

# Optical Character Recognition Using Fuzzy Logic

by William A. Gowan

## OVERVIEW

This application note shows how to envision, describe, and realize a design using fuzzy logic. It is not intended to be an introduction to fuzzy logic, but it is basic enough to be understood by designers with a cursory understanding of the subject. For those who seek an introduction to fuzzy logic, the Motorola Fuzzy Logic Educational Kit is an excellent primer. Other sources of information are shown in the list of references at the end of this document.

Fuzzy logic facilitates design of systems that mimic human reasoning. A fuzzy system accepts data input from sensors, then makes decisions based on that input. In most cases, these decisions are the basis for a control system. However, a fuzzy rule-driven system can simply be a classification engine that draws distinctions between and labels differing types of input data.

This note explains the design of an optical character recognition engine called the Optical Character Associator (OCA). Optical character recognition systems must classify optical inputs as specific letters, numbers, or other characters, and are thus ideal candidates for fuzzy logic implementation. OCA is a classification engine that recognizes the set of fourteen characters used by the US banking industry to encode account numbers along the lower edge of checks. The engine is implemented using an MC68HC11E9 8-bit microcontroller, although it could have been implemented using devices from the M68HC05, M68HC16, or M68300 MCU families.

OCA accepts input from a 64 x 1 pixel charge-coupled device (CCD) sensor. After an input preprocessor program formats sensor data into an easily "fuzzifiable" structure, the inference engine fuzzifies the data, applies the fuzzy rule set, and generates an output that corresponds to the character being read.

This application note presents OCA design methodology, and defines all input variables, fuzzy rules and output variables. Although preprocessor operation is fully described, and internal variables used by the preprocessor are explicitly defined, a designer must provide the actual preprocessing code in order to implement the system described. System resources not directly related to the optical portion of the system, such as motor transport for document movement beneath an optical sensor, must also be provided.

OCA was designed from simulated sensor data input. In order to implement a physical system, the simulated data should be carefully compared to actual character reads from the sensor to be used. Modification to fuzzy membership functions and rules may be required. The operation of OCA was verified using simulated testing as described in **TESTING RESULTS**. Motorola does not guarantee the operation of the software described in this document.

## PROBLEM DEFINITION

The industry definition for the character set to be recognized appears in **Figure 1**. There are fourteen valid characters — numeric characters zero through nine, and four special characters, SS1 through SS4.



AN1220  
OCR CHAR

**Figure 1 Character Set To Be Recognized**

Each character is right-justified in a 125 mil wide frame. Other characters cannot intrude into the frame. The widest characters, 0, 8, SS1, SS2, SS3, and SS4, have a specified width of 91 mils. Characters 4, 6, and 9 are specified at 78 mils wide. Characters 3, 5, and 7 are specified at 65 mils. Characters 1 and 2 are specified at 52 mils. Because of the differing specified character widths, there is a variable amount of white space to the left of each character in a string of characters.

The optical sensor chosen for this design is the Texas Instruments TSL214. This device consists of 64 vertically-aligned CCD elements. Each pixel is 4.72 mils wide and 2.756 mils high—a data "slice" is 4.72 mils wide and 319 mils high. Since maximum character height is 117 mils, there is a pixel area of approximately 100 mils above and below each character.

Spacing between data slices is determined by the relationship between the width of one data slice and the width of a character, or more specifically, the width of the narrowest line segment of a character.

The width of the narrowest line segment of a character (for instance, the thin vertical line that forms the left side of the character 0) is 13 mils. To insure detection, a minimum of two slices must be taken in the width dimension of any segment. If two slices were not taken, a line segment could straddle two data slices and thus not be detected. Data slices need not touch each other, but the gap between them must be small. Dividing a 125 mil frame into 22 vertical slices yields a spacing between data slices of 5.68 mils. This spacing insures that at least two samples are taken in a 13 mil wide character element.

