Proceeding of the IEEE 28th
Canadian Conference on Electrical and Computer Engineering
Halifax, Canada, May 3-6, 2015

Anti-Ghost Key Design for a Notebook Keyboard

Chun-Hung Cheng and Ying-Wen Bai
Department of Electrical Engineering, Fu Jen Catholic University, New Taipei City, Taiwan
E-mail: bai@ee.fju.edu.tw

Abstract—**This paper proposes a new anti-ghost key design to prevent any conflict of the locations of the keys of a keyboard in a notebook. Previous notebooks often have an embedded controller (EC) both for typical I/O functions and for the keyboard function. In order to provide a better ghost key prevention, as we mainly expand the matrix of the internal keyboard matrix from 8*16 pins to 8*44 pins, we need to add on a second low-power embedded controller not only to control the internal keyboard independently but also to increase the length of the ghost keys by a wide margin. This design will not cause any conflict if 43 keys are pressed simultaneously and will provide a better notebook keyboard design for multiple-key pressing while playing most game software.**

I. INTRODUCTION

The keyboard of a notebook is usually used to key in data or play a game. Sometimes users may push many buttons simultaneously in order to execute some particular functions. Sometimes users may find, by only pushing 2 keys there is no problem but when pushing the third key the signal is not correctly output. Due to a keyboard key conflict, if a user is typing a report and encounters this kind of situation, then he may type an error. If a user is playing a game, the response caused by pressing the keys may result in a missed action of the desired design function [1]-[3].

There are two aspects of a conflict of locations of a notebook keyboard's keys. First, one should understand the architecture of a keyboard, how the signal of the keyboard operation is judged, and how to avoid pressing a key by mistake. Second, one should understand how to transmit the keyboard signal to the notebook. A good design will not cause the conflict of the locations of the keys; however the connection interfaces of different keyboards may cause different results so that a keyboard may be unable to send each key signal smoothly. If we consider these two aspects together, one can determine the specific conflict problem which is a result of the location of certain keys [4] [5].

The keyboard is classified into several types according to various codes. Typically, the encoded keyboard cannot be encoded simultaneously by two kinds of codes. Usually when a user presses keys, the keyboard controller will produce an ASCII code by means of a scanning mechanism. A keyboard design arranges the keys based on a keyboard matrix to encode all keys of the keyboard. When a key is pressed, the keyboard controller will decide which key is pressed and based on the code arrangement of the keyboard matrix code sent out by the pressed key [6].

II. KEYBOARD OPERATION AND GHOST KEY PROBLEM

How can a keyboard produce a ghost key? When three related keys are pressed at the same time, the keyboard controller may send two different output codes which are called a ghost key. When a ghost key appears, the controller provides a wrong judgment of information because the controller is unable to find the correct key. In order to prevent the wrong signal from being sent, some keyboard controllers may neglect the pressed 3rd key, the keyboard controller will only output the signal sent by the first 2 keys. When a notebook receives the wrong code, it produces a ghost key which is caused as a result of the key location contradiction [7]-[10].

Fig. 1 (a) and (b) show parts of keyboard matrix, with 4 keys Q, W, A, S in 2 columns and 2 rows. A real keyboard matrix will not be so simple, it will be more complicated. In Fig.1 (b), when the Q key is pressed a scanning begins from Column 1. As Row 1 has led into a coherent circuit but Row 2 has not, the resulting inference produces a Column 1/Row 1 pigeonhole. If the keyboard controller begins the scanning from Column 2 again, Row 1 and Row 2 will have not been formed as a result of the operation. The controller conclusion obtains Column 1/Row 1 whereby the Q key is pressed.

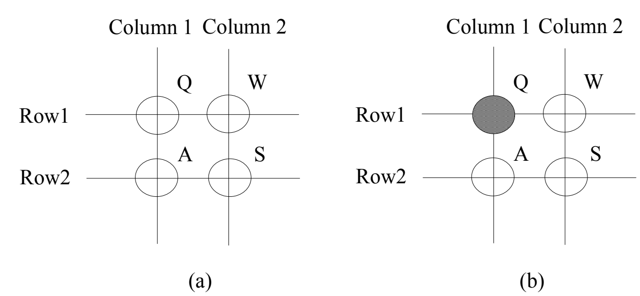

Fig. 1. (a), (b) is a part of the keyboard matrix, with 4 keys Q, W, A, S in 2 columns and in 2 rows.

