SOLUTIONS TO MINIMIZE THE PORT LINES NUMBER FOR KEYPAD INTERFACES TO MICROCONTROLLERS

Alin Dan POTORAC

University of Suceava, Department of Computer Engineering, 9 University Street, 5800 Suceava, Romania, E-mail: alinp@eed.usv.ro

Abstract. The paper is describing some solutions for efficient use of microcontrollers' port lines interfacing with keypads. Starting from classic matrix structure, some new solutions are proposed in order to obtain the maximum number of keys with minimum costs. Mathematical functions are generated to evaluate their extreme values. Also, compared results are shown. Based on this evaluation for unidirectional port lines some optimal solutions are also suggested for bidirectional port lines, using the available two way binary transfer.

Keywords: digital design, microcontrollers, interfacing, keypad

INTRODUCTION

Connecting keypads to microcontrollers' ports is already a classical problem in digital design. Many solutions were proposed related to the ports types and their futures. As is shown bellow, starting with simple reading of a switch state and passing through the well known matrix keypad complex solutions could be implemented based on modern microcontrollers.

Because of the complexity of actual digital systems, larger amount of switches and buttons are used like human interfaces. Increasing the number of buttons involve costs which have to be taken into consideration in order to elaborate an optimal design. There is a point when the designer has to estimate if the best solution is to add a new or powerful microcontroller or to expand the interfacing possibilities using specific techniques. The motivation of this study is directly connected with the necessity of finding optimized solution to connect a large number of buttons to interface with microcontrollers systems at low costs.

CLASSICAL SOLUTION

The classical way of connecting a large number of buttons to a microcontroller port is to place them in the nodes of a matrix (figure 1).

Having unidirectional port lines split in inputs and outputs always they can be connected in a matrix shape. If B_1 is the number of buttons, n is the number of the port lines and x is the lines (or columns) number, than

(1) $$B_1 = x \cdot (n - x)$$

To find the maximum of B_1 like x function, the derivate has to be calculated.

(2) $$\frac{dB_1}{dx} = n - 2x; \quad n - 2x = 0 \Rightarrow x = \frac{n}{2}$$

So, as square the matrix is as the optimization is better.

For a usual 3 x 4 keypad matrix, 7 port lines are involved, 3(4) as inputs and 4(3) as outputs. The software reading algorithm means to activate only one output line and to read the input lines, each 7 bits binary word being associated with a key.

This solution is the cheapest way of interfacing a large number of keys with a microcontroller but sometimes the maximum available port lines are not covering the necessities and alternative solutions has to be implemented.

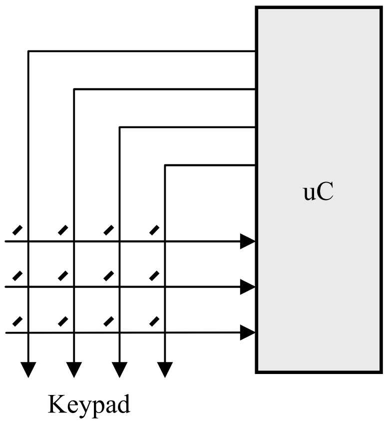

Figure 1. Basic keypad interface
using two unidirectional ports

MINIMIZING PORT LINES NUMBER FOR UNIDIRECTIONAL PORTS

Decoder/Encoder Solution

Having only unidirectional port lines the problem which has to be further solved is how to obtain increased number of keys at low costs.

The first and simplest way is to use a decoder which is a very cheap device. Based on the fact that the software reading algorithm means to activate only one output line at the time, a decoder could be used to expand the number of output lines (figure 2).

With the same notation as before, the possible number of keys is

(3) $$B_2 = 2^x \cdot (n - x)$$

Now, the problem is how to choose x to obtain the maximum keys number.

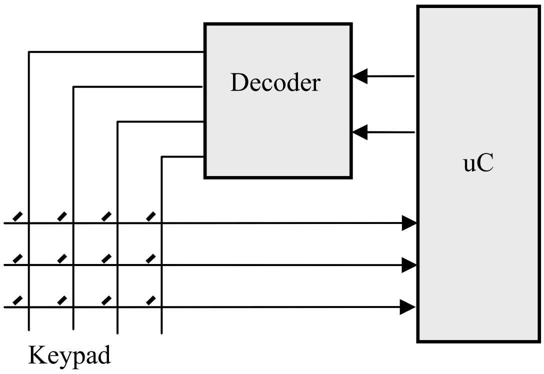

Figure 2. Keypad decoder interface for unidirectional ports

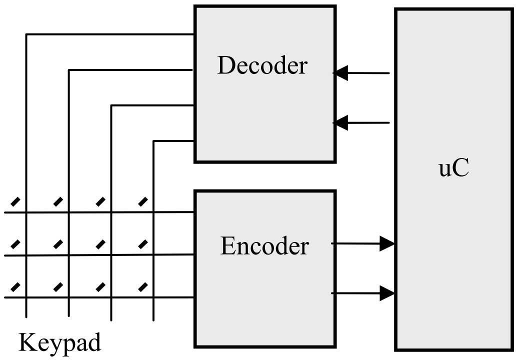

Figure 3. Decoder/Encoder optimized keypad interface for unidirectional ports

The maximum value of B_2 can be determinate by calculating dB_2/dx but the solution is not an evident one. Another way to evaluate x is based on observation that the maximum occur when the number of columns is equal (or as close as possible) with the number of lines - see condition (2), so

(4) $$2^x = n - x$$

is the maximum condition. Solving the equation in a graphical way, for $n=5$ the result is $x=3$ and the maxim of B_2 is 16.

