

Computer Architecture
Individual Project
BCD to 7 Segment Decoder

Academic Year: 2007-2008

University of Urbino "Carlo Bo"
Information Science and Technology Institute
Applied Computer Science
Worldwide Distance Degree Program

Grader: Professor Alessandro Bogliolo

Author: Student Antonio Merola
ID: 231050

Abstract

This is the required paper (individual project) to access to the oral exam of Computer Architecture of the distance degree program in applied computer science from University of Urbino in Italy. It consists of a Logic-level design and simulation of a circuit made with TkGate, a graphical editor and event-driven simulator for digital circuits with a tcl/tk-based interface.

1 BCD

The goal of this paper is to deliver a project about a BCD to 7 Segment Decoder (a code converter); a Logic-level typology report where is described how the BCD input data is trasformed to code able to drive a seven segment display. A proper way to start this topic, is an appropriate explanation of terms within the title of this paper; basically first of all would be appropriate to introduce the BCD.

BCD stands for **B**inary-**C**oded **D**ecimal and is a way to encode decimal numbers where each digit corresponds to a binary sequence.

The BCD encoding is considered *irreduntant* hence each element is assigned with a unique word, is of a *costant lenght*, hence all code words are of the same lenght, finally is *exact*, hence all elements are encoded and there are no elements associated with the same word. It is worthwhile to remember that a byte consists of a binary word of 8 bit and a group of a binary word of 4 bit is usually referred as nibble. A BCD digit is usually represented by a nibble which represent the decimal digit in the range 0-9:

| | Note: BCD Encoding | | | | | | | | | |
|---------|---------------------------|------|------|------|------|------|------|------|------|------|
| Decimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| BCD | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 |

The minimum number of digits of a constant length exact encoding of a set of M elements is: $N = \lceil \log_2 M \rceil$ hence $2^N \geq M$ in this scenario we have $2^4 = 16$ dealing with an encoding of a finite sets. The reader might notice that rather than 16 values only 10 output values are considered (range 0000-1001), the other 6 combinations are not considered.

Industry available circuits, relies on the overall 16 values introducing also additional controls or just other conditions in order to deliver a more featured product. This would change the truth table hence the circuit, however the applied discussion here reported remain more less the same.

2 Seven Segment Display

A seven segment display is a display device composed of seven elements that can be illuminated to produce a representations of the 0-9 digits. The seven segments are arranged as showed in Figure 1:

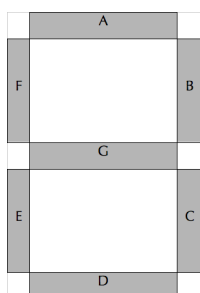


Figure 1 – the individual segments of a seven-segment display

Each of the numbers 0-9 may be represented by two or more different segments of the display; for instance number 3 is obtained illuminating segments A, B, C, D and G as showed in Figure 2:

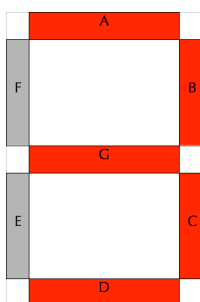


Figure 2 – the representation of number 3 with a 7 segments display

3 Decoders

In a digital system, is possible to convey both instructions or numbers by means of binary levels or impulse signal train. If for instance 4 bits of a message are utilized to convey commands, there are 16 different instructions available; the output of this operation is referred as binary encoded information. Sometimes it might be necessary that a system works with this binary information, hence for each of the 16 available input code it has to stimulate one output. The operation to identify a given code is referred as decode process, or rather a decoder is the opposite of a coder, it convert the n-bit input binary code into a single output. Basically there are two type of decoders: code converters and commutators, code converters transform determinate information code into the same information by means of another code as in the case of the BCD to seven segment where the BCD input data code is trasformed into a valid code able to drive a seven segments display.