In Fig.2 (c), when both the Q, and S keys are pressed, the controller begins scanning from Column 1. Row 1 has led into a coherent circuit but Row 2 has not. Therefore the inference produces a Column 1/Row 1 pigeonhole. The controller then begins scanning from Column 2. Row1 has not formed a thoroughfare. Row 2 has led the coherent circuit, so the inference produces Column 2/Row 2 which is then pressed. The conclusion is Column 1/Row 1, Column 2/Row 2, which is, namely, when the Q key, and S key are pressed.

In Fig.2 (d), When the Q, S, and A keys are pressed, the controller begins scanning from Column 1. Row 1 leads into a coherent circuit, Row 2 also leads into a coherent circuit, so the resulting inference produces Column 1/Row 1. The controller begins scanning from Column 2, Then Row 2 leads into a coherent circuit. Row 1 also leads into a put through circuit through Column 1 at this moment, so the resulting inference appears as Column 2/Row 2. The controller scanning results obtain Column 1/Row 1, Column 1/Row 2, and Column 2/Row 2, which are, namely, the Q key, the A key, and the S key. The A key was not pushed but there are no electronic contact. This key is, the ghost key [11] [12].

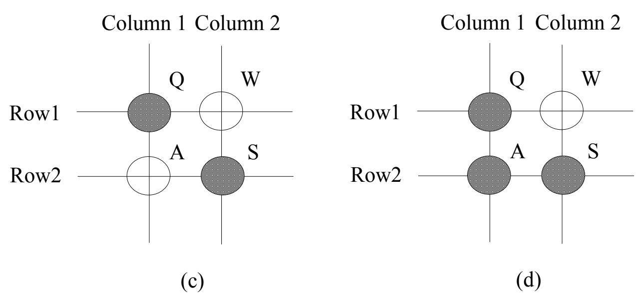

Fig. 2. (c), (d) is a part of the keyboard matrix, with the 4 keys Q, W, A, S in 2 columns and 2 rows.

III. Hardware and Software Design

This section describes the proposed design of an anti-ghost key in a keyboard of a notebook and this new keyboard extends the length of the ghost key number by a wide margin.

A. *Hardware design*

This new design adds on a second low-power embedded controller (EC) to the notebook as shown in Fig. 3. The main EC and the second EC which are connected to the south bridge chip control the peripheral devices. The main EC controls the system power, battery charge/discharge, thermal temperature, fan speed, etc. The second EC controls the keyboard and touchpad.

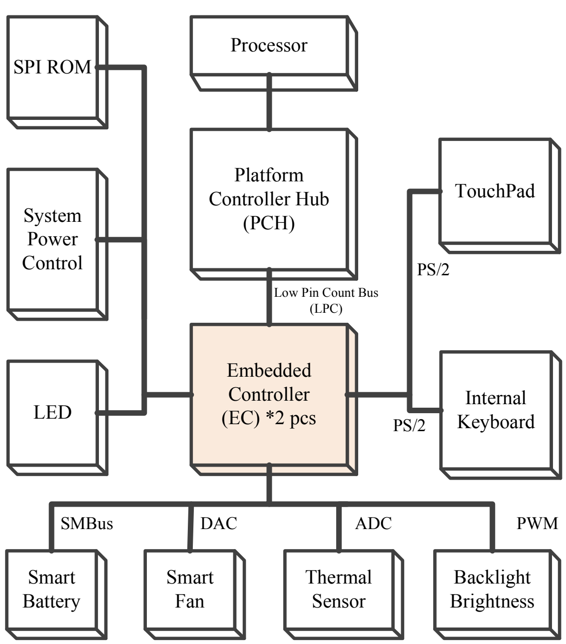

Fig. 3. System architecture of the notebook and the ECs

Fig. 4 shows the system architecture of the main EC and second EC architecture. This design use a SMBus interface with one interrupt pin to communicate between two ECs. The second EC consists of mainly scanning 8*44 pins of an internal keyboard and a touchpad device. The previous design of a notebook has one EC only, because the scanning matrix of the keyboard matrix is increased to 8*44 pins. This design adds on a second low-power EC to control the keyboard, which can lighten the working load of the main EC.