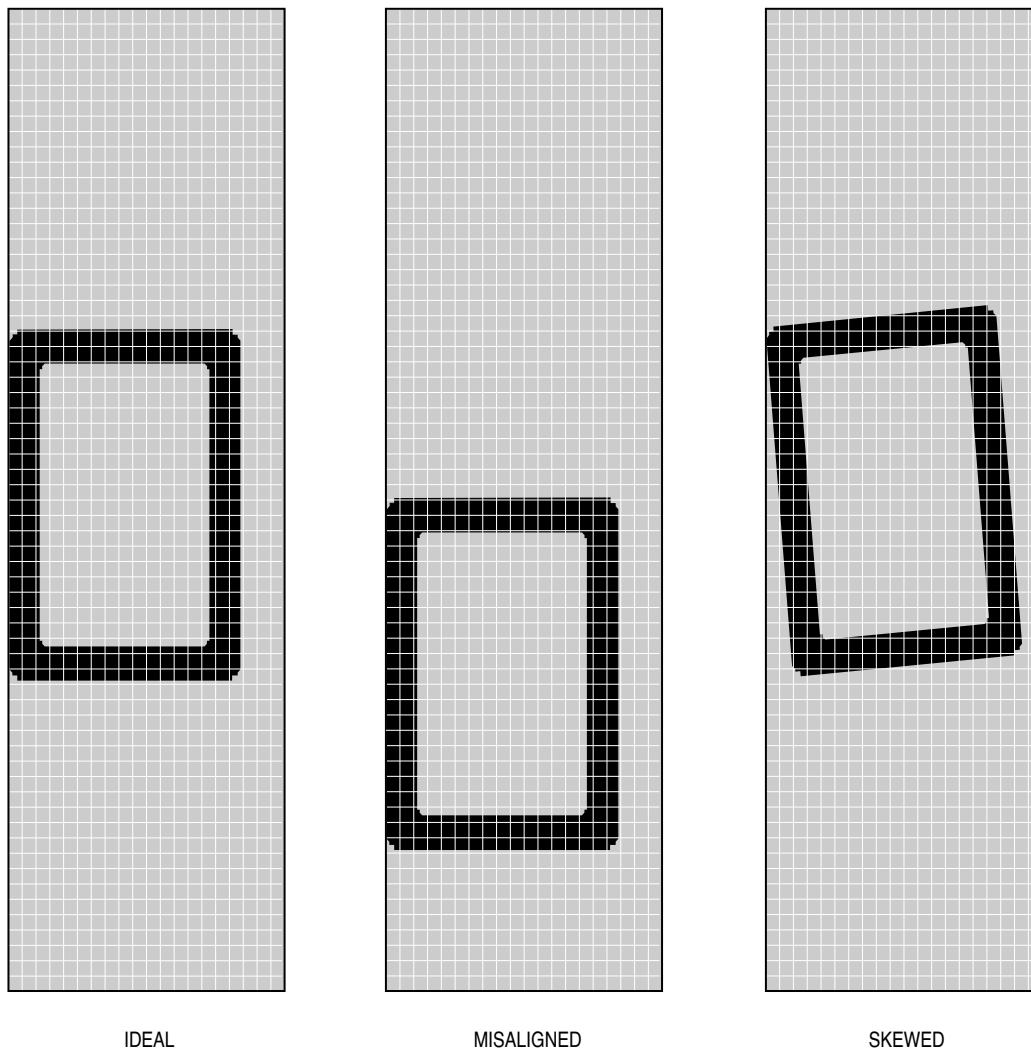
The TSL214 has a specified integration time, measured in ms, which is a function of light intensity. Satisfying the integration time specification allows every CCD in the device to respond to the light level striking it. For light intensities ranging from 15 to 42  $\mu\text{w}/\text{cm}^2$ , an integration time of 6 ms is adequate.

After integration time has elapsed, data may be read out of the optical sensor serially as analog values. When the sensor SI input is enabled, sensor output voltage represents the analog value from CCD#1. Upon the next clock transition, the output becomes the analog value from CCD#2, and so on until all 64 pixels have been read out.

Since sensor data is produced in the form of an analog value, an MCU A/D converter channel can be used to read the value in. In this high-contrast application, it is also possible for sensor output to be read as serial digital data, provided that saturated CCD output is greater than TTL  $V_{IH}$  and unsaturated output is less than TTL  $V_{IL}$ . Backlighting the document being scanned with a bright red LED provides high contrast.

At this point the problem can be defined. Hardware must provide a mechanism to light the document and move it under the sensor in 5.68 mil increments. The microcontroller receives a 64-bit stream of values for each slice. From this data, the classification engine must correlate contiguous data slices against the labels of recognizable characters.