4 Truth Table

In order to obtain a truth table to build the seven segment logic circuit, a complete view of all segments is needed, basically in *Figure 3* all numbers are showed to better understand segments involved, that need to be turned on:

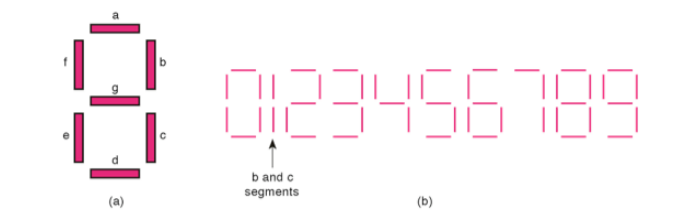


Figure 3 – seven segment display output

This means that, for instance, to visualize the number 0 on the display, all segments but g are to be turned on, while to obtain number 1 only segment b and c and so on; with this informations it is possible to obtain the truth table which is reported in the following table:

| Input | | | | Output | | | | | | | N. |
|-------|---|---|---|--------|---|---|---|---|---|---|----|
| D | C | B | A | a | b | c | d | e | f | g | |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 2 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 3 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 4 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 5 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 6 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 7 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 8 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 9 |

Basically we are dealing, with a combinatorial block logic with 4 input and 7 output, that have to become 7 boolean functions made of 4 input and 1 output. Yet, to reach the goal is necessary to work out the logic for the gates that would be necessary to produce such an output; below the 7 functions (the canonical form) expressed as SOP (Sum Of Products) are reported:

$$a = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}BC\overline{D} + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} + A\overline{B}\overline{C}D + A\overline{B}CD$$

$$b = \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BCD + A\overline{B}\overline{C}D + A\overline{B}CD + AB\overline{C}D + ABCD$$

$$c = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}BC\overline{D} + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} + AB\overline{C}\overline{D} + AB\overline{C}D + ABC\overline{D}$$

$$d = \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BCD + A\overline{B}\overline{C}D + A\overline{B}CD + AB\overline{C}D$$

$$e = \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BCD$$

$$f = \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BCD + A\overline{B}\overline{C}D + A\overline{B}CD$$

$$g = \overline{A}\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}\overline{D} + A\overline{B}C\overline{D} + \overline{A}BC\overline{D} + \overline{A}B\overline{C}D + A\overline{B}CD$$

These SOP's can be minimized in order to get the minimum number of literals, this by means of boolean functions properties or easier using Karnaugh maps; there are also some tools to minimize boolean function. Let's consider, for instance, the segment e (the shortest one) in order to show how to handle the function and came to the circuit. Considering:

$$e = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}BC\overline{D}$$

this function became:

$$e = \overline{A}\overline{B}\overline{C}(D + \overline{D}) + \overline{A}B\overline{D}(C + \overline{C}) = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{D}$$

following in Figure 4 the Karnaugh map is showed and in Figure 5 the logical circuit of the function e, even as a module, where is possible to observe that with all inputs set to 0 the led representing segment e is illuminated as expected:

| | | | | | | |
|---|---|---|---|---|---|---|
| | | 0 | 1 | 1 | 0 | A |
| | | 0 | 0 | 1 | 1 | B |
| 0 | 0 | 1 | 0 | 0 | 1 | |
| 0 | 1 | 0 | 0 | 0 | 1 | |
| 1 | 1 | 0 | 0 | 0 | 0 | |
| 1 | 0 | 1 | 0 | 0 | 0 | |
| D | C | | | | | |

Figure 4 – Karnaugh map

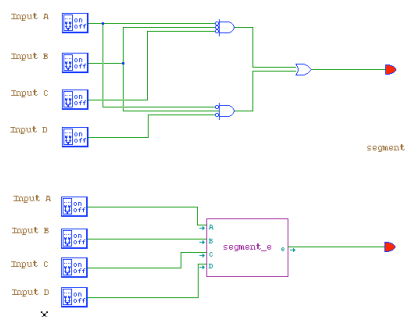


Figure 5 – tkgate: logic circuit of segment e

Following the minimization of the others segments:

$$a = B\overline{D} + A\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C} + \overline{B}\overline{C}\overline{D}$$

$$b = \overline{C}\overline{B} + \overline{A}B\overline{D} + A\overline{B}\overline{D} + \overline{C}\overline{D}$$

$$c = A\overline{D} + \overline{B}\overline{C} + \overline{C}\overline{D}$$

$$d = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{D} + B\overline{C}\overline{D} + \overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D}$$

$$f = \overline{B}\overline{C}\overline{D} + \overline{A}B\overline{D} + \overline{A}\overline{C}\overline{D} + \overline{B}\overline{C}\overline{D}$$

$$g = \overline{A}B\overline{D} + B\overline{C}\overline{D} + \overline{B}\overline{C}\overline{D} + \overline{B}\overline{C}\overline{D}$$

It is worthwhile to highlight that this kind of circuits can be realized with NAND gates, and considering also “do not care” conditions as valid input values, would be possible to reduce the final circuit optimizing area, power and performance hence finding the best implementation of a given specification. However this is not this case, here going step by step every output is made of AND, OR and NOT gates, in the end the complete logic circuit will be the one of Figure 6:

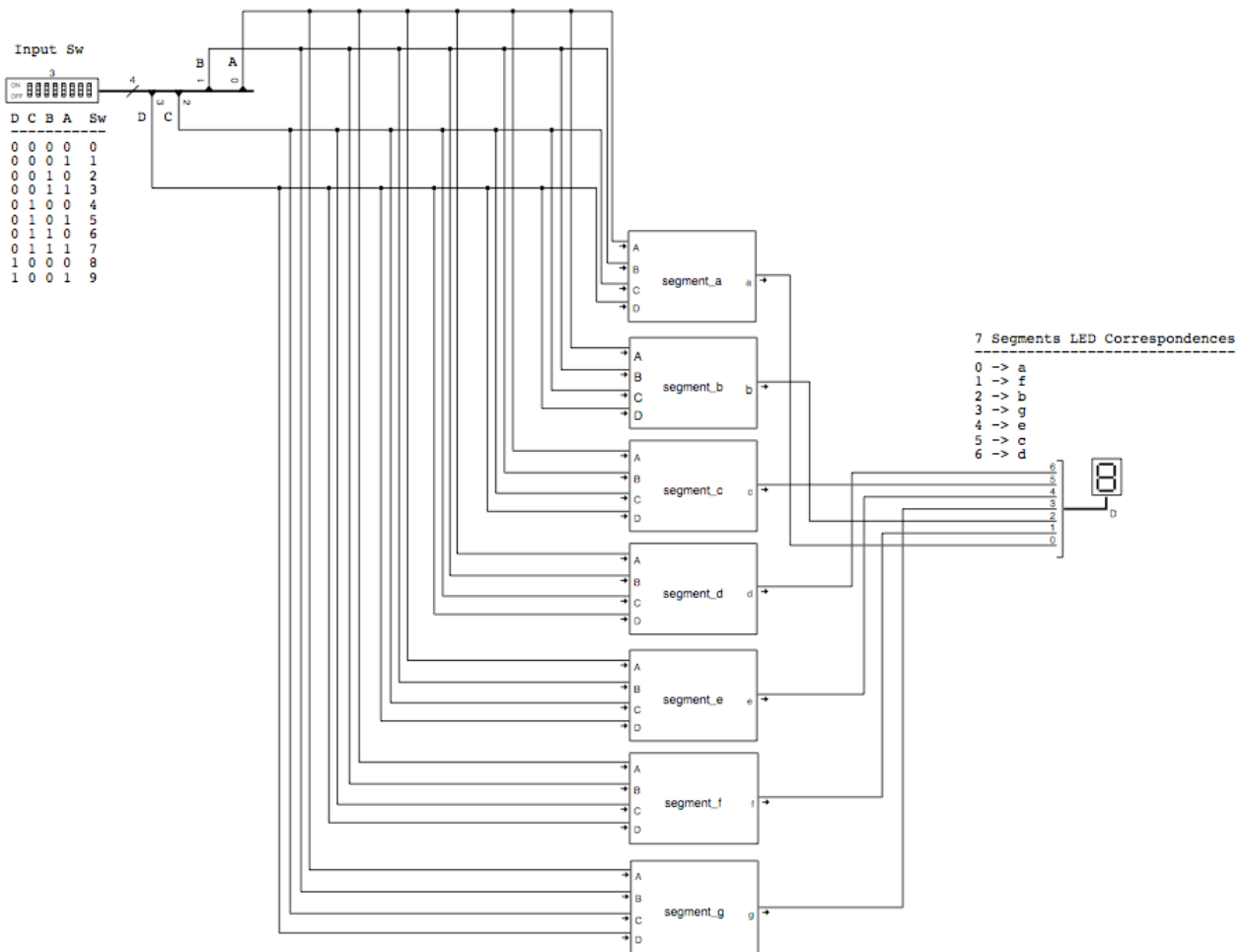


Figure 6 – tkgate: logic circuit of BCD to 7 segments display

The attached *txt* file named *readme.txt* explain how to use tkgate in order to validate the project, while the file named *7segments_module.pdf* shows the circuit module details of each segment of the display obtained with the printing feature of tkgate.