The expanding solution described above can be also implemented for input lines if an encoder is used. Also, both, decoder and encoder can be used (figure 3).

For the structure from figure 3 the keys number could be calculated as:

(5) $$B_3 = 2^x \cdot 2^{n-x} = 2^n$$

Now, the maximum number of keys is not dependent anymore with the way of splitting the lines (x parameter), but only with the number of the port lines, n.

Using decoder/encoder solution is quite cheap but some situation can't be well managed, for example if pressing two buttons in the same time (*shift* like). If more than one key is expected to be pressed at the time, to avoid coding errors, a priority encoder can be used to offer higher priority for priority keys.

Shift Register Solution

A more interesting solution is based on shift register (or a decoded outputs Johnson counter), as shown in figure 4. Usually only to outputs are required, for *clock* and *reset*. Reset is controlling the initial state while the number of clock pulses decides the actual state of the counter and, like that, what output is activated every moment.

The number of allowed keys is

(6) $$B_4 = M \cdot (n-2)$$

where M represents the number of the cells of the used register/counter.

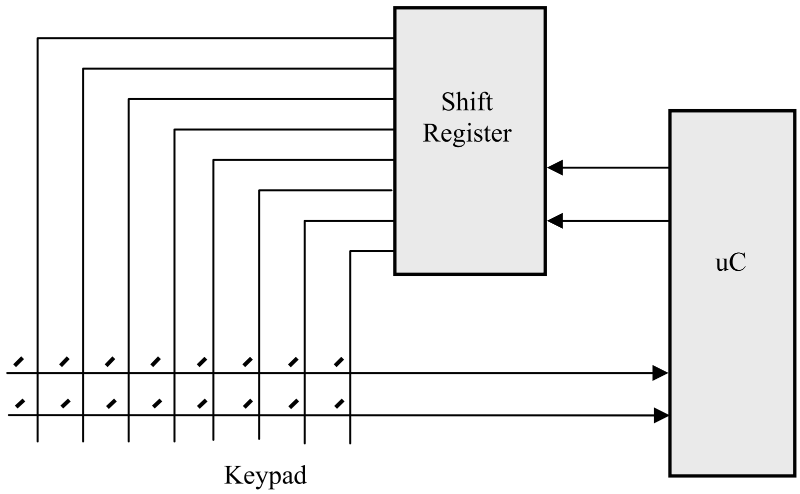

Figure 4. Optimized keypad interface with shift register

This structure allows having a large number of buttons. Comparing with above solutions, if having 5 port lines, 2 outputs and 3 inputs and an usual 8 cells register, the number of keys which can be managed is 24, more than any other solution.

SUGGESTIONS FOR OPTIMIZATION OF BIDIRECTIONAL PORTS INTERFACES

The new microcontrollers' generations are including bidirectional port lines, [4], [5]. This facility offers the possibility of exploiting the two way communication of each line, for example by writing out data into a latch register and than reading the latch outputs via the keypad matrix (figure 5).

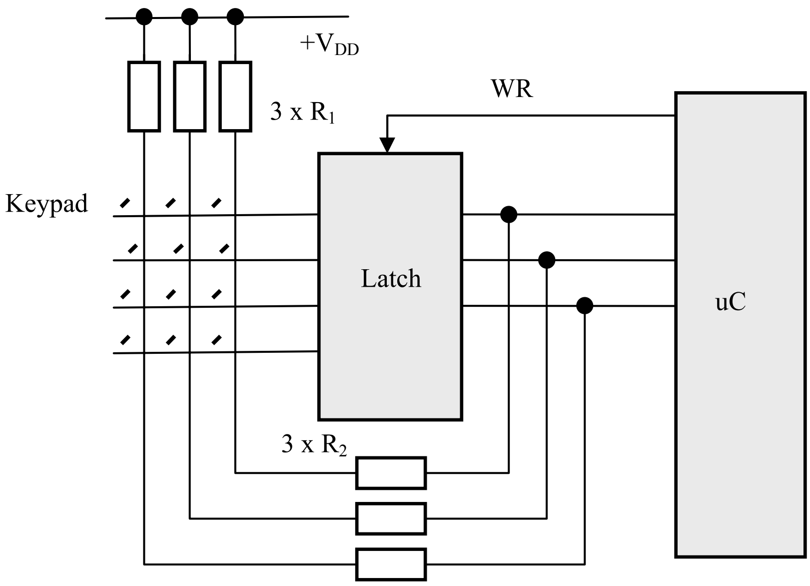

Figure 5. Keypad interface optimized for bidirectional lines port

Once the latch is up-loaded with the binary word having only one active line ("0" for figure 5, but id depending on the algorithm), the port lines direction are reversed. The data is read from the column lines. If no key is pressed, "1111" is read, if a key is pressed, the corresponding line goes to zero.

The R_2 resistors are necessary to avoid the latch outputs feedback to the inputs while the load process is occurring. They introduce as much impedance value as necessary to assure the logical priorities for the logical outputs of the microcontroller port. R1 resistors are pull-up type and can be omitted if internal microcontroller pull-up resistors are available.

Pressing two keys at the time is also possible if write-to-latch and read processes are speed enough.

The maximum number of the controlled keys is

(7) $$B_5 = (n-1)^2$$

We can observe that for 5 bidirectional lines port 16 keys are available.

The efficiency of this principle can be significant improved if an addressable latch is used instead of the direct one. Figure 6 is shown this solution. This time the maximum number of controlled keys is significantly increased as the value is

(8) $$B_6 = 2^{n-2} \cdot (n-1)$$

That is because 2 output lines are reserved for WR signal and DATA signal, so only *n-2* are available as output while when lines are working like inputs only WR line has to be left out to avoid overwriting by WR accidentally activation from the keypad.

Let's observe that for the same 5 lines like before the maximum number of controlled keys is 32.

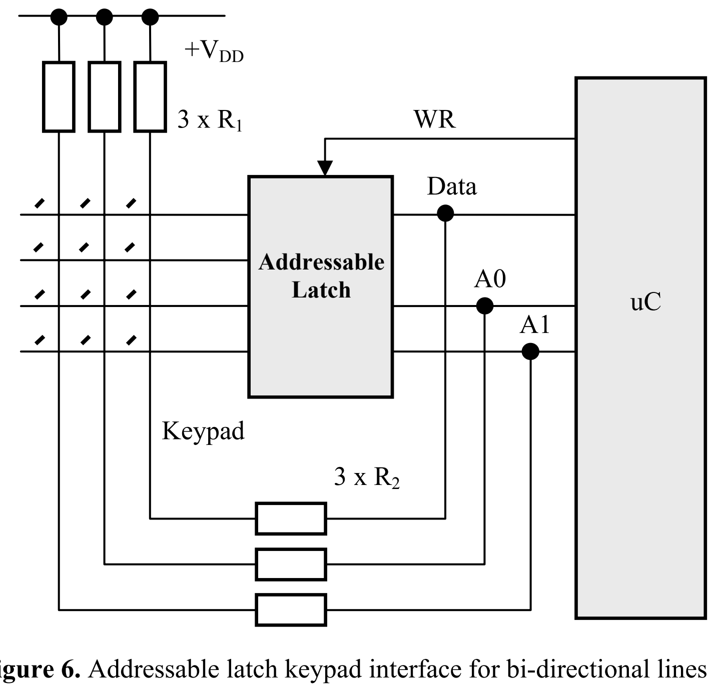

Figure 6. Addressable latch keypad interface for bi-directional lines port

CONCLUSIONS

Using the best solution to interface the microprocessor systems with human operators via keypads is a permanent challenge. Starting with classical matrix keypad interface some improvements were proposed in order to obtain a better efficiency at the level of the compromise costs – available number of interface keys. Table 1 is presenting few comparative results for the solution proposed in the paper.

Table 1. Comparative results for different types of keypad interfaces

Type	Number of Port Lines	Number of Keys	Average Number of Keys per Line
Simple matrix	3	2	0.66
	5	6	1.20
	7	12	1.71
Decoder	3	4	1.33
	5	16	3.20
	7	32	4.57
Decoder-Encoder	3	8	2.66
	5	32	6.40
	7	128	17.14
Shift Register with 8 Stages	3	8	2.66
	5	24	4.80
	7	40	5.71
Latch	3	4	1.33
	5	16	3.20
	7	36	5.14
Addressable Latch	3	4	1.33
	5	32	6.40
	7	192	27.42

Even based on usual unidirectional port lines interface the possible keys number could be increase by adding cheap additional devices instead of using powerful and more expensive microcontrollers. More, if bidirectional port lines are available than a much better efficiency can be obtained by exploiting the programmable two ways of data directions. As shown in the paper, depending on the digital system structure, many alternative solutions can be implemented. Keypads with large number of keys can be easily implemented.

REFERENCES

[1] Potorac A.D. *The Basis of Digital Design*, Ed.Matrix, Bucuresti, 2002

[2] Velchev V. A keypad Controller for Bi-directional Key Matrix, Microchip “Design for Dollars”, 1997

[3] Taub H., Schilling D.*Digital Integrated Electronics*, Ed.McGrawHill, 1977

[4] Microchip Technical Library CD-ROM, 2002

[5] www.microchip.com