the circuit, and the internal state of the circuit. In other words, a sequential circuit is a circuit which has memory elements as well as combinational logic elements. In a digital systems design course, the memory elements used in low level synchronous sequential circuits are electronic devices known as flip flops. Typically, the topic of converting one flip flop to another, or more specifically, providing external combinatorial circuitry with a particular flip flop to allow it to be used in place of a flip flop of a different type, is treated fairly

-1-



developed)

lightly in digital systems textbooks. Usually either one or two simple conversion circuits are given to the student, and the student is expected to recognize from the flip flop truth tables how those circuits were derived [1][2][3][4], or the problem of flip flop conversion is treated as an exercise for the student [3][4]. Treating this topic lightly is reasonable since converting one flip flop to another is of limited usefulness in circuit design. However, a flip flop conversion methodology can be an excellent teaching tool to help introduce students in a digital systems course to the use of flip flops in synchronous sequential circuits.

I have developed a simple methodology which provides a systematic method a student can use to convert one flip flop to another. Teaching this methodology in a digital systems design course after flip flops are introduced, but before the use of flip flops in such circuits as sequence detectors is discussed, provides a mechanism by which the student can practice flip flop techniques without worrying about non-flip-flop-related sequential circuit issues such as the development of a state transition diagram (Mealy machine or Moore machine). Also, this methodology removes a source of student confusion: it is usually not clear to the average digital systems student how the flip flop conversion circuits given in the typical digital systems textbook are derived.

2.0 Methodology

This section discusses a methodology that provides a systematic conversion technique to convert one standard flip flop to another: it provides external combinatorial circuitry that allows a particular flip flop (the Flip-flop-you-have) to be used in place of a flip flop of a different type (the Flip-flop-you-want). The steps of this methodology are as follows (the examples shown to illustrate this methodology show a D flip flop being converted into a JK flip flop by the use of external combinatorial circuitry):

- Draw a picture of the desired circuit, with the required external circuitry represented as a grayed-out cloud (to show that it has not yet been developed). See Figure 1 for an example of the initial circuit picture—in this case it is desired to build a JK flip flop using a D flip flop. This initial picture helps prevent the student from confusing the Flip-flop-you-have with the Flip-flop-you-want.
- 2. Write characteristic tables for the Flip-flop-you-want. See Table 1 for the JK characteristic table.

| J | K | Q(t) | Q(t+1) |
|---|---|------|--------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Table 1. Flip-flop-you-want Characteristic Table. In this casea JK flip flop is the Flip-flop-you-want

Now build a table that looks at the Q(t) → Q(t+1) from step 2, and determines what transitions must occur on the inputs of the Flip-flop-you-have in order to cause the desired transition from Q(t) to Q(t+1). In this example (see Table 2), the Flip-flop-you-have is a D flip flop. For this case the D inputs are simply the values of Q(t+1) (whatever the inputs to a D flip flop are before the clock, is what the output of the D flip flop will be after the clock).

| Q(t) | Q(t+1) | D |
|------|--------|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |

Table 2. Required values for the inputs to the Flip-flop-you-have in order to cause the desired transition from Q(t) to Q(t+1).

- 4. Now look at the input(s) to the Flip-flop-you-have as a function of the inputs to the Flip-flopyou-want and Q(t). In this case, D = F(J, K, Q(t)). See Table 3.
- 5. A Karnaugh map (or other circuit reduction technique such as the Quine McCluskey technique, which, however, would be overkill in this case given the limited number of variables under consideration) can be used to reduce the equation for the input(s) to the Flip-flop-you-have as a function of the input(s) to the Flip-flop-you-want and Q(t). See Figure 2.

Replace the grayed-out "cloud" representing the combinatorial circuit that was shown in Figure
1 with the circuitry that implements the reduced form of the equation(s) for the input(s) to the
Flip-flop-you-have.

| J | K | Q(t) | D |
|---|---|------|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Table 3. Inputs of the Flip-flop-you-have as a function of the inputs of the Flip-flop-you-want and Q(t). In this case D = F(J, K, Q(t))

3.0 Conclusions

The example shown in this paper consists of implementing a JK flip flop by using a D flip flop and some external combinatorial circuitry, by the use of the simple flip flop conversion methodology discussed in this paper. In one sense this is a good example, since it shows a flip flop conversion that can be fairly difficult to determine by the simple examination of flip flop truth tables. However, in another sense another flip flop conversion, such as using an SR flip flop to implement a JK flip flop, would be a better example to teach students in a digital systems design course how to handle the values on a flip flop's inputs that cause the desired transitions from $Q(t) \rightarrow Q(t+1)$. For example, on an SR flip flop, if $Q(t) \rightarrow Q(t+1)$ has the transition $0 \rightarrow 0$ then the SR inputs required would be 0 d (where the d represents a don't care). In this case there would be two columns for the inputs to the Flip-flop-you-want in step 4, Table 3, and this would result in two K-maps, one for S and one for R in step 5, Figure 2.

The methodology described in this paper has been used over a period of five years in a digital systems course here at the Computer Science department in the University of Alabama in Huntsville. The observed advantages of this method as a teaching aid are as follows:



Figure 2. Karnaugh map to reduce the equation(s) for the inputs to the Flip-flop-you-have

- 1. It provides a method for students to practice the flip flop techniques that are typically used in synchronous sequential circuits such as a sequence detector, without having to worry about the state transition diagrams, etc., that are required when implementing a sequence detector.
- 2. After the student implements a particular flip flop (the Flip-flop-you-want) using another flip flop (the Flip-flop-you-have) and external circuitry, the student can easily test the circuit by tracking it through the characteristic table of the particular flip flop implemented (the Flip-flop-you-want). Thus the student is able to check his or her work to determine if he or she is handling the flip flop techniques appropriately.
- 3. It gives the students additional practice in the use of Karnaugh maps before the student must address implementing a sequence detector.
- 4. The methodology is easily taught in a single class period, after flip flops have been introduced.

The only negative observation about the use of this methodology, is that if the student is given a problem on a test or homework in which he or she is required to convert one flip flop into another

flip flop, care must be taken in wording the problem. There is a tendency for some students to "read the problem backward," and mix up the Flip-flop-you-want with the Flip-flop-you-have.

References

- [1] Mano, M.M., Digital Design, Second Edition, Prentice-Hall, Englewood Cliffs, NJ, 1991.
- [2] Greenfield, J.D., Practical Digital Design Using ICs, John Wiley and Sons, NW, 1977.

[3] Shiva, S.G., <u>Computer Design and Architecture</u>, Second Edition, HarperCollins Publishers, Inc., 1991.

[4] Shiva, S.G., <u>Introduction to Logic Design</u>, Second Edition, Marcel Dekker, Inc., New York, NY, 1998.