AN1220  
OCR SLICE

Figure 2 19 Data Slices for Ideal, Misaligned, and Skewed Character 0

Table 1 Slice Totals for the Three Readings of Figure 2

Slice	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Ideal	23	23	5	4	4	4	4	4	4	4	4	4	4	4	23	23	0	0	0
Misaligned	23	23	5	4	4	4	4	4	4	4	4	4	4	4	23	23	0	0	0
Skewed	6	21	21	8	5	5	6	5	5	5	5	4	4	4	12	23	16	3	0

**PREPROCESSOR OUTPUT: THE TRANSITION CONCEPT**

The data in **Table 1** provides a useful insight. It is apparent in all three cases that the magnitude of the slice total increases to a high value of approximately 23, decreases to a low value of approximately 4, increases again to a high value of approximately 23, and then finally decreases to zero. **Figure 3** is a plot of the slice total for the three cases. Even though the bit patterns and slice totals are different, plots of the slice totals have the same shape.



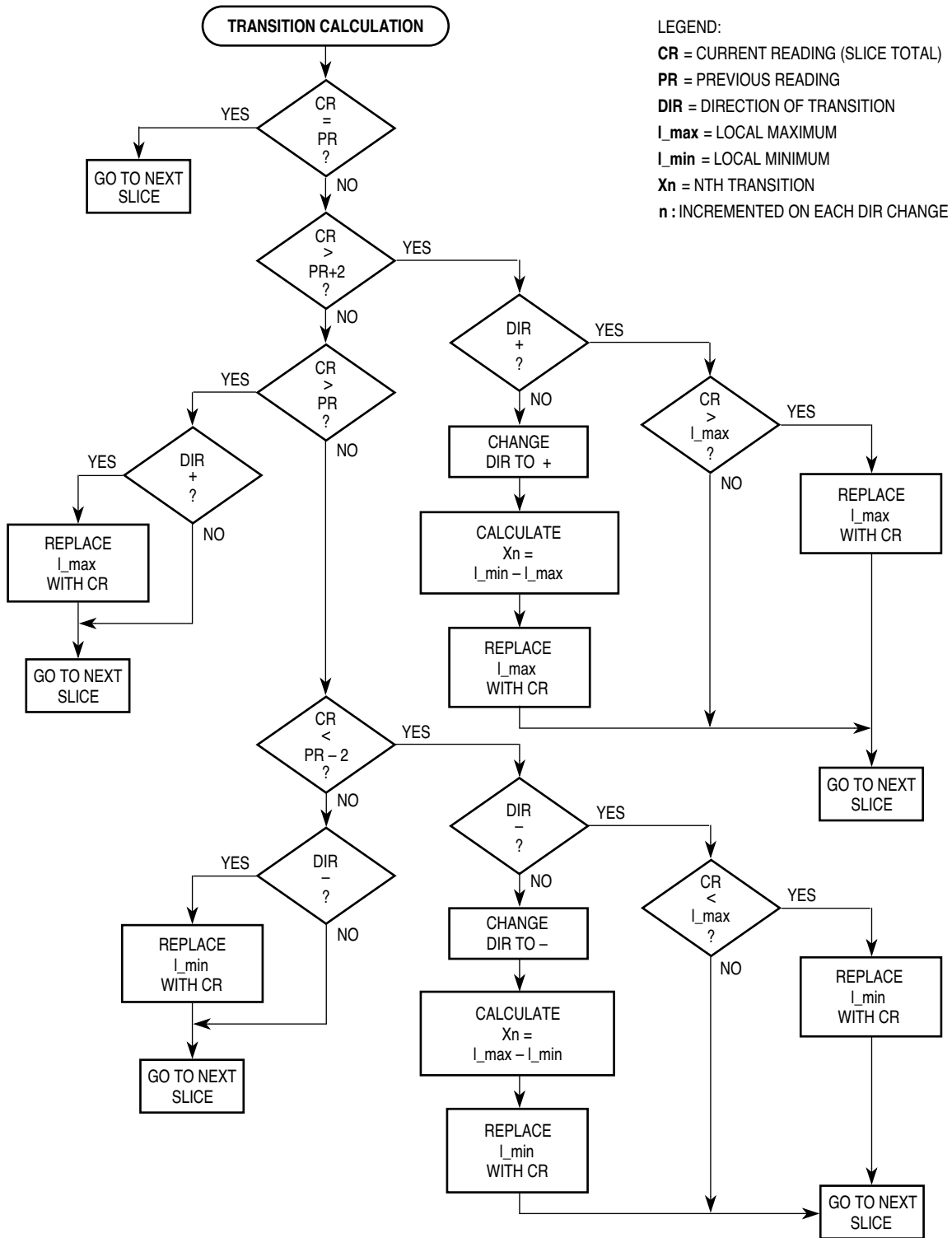


Figure 4 The Preprocessor for Transition Calculation

AN1220  
OCR FLOW



