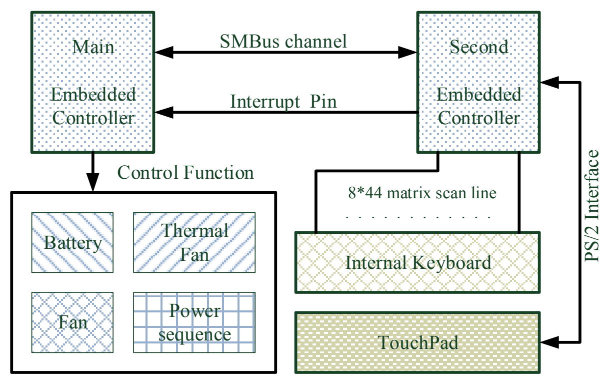

Fig. 4. System architecture of the main EC and second EC

B. *Software design*

As shown in Fig. 5, the main EC needs to be able distinguish the second EC in the IO Port structure. As this design is to let the second EC control both the keyboard and the Touchpad, the IO Port is set up as 0x60/0x64. Since the main EC controls other peripheral functions, so the IO Port is set up as 0x62/0x66. Hence communication between the two ECs will not interfere with each other.

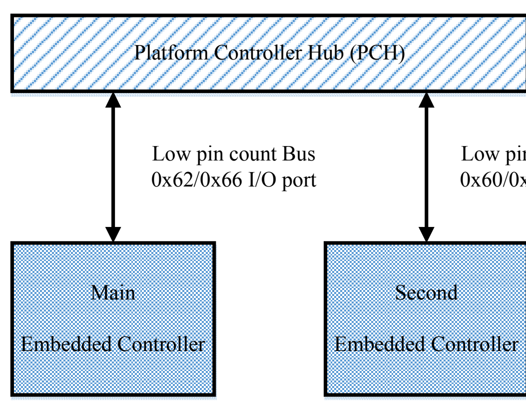

Fig. 5. Low pin count Bus architecture of the main EC and second EC

The flowchart of the keyboard scanning program is shown in Fig. 6. By using the interrupt mechanism, the keyboard is able to start scanning. The total scan is 44 lines. When any key is pressed, the EC, by scanning the line detects which specific column and row of the keyboard matrix. After confirming the column and row of a key, then the second EC checks the ghost key rule, finds the corresponding ASCII code and sends out the ASCII code of this key to the operating system.

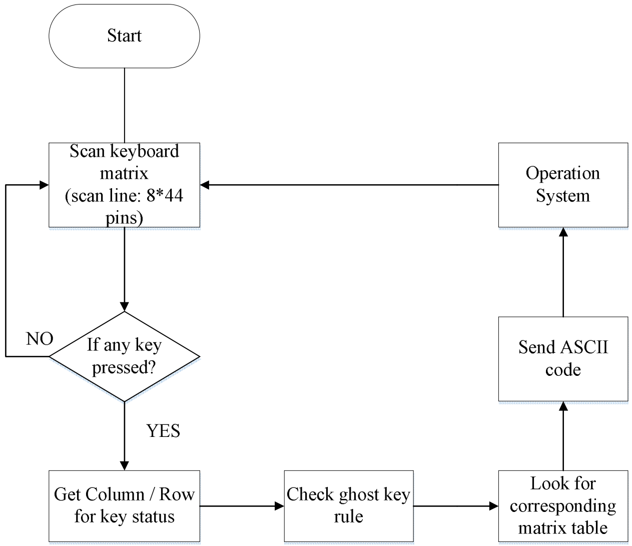

Fig. 6. Flowchart for keyboard scan rule

Fig. 7 shows the full size keyboard (102 keys) sketch map design.

Fig. 7. Layout of a full size keyboard

Fig. 8 is the keyboard matrix design. This design expands the keyboard matrix from 8*16 pins to 8*44 pins. Among this matrix the keyboard scan input (KSI) is 0-7; and the keyboard scan output (KSO) is 0-43. 102 keys make up a subsection at the keyboard matrix. This design expand some keys in a KSO of independence, such as 0-9, a-z, etc.

