Journal of Computers Vol. 27, No. 4, 2016, pp. 131-148
doi:10.3966/199115592016122704011

Simulator Design of Small-Sized Chip Automatic Programming Equipment

Ching-Chang Wong[1*], Yu-Ting Yang[1], An-Song Wang[1],
Meng-Cheng Lau[2], and Hsuan-Ming Feng[3]

[1] Department of Electrical Engineering, Tamkang University, Taiwan
wong@ee.tku.edu.tw

[2] Department of Computer Science, University of Manitoba, Canada

[3] Department of Computer Science and Information Engineering, National Quenoy University, Taiwan

Received 05 November 2015; Revised 23 January 2016; Accepted 15 February 2016

Abstract. A small-sized chip automatic programming equipment simulator is presented and experimented in this paper. The chip automatic programming equipment is combined with one pick-and-place system, two trays, and several programmers on a platform. An equipment simulator is designed to simulate the general specification and research the production capacity. Two cases with four programmers and eight programmers are presented in this paper, and each one with three configuration types. A simply behavior strategy is designed to make decision of behavior and test configurations for the simulator. From the experimental results, the proposed simulator can be used to simulate the production capacity of the entity equipment so that the proposed simulator can be used to assist the inference of the appropriate relationship between the programming time and the programmer number.

Keywords: automatic equipment, chip programming equipment, equipment simulator

1 Introduction

The chip automatic programming equipment is designed for programmable chips so that some program can be stored in them automatically. Improve the system's efficiency becomes one behavior control problem of the chip programming equipment. Many similar issues focus on manufacturing methods of chip [1-4], but no one discusses on manufacturing methods of equipment. About the equipment, the greatest impact of production capacity is the time spending of picking and placing. It means to pick up a chip from one place and place it into another place. Therefore, the production capacity could be raised up if the time spending of picking and placing can be reduced.

The common chip automatic programming equipment is divided into: (1) Large-sized equipment and (2) Small-sized equipment. They are described as follows:

(1) Large-sized equipment

Chip automatic programming equipment, which the weight is about 1000 kg and the height is above 100 cm, can be classified to large-sized equipment. In 1969, the first large-sized programming equipment was designed by Data IO. Since then, many companies continued to follow up the design and manufacture. The most common designing company was changed by the Semiconductor Manufacturing Company, like Sun-s and MINATO in Japan. Others, like Leaptronix is the first designing company of chip automatic programming equipment in Taiwan. A variety equipment was introduced by Dediprog from 2005. Hilosy is the first manufacturing company of universal type programmer. By the rising demand of consumer electronics and the rising cost of the manpower, it makes the demand of the large-sized automatic programming equipment into peak. Although the production capacities of large-size equipment are

* Corresponding Author

3600 units per hour (UPH), the size and price of the equipment are not easily accepted for small companies. Consequently, the research of small-sized equipment is started.

(2) Small-sized equipment

Chip automatic programming equipment with below 100 cm height can be classified to small-sized equipment. The biggest feature is the low price so that small companies can easier to buy. Another feature is small volume so that the same area can be placed more equipment. Data IO [5] is the first company to design and sell the small-sized automatic programming equipment in the world. The features of Data IO are the modular approach and special programmer design. Suns owns the smallest size of equipment in the world. Leaptronix is main designed to compatible different programmers. Although the small-sized chip automatic programming equipment is designed for small production. However, the current production capacity is only 550 to 600 UPH.

The small-sized chip automatic programming equipment has one pick-and-place system, two trays, and programmers on a platform. Example of mechanism architecture like Fig. 1, it is the equipment example from Leaptronix. Un-programmed chip is placed in left side (In Tray), programmed chip is placed in middle (Out Tray), and programmers are placed in right side (Programmer). The 3-axis pick-and-place system is crossed over the In Tray and Programmers. The moving range can reach all parts. All parts are connected by a platform. The mechanism architecture is consisted by four parts: (1) Platform, (2) Programmer, (3) Tray, and (4) Pick-and-place system. They are described as follows:

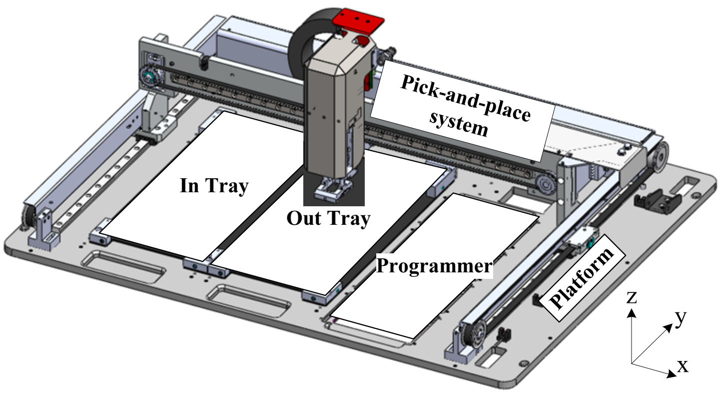


Fig. 1. Mechanism architecture of small-sized chip automatic programming equipment

(1) Platform

Platform is the basic working plane of equipment. All parts are combined on the table. The table has been positioned horizontally to avoid the chip falling.

(2) Programmer

Programmer is set several sockets on the top. The socket is used to stabilize the chip on the programmer. The purpose is maintaining a good contact between the chip and the programmer. Ensure the programming process does not be affected. Accordance with different size of chip, the number of sockets on one programmer is not always the same. In this paper, there are four sockets on one programmer, and there are 16 sockets in the equipment.

(3) Tray

Tray is a chip loading plant which has international specifications. The tray is used to place the packaged chip (see Fig. 2). There are many pockets for placing chips on the tray. Chips are kept in pockets independently without interference. Accordance with different size of chip, the number of pockets on one tray is also different. In this paper, the TQFP-CS-007AG of Topline JEDEC (Joint Electron Device Engineering Council) is used to be an example for simulating. There are 60 pockets on the tray TQFP-CS-007AG. The detail specification is in.

(4) Pick-and-place system

Pick-and-place system is a xyz 3-axis moving system which is made by motors and gears (see Fig. 3). Each axis is driven by one motor and connected two gears by one industrial belt. The moving object is installed on the industrial belt, so that the motor can move the object linearly. Y-axis which is front and rear axial is designed on the platform. The gantry of Y-axis is designed to carry the heavy pick-and-place

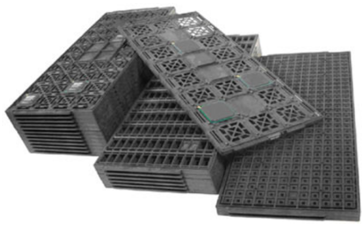


Fig. 2. Topline JEDEC tray [5]

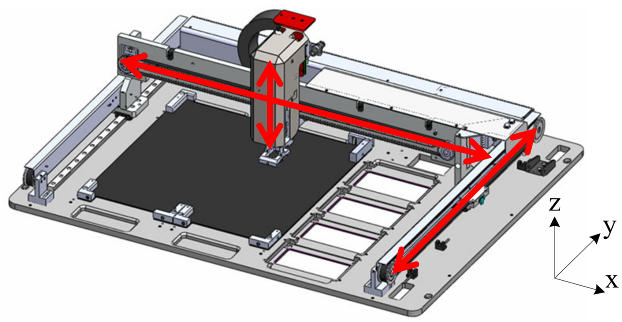


Fig. 3. Pick-and-place system

system. X-axis which is left and right axial is designed on the Y-axis. Z-axis which is up and down axial is the endpoint of the system and designed on the X-axis.

The purpose of this paper is to design a simulator so that the production capacity of small-sized automatic programming equipment can be improved. In accordance to the user and market, there are four restrictions of specification: (1) weight and volume restriction, (2) IC placing restriction, (3) programmer restriction, and (4) programming failure restriction. They are described as follows:

(1) Weight and volume restriction

The weight and volume restriction is the biggest difference between large-sized and small-sized equipments. In this paper, the equipment is limited in one pick-and-place system. Only one chip is taken in one action.

(2) IC placing restriction

Theoretically, the best strategy order is to pick up a chip from the nearest programmer and place it into the most suitable location. ICs in tray are messy in the processing, and neatly filled and allows user to remove ultimately. But in fact, the processing may end temporary and IC may programming fail. Therefore, the ICs need to be picked and placed in the order of the tray.

(3) Programmer restriction

Programmer is set several sockets on the top. The socket is used to stabilize the chip on the programmer. In this paper, there are four sockets on one programmer. These four sockets belong to one programmer. The ICs are programmed together after four sockets are placed.

(4) Programming failure restriction

When the IC is programmed failure, the programmer will re-program all ICs on programmer.

The purpose of this paper is to discuss how to increase production capacity of small-sized automatic programming equipment in these four restrictions of specification. The rest of this paper is organized as follows. In Section 2, a simulator with some behavior strategies and codes is described in detail. In Section 3, some results of the simulator are presented to illustrate the efficiency of the proposed method. Some design approaches for an appropriate equipment configuration are discussed. Finally, some conclusions are made in Section 4.

2 Simulator of Small-sized Chip Automatic Programming Equipment

2.1 Behavior strategy and behavior codes

Behavior strategy is the decision of step order of system. In this paper, a decision-making of output priority is proposed to determine its sequence. Each programmer returns the current state to behavior strategy independently. Programmers with the same state are stacked by First-In-First-Out mode. The decision-making process is shown in Fig. 4, where $a_{act}^{\ell^*}$ is the behavior of pick-and-place system, L_{com} is the number list of completed programmers, L_{non} is the number list of non-working programmers, ℓ^* is the action number of programmer. First, it is determined whether there has been accomplished programmer. The second is judged the programmer which doesn't programming. If all programmers are working, then the system will wait 1s. These actions are described as follows:

$a_{act}^{\ell^*} = 0$: Wait 1s.

$a_{act}^{\ell^*} = 1$: Pick the chip from the In Tray and place it into the programmer.

$a_{act}^{\ell^*} = 2$: Pick the chip from the programmer and place it into the Out Tray.

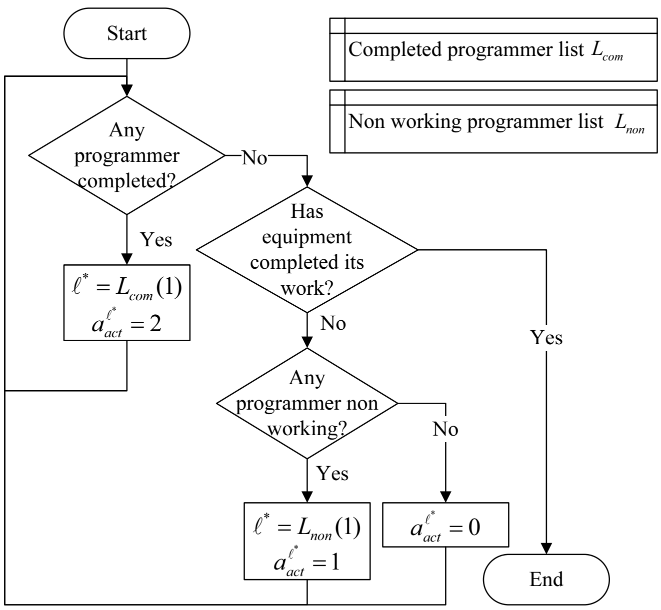


Fig. 4. Flowchart of behavior strategy

2.2 Equipment simulator

The flowchart of equipment simulator is shown in Fig. 5. Besides behavioral strategies, there are total of six steps: (1) Data initialization, (2) Pick-and-place system update, (3) Programmer update, (4) Chip update, (5) Tray update, and (6) Ending check. They are described as follows:

(1) Data initialization

Initialing data is included the information of platform, pick-and-place system, programmer, chip and tray. In this paper, there are one platform, one pick-and-place system, four programmers, 60 chips, and two trays. The size perimeters are shown in Fig. 6. Each important location on platform is assigned a coordinate, such as pockets of tray and sockets of programmer. The size perimeters on platform are show in Fig. 6. All coordinates are defined initially.

The specification of sockets on programmer is shown in Fig. 7. The coordinate of each socket is described by

$$x_\ell^{s_\ell} = d_b^{left} + 2d_x^{tr} + d_b^{x_1} + d_b^{x_2} + d_p^{M} + s_\ell d_p^{M_2} \tag{1}$$

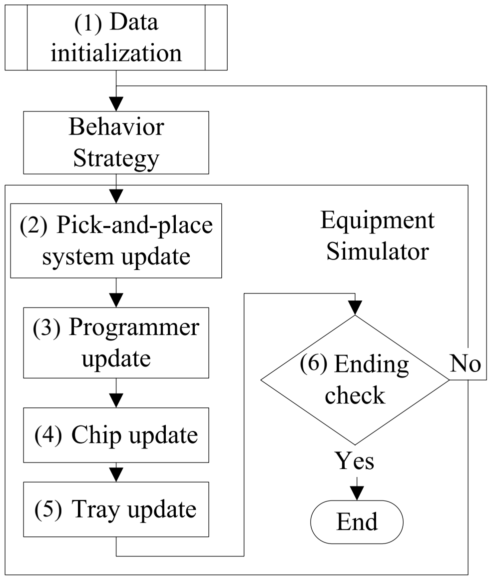


Fig. 5. Fowchart of equipment simulator

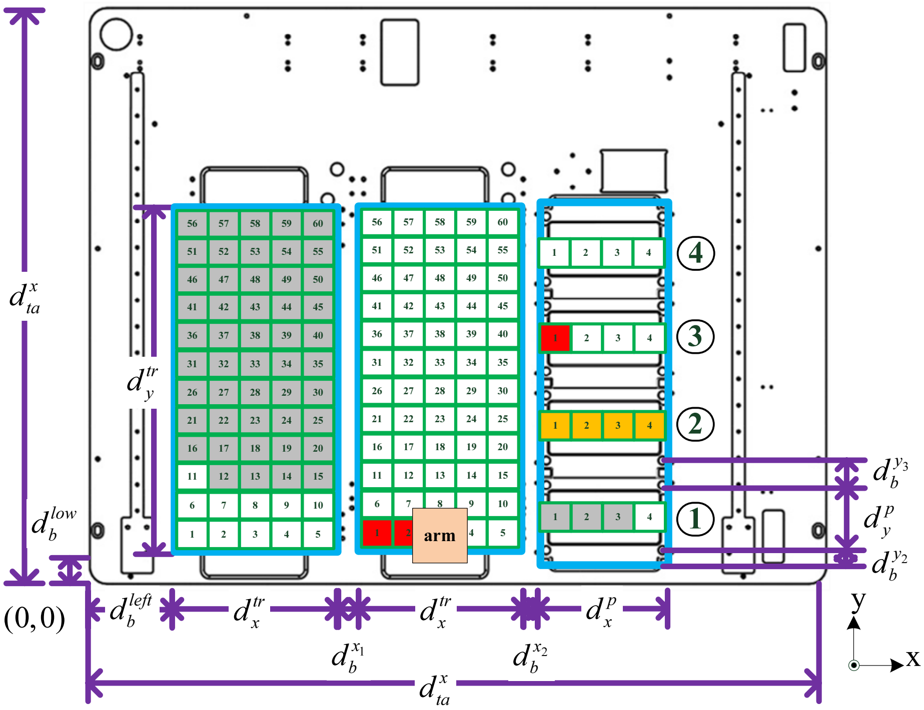


Fig. 6. The size perimeters on platform

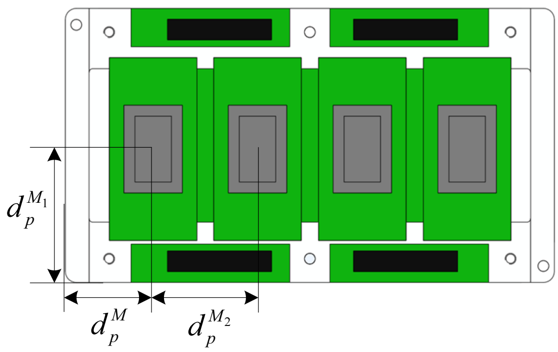


Fig. 7. The specification of sockets

and

$$y_{\ell}^{s_\ell} = d_b^{low} + d_b^{y_2} + d_p^{M_1} + (\ell - 1)(d_b^{y_3} + d_y^{p})\,\mathrm{c} \quad \textbf{(2)}$$

where $(x_\ell^{s_\ell}, y_\ell^{s_\ell})$ are coordinates of every socket, and $\ell \in \{1,2,\ldots,n_p\}$. The coordinates each pockets of two trays are described by

$$x_c^{in} = d_b^{left} + \frac{d_{tr}^{M}}{2} + (c-1)d_{tr}^{M_3} \quad \textbf{(3)}$$

$$y_c^{in} = d_b^{low} + \frac{d_{tr}^{M_1}}{2} + (c-1)d_{tr}^{M_2} \quad \textbf{(4)}$$

$$x_c^{out} = d_b^{left} + \frac{d_{tr}^{M}}{2} + (c-1)d_{tr}^{M_3} + d_{tr}^{x} + d_b^{x_1} \quad \textbf{(5)}$$

and

$$y_c^{out} = d_b^{low} + \frac{d_{tr}^{M_1}}{2} + (c-1)d_{tr}^{M_2} \quad \textbf{(6)}$$

where $(x_c^{in}, y_c^{in}), (x_c^{out}, y_c^{out})$ are coordinates of In Tray and Out Tray pocket, $c \in \{1,2,\ldots,n\}$ is the serial number of each chip. n_{ic} is number of IC on tray, d_{tr}^{M} is the horizontal distance between left side of tray to center of next pocket. $d_{tr}^{M_1}$ is the vertical distance between left side of tray to center of next pocket, $d_{tr}^{M_2}$ is vertical distance between pocket to pocket, and $d_{tr}^{M_3}$ is horizontal distance between pocket to pocket. The specification of tray is shown in Fig. 8.

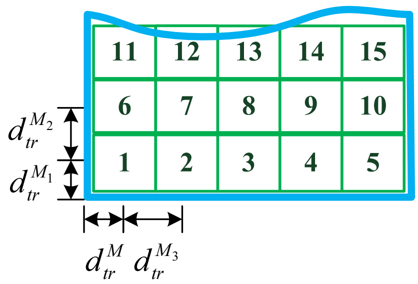


Fig. 8. The specification of tray

(2) Pick-and-place system update

Updating the pick-and-place system is included to calculate the distance of moving and the spending time of execution. The order of behavior strategy is included picking and placing chip, so that the distance of moving is from the current location to the picking location and then to the placing location. The distance of moving is described by

$$d^{\ell^*} = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2} + \sqrt{(x_g - x_b)^2 + (y_g - y_b)^2} \quad \textbf{(7)}$$

where (x_a, y_a) is the current coordinate of the endpoint of pick-and-place system, (x_b, y_b) is the coordinate of IC which will be picked up, (x_g, y_g) is the coordinate of IC which will be placed into. The spending time of execution is described by

$$t_o = \begin{cases} 1, & \text{if } a_{act}^{\ell^*} = 0 \\ \dfrac{d^{\ell^*}}{v_a}, & \text{if } a_{act}^{\ell^*} \neq 0 \end{cases} \tag{8}$$

where v_a is the average speed of picket-and-place system. Then the current coordinate is updated by

$$x_a = x_g\,,\ y_a = y_g \tag{9}$$

(3) Programmer update

Updating programmer is included the state of sockets, the remaining time of programming and the state of programmer. The state of sockets which are updated by the order of behavior strategy are described by

$$n_{ps}^{\ell^*} = \begin{cases} n_{ps}^{\ell^*} + 1, & \text{if } a_{act}^{\ell^*} = 1 \\ n_{ps}^{\ell^*} - 1, & \text{if } a_{act}^{\ell^*} = 2 \end{cases} \tag{10}$$

where ℓ^* is the number of sockets. The remaining time of programming and the state of programmers are described by

$$t_r^{\ell} = \begin{cases} t_r^{\ell} - t_o, & \text{if } t_r^{\ell} > 0 \\ t_r^{\ell}, & \text{if } (n_{ps}^{\ell} = n_s \text{ and } q_p^{\ell} = 0) \text{ or } (t_r^{\ell} = 0 \text{ and } q_{out} > n_{ic}),\ \ell \in \{1, 2, ..., n_p\} \\ 0, & \text{if } t_r^{\ell} \leq 0 \end{cases} \tag{11}$$

and

$$q_p^{\ell} = \begin{cases} 0, & \text{if } t_r^{\ell} = 0 \text{ and } n_{ps}^{\ell} = 0 \\ 1, & \text{if } t_r^{\ell} \leq 0 \end{cases} \tag{12}$$

where q_p^{ℓ} is the index of finishing programming (0 is unfinished, and 1 is finished).

(4) Chip update

Updating chip is included the current coordinate. The current coordinates of chip which are updated by the order of strategy and state of programmer are described by

$$x_c^{ic} = \begin{cases} x_g, & \text{if } a_{act}^{\ell^*} \neq 0 \\ x_c^{ic}, & \text{otherwise} \end{cases},\ c \in \{1, 2, ..., n_{ic}\} \tag{13}$$

and

$$y_c^{ic} = \begin{cases} y_g, & \text{if } a_{act}^{\ell^*} \neq 0 \\ y_c^{ic}, & \text{otherwise} \end{cases},\ c \in \{1, 2, ..., n_{ic}\} \tag{14}$$

where n_{ic} is the total number of tray, (x_g, y_g) is the current coordinate of chip.

(5) Tray update

Updating tray is included the index number of In Tray and Out Tray. The index number which is a serial number of pocket at tray is specified to next working unit. They are described by

$$q_{in} = \begin{cases} q_{in} + 1, & \text{if } a_{act}^{\ell^*} = 1 \\ q_{in}, & \text{otherwise} \end{cases} \tag{15}$$

and

$$q_{out} = \begin{cases} q_{out} + 1, & \text{if } a_{act}^{\ell^*} = 2 \\ q_{out}, & \text{otherwise} \end{cases} \quad \textbf{(16)}$$

where q_{in} and q_{out} are the index number of In Tray and Out Tray.

(6) Ending check

The ending is checked by index number q_{out}. If q_{out} is bigger than total number of chips, then the work has been completed. The production capacity UPH is calculated at the end of work. It is described by

$$UPH = \frac{n_f}{t_{nf}} \cdot 3600 \quad \textbf{(17)}$$

where n_f is the total number of chip. The UPH which is an abbreviation of unit per hour is a measurement unit of production capacity. In this paper, time unit is defined by seconds, so that production capacity is multiplied (60x60=3600) to replace the unit for hour.

2.3 Simulator validation

The validation of simulator is based on the production capacity. The general specification of small-sized chip automatic programming equipment which is sold in market is a validation criterion. The production capacity of Data IO is 660 PPH (Produce Per Hour) [5] and Leaptronix is 550 UPH. The zero programming time is used by both two companies. One simulation situation of the proposed simulator for the small-sized equipment is shown in Fig. 9 and a simulation result of simulator validation is shown in Fig. 10 with zero programming time. The horizontal axis is the total execution time. The vertical axis in Fig. 10(a) is the execution serial number of programmer. The vertical axis in Fig. 10(b) is the value of behavior strategy. The production capacity is 558.9591 UPH in this simulation which is conformed to the general specification. Therefore, the proposed simulator can be used to simulate the production capacity of the entity equipment.

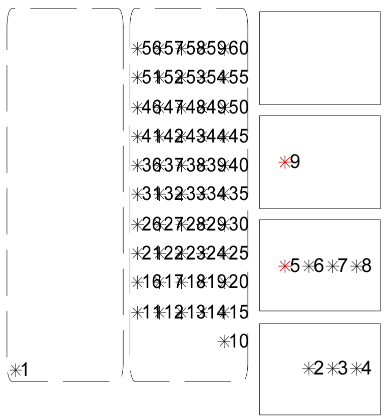
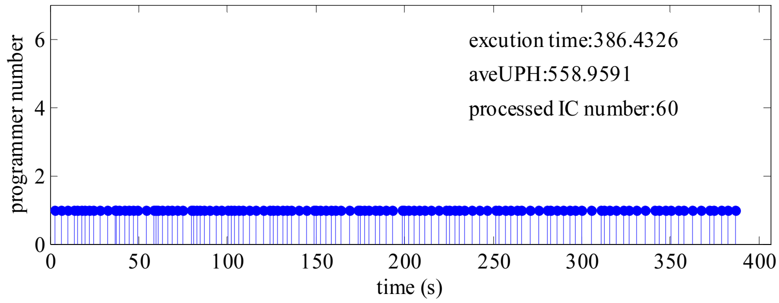


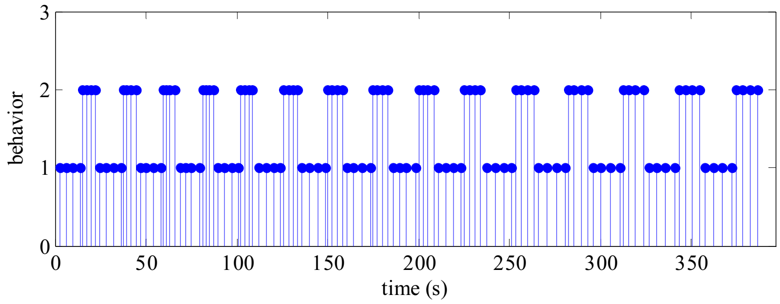
Fig. 9. One simulation situation of the proposed simulator for the small-sized equipment

3 Experiment and Discussion

This paper proposes a design approach of small-sized chip automatic programming equipment simulator. The following experiments are proposed to compare the production capacity with the different number of programmer, different programming time, and different configuration types of equipment. There are total 38 different programming times which with each 10 second interval under 10 second to 360 second. The major environment is designed by two types: (a) four programmers and (b) eight programmers. They are described as follows:



(a)



(b)

Fig. 10. Simulation results of simulator validation with zero programming time

3.1 Four programmers experiment

In this study, there are total four programmers designed in equipment. Each programmer is having same programming time ($t_p^1 = t_p^2 = t_p^3 = t_p^4$), and four sockets ($n_s = 4$). There are three configuration types of equipment: (1) Plan A, (2) Plan B, and (3) Plan C. They are described as follows:

(1) Plan A

The configuration sequence of Plan A is In Tray, Out Tray, and Programmers. As shown in Fig. 11, the In Tray is in the left side; the Out Tray is in the middle; and Programmers are in the right side. This is a produced example form Leaptronix. The experimental results with 36 programming times are tabulated in Table 1 and Fig. 12. Although the UPH is decreased in the longer programming time, but the magnitude is gradually reduced. This is because the time of programming longer than the spending time of execution of pick-and-place system, then the production capacity is impacted less by the time of moving.

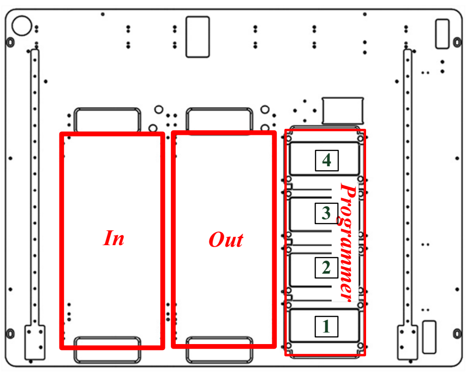


Fig. 11. Configuration of Plan A

Table 1. Experiment results of Plan A with four programmers

t_p^{ℓ}	10s	20s	30s	40s	50s	60s	70s	80s	90s
UPH	569.95	550.59	562.76	537.88	537.88	515.27	494.52	455.28	413.85
t_p^{ℓ}	100s	110s	120s	130s	140s	150s	160s	170s	180s
UPH	384.39	358.85	336.48	316.75	299.20	283.49	269.35	256.55	244.92
t_p^{ℓ}	190s	200s	210s	220s	230s	240s	250s	260s	270s
UPH	234.29	224.55	215.58	207.31	199.64	192.53	185.90	179.71	173.92
t_p^{ℓ}	280s	290s	300s	310s	320s	330s	340s	350s	360s
UPH	168.50	163.40	158.60	154.07	149.80	145.76	141.92	138.29	134.84

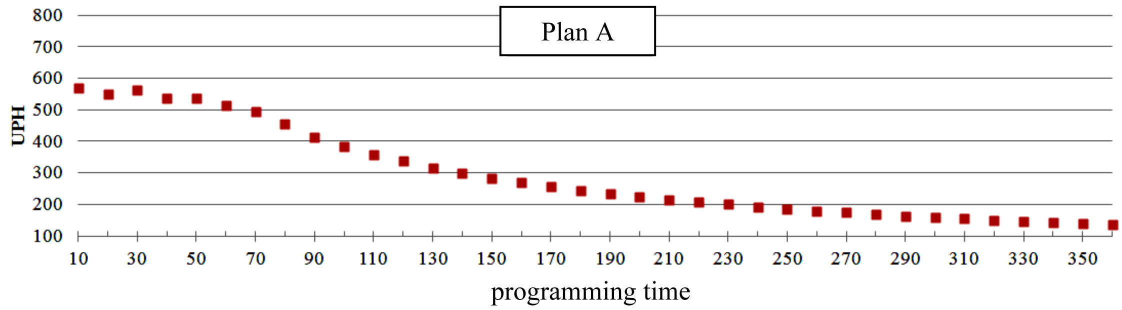


Fig. 12. Experiment line chart of Plan A with four programmers

(2) Plan B

The configuration sequence of Plan B is Out Tray, In Tray, and Programmers. As shown in Fig. 13, Out Tray is in the left side; In Tray is in the middle; and Programmers are in the right side. This arrangement is designed to compare the production capacity between In/Out Tray and Programmers with different distance. The experimental results with 36 programming times are tabulated in Table 2 and Fig. 14.

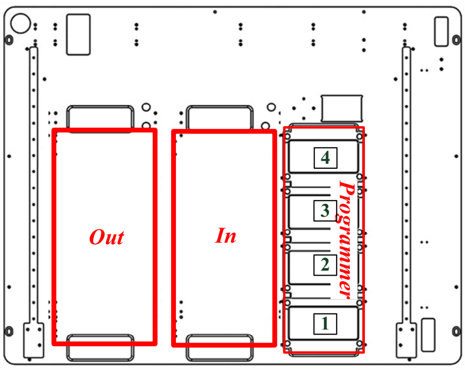


Fig. 13. Configuration of Plan B

Table 2. Experiment results of Plan B with four programmers

t_p^{ℓ}	10s	20s	30s	40s	50s	60s	70s	80s	90s
UPH	558.06	537.18	563.99	501.32	458.73	422.81	392.11	365.57	342.39
t_p^{ℓ}	100s	110s	120s	130s	140s	150s	160s	170s	180s
UPH	321.97	303.86	287.67	273.12	259.97	248.03	237.14	227.16	217.99
t_p^{ℓ}	190s	200s	210s	220s	230s	240s	250s	260s	270s
UPH	209.53	201.71	194.44	187.69	181.38	175.49	169.96	164.78	159.90
t_p^{ℓ}	280s	290s	300s	310s	320s	330s	340s	350s	360s
UPH	155.30	150.96	146.85	142.96	139.28	135.78	132.45	129.27	126.25

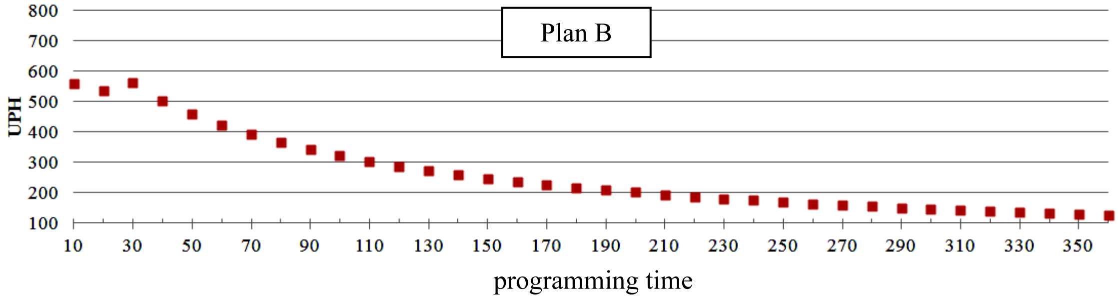


Fig. 14. Experiment line chart of Plan B with four programmers

(3) Plan C

The configuration sequence of Plan C is In Tray, Programmers, and Out Tray. As shown in Fig. 15, In Tray is in the left side; Programmers are in the middle; and Out Tray is in the right side. This is a produced example form Data IO [5]. The experimental results with 36 programming times are tabulated in Table 3 and Fig. 16.

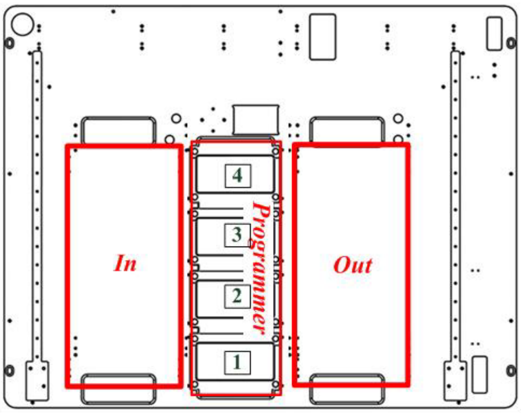


Fig. 15. Configuration of Plan C

Table 3. Experiment results of Plan C with four programmers

t_p^ℓ	10s	20s	30s	40s	50s	60s	70s	80s	90s
UPH	734.78	716.54	700.56	661.07	588.97	531.05	483.50	443.76	410.07
t_p^ℓ	100s	110s	120s	130s	140s	150s	160s	170s	180s
UPH	381.12	356.00	333.98	314.53	297.22	281.71	267.74	255.09	243.59
t_p^ℓ	190s	200s	210s	220s	230s	240s	250s	260s	270s
UPH	233.07	223.43	214.55	206.35	198.76	191.70	185.13	178.99	173.25
t_p^ℓ	280s	290s	300s	310s	320s	330s	340s	350s	360s
UPH	167.87	162.80	158.04	153.55	149.30	145.28	141.48	137.87	134.43

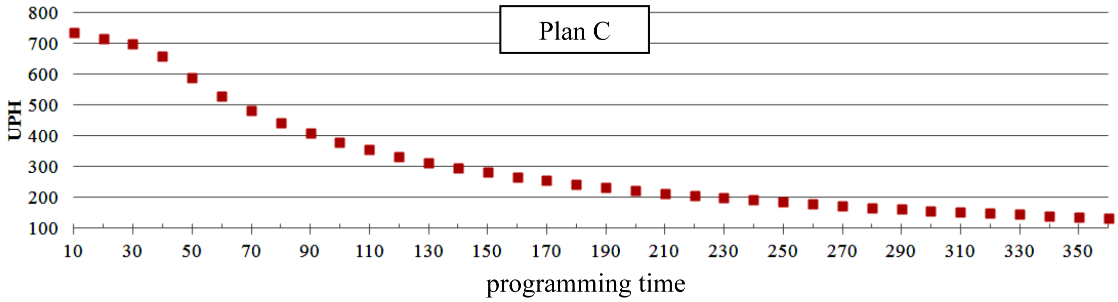


Fig. 16. Experiment line chart of Plan C with four programmers

Fig. 17 and Fig. 18 are comparison charts of these three experiments. Three methods are quite different in the 60 seconds programming time or less. Based on these data, the following information is observed: (1) Plan C is occupied considerable advantage in 10-60 programming time, (2) Plan A gets better production capacity in 70-360 programming time, and (3) The programming time is almost no effect the production capacity when the programming time is more than 330 seconds.