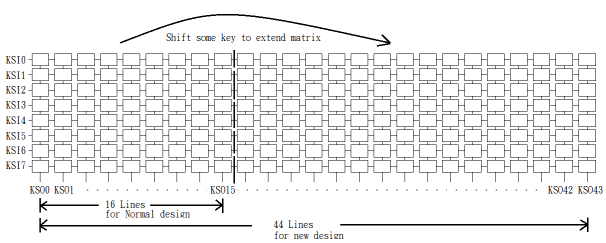

Fig. 8. An expansion of the scan line from 8*16 pins to 8*44 pins for the keyboard matrix

IV. COMPARISON AND IMPLEMENTATION RESULTS

Fig. 9 shows the main EC and the second EC in the position picture on the main board.

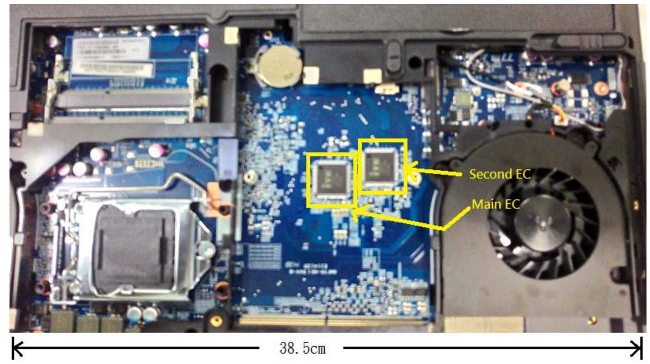

Fig. 9. The main EC and the second EC in the position picture on the main board

Fig. 10 shows the picture of a full size keyboard with an 8*44 scan line.

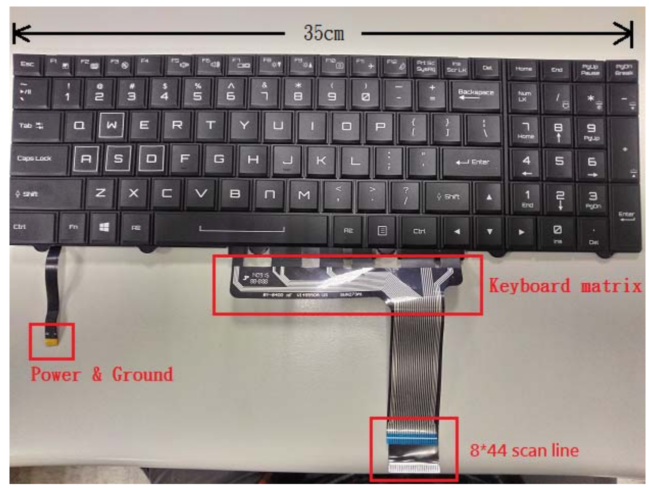

Fig. 10. Physical picture of a full size keyboard with an 8*44 scan line

Table I shows the compound key for an anti-ghost key of this design. This design provides the compound key most often used in comparison to the original design. If a user presses the compound key at the same time, this design will not have any ghost key problem. This design can support the simultaneous pressing of 43 keys and will not produce any ghost key.

TABLE I
A COMPOUND KEY FOR AN ANTI-GHOST KEY

Keys pressed	Ghost key
A-Z + “arrow key up”, “arrow key down”	No
A-Z + “arrow key left”, “arrow key right”	No
0~9 + Any key	No
A-Z + 0-9 + Tab + Alt + Ctrl + Shift + Win + Fn + Space + Enter	No

Table II shows this design in comparison with other designs. There are many keyboards on the market at present. All have designs that include an anti-ghost key design. In Table II, we have enumerated three designs which we compare with this design. By using Table II, this design can allow the user to press 43 different keys at the same time without any key conflict. Moreover the length of the anti-ghost key will be longer than that of other designs.

TABLE II
COMPARISON WITH OTHER DESIGNS

Keyboard designs	Anti-Ghost key
Our design	43 keys
Design 1	40 keys
Design 2	24 keys
Design 3	15 keys

Table III shows the specification of the second EC which serves as the keyboard controller. When the keyboard is being used, the power consumption of the second EC is 59.6mW. When the keyboard is not being used, as the EC will enter the idle mode, the power consumption will be reduced to 27.7mW. The total time that the keyboard controller uses to scan once is 1.2 ms, and it scans all lines from 1 to 44. If a user presses a key, the response time of the second EC is 15us.

Other designs use 8*16 keyboard matrixes. The total scan time of the keyboard controller has a small difference. The time used to generally press one key normally is probably 40 ms to 45 ms, so during the keyboard operation, as the keyboard controller scan time only ranges between 0.5 ms and 1.2 ms, a user cannot feel the difference.

TABLE III
SPECIFICATION OF THE SECOND EMBEDDED CONTROLLER

	Main Embedded Controller for 8*16 keyboard matrix	Second Embedded Controller for 8*44 keyboard matrix
Power consumption of operation state	59.6mW	59.6mW
Power consumption of idle state	27.7mW	27.7mW
Total time of the keyboard scan	0.5ms	1.2ms
Response time of a pressed key	15us	15us

V. CONCLUSION

Because the development of the game industry is continually advancing by leaps and bounds, many users may press multiple keys which can cause a ghost key problem. However, with an appropriate design, any notebook keyboard ghost key problem can be avoided.

This design adds on a second low-power EC as a keyboard controller, and expands the keyboard matrix from 8*16 pins to 8*44 pins. This design will not produce a ghost key problem when 43 keys are pressed at the same time. This design can meet a high percentage of market satisfaction when used in connection with game software, and will not produce any ghost key problem. Experimentation shows that this new design has a better performance than the other designs on the market at present.

REFERENCES

[1] Ying-Wen Bai and Ciun-Hung Cheng, "Using Fuzzy Logic to Reduce Power Consumption for Notebooks," *IEEE The 13th International Symposium on Consumer Electronics*, pp. 488-492, May 2009.

[2] Ying-Wen Bai and Ciun-Hung Cheng, "Using an Embedded Controller with Fuzzy Logic to Reduce Power Consumption of Mobile Computers," *The 36th Annual Conference of the IEEE Industrial Electronics Society (IECON-2010),* Phoenix, Arizona, USA, pp. 230-235, Nov. 7-10, 2010.

[3] Chun-Hung Cheng and Ying-Wen Bai, "A Peak-Shift Control Method for Charging and Discharging of the Battery in an Ultrabook," *IEEE International Conference on Consumer Electronics*, pp. 296-297, Jan. 2014.

[4] Jong Yeon Park, Dong-Guk Han, Okyeon Yi, and Dooho Choi, "Ghost key patterns with Equidistant Chosen Message attack on RSA-CRT," *IEEE International Carnahan Conference on Security Technology*, pp. 1-5, Oct. 2011.

[5] Shiqian Wu, Shoulie Xie, S. Rahardja, and Zhengguo Li, "A robust and fast anti-ghosting algorithm for high dynamic range imaging," *IEEE International Conference on Image Processing*, pp.397-400, Sept. 2010.

[6] Taekyoung Kwon, Sarang Na, and Sang-Ho Oark, "Drag-and-type: a new method for typing with virtual keyboards on small touchscreens," *IEEE Transactions on Consumer Electronics*, pp. 99-106, vol. 60, no. 1, Feb. 2014.

[7] D. Rozado, "Mouse and Keyboard Cursor Warping to Accelerate and Reduce the Effort of Routine HCI Input Tasks," *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 5, pp.487-493, Oct. 2013.

[8] M. Rofouei, M. Potkonjak, and M. Sarrafzadeh, "Energy Efficent Collaborative Sensing-Based Design: Soft Keyboard Case Study," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 1, pp.115-124, Jan. 2014.

[9] H. A. Habib and M. Mufti, "Real time mono vision gesture based virtual keyboard system," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 4, pp.1261-1266, Nov. 2006.

[10] A. B. Usakli and S. Gurkan, "Design of a Novel Efficient Human-Computer Interface: An Electrooculagram Based Virtual Keyboard," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 8, pp. 2099-2108, Aug. 2010.

[11] Bhattacharya S., Samanta D. and Basu A., "Performance Models for Automatic Evaluation of Virtual Scanning Keyboard," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 5, pp. 510-519, Oct. 2008.

[12] Ying-Wen Bai and Ming-Yuan Yang, "An improved design of a keyboard powered by solar cells and large capacitor," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1355-1359, Apr. 2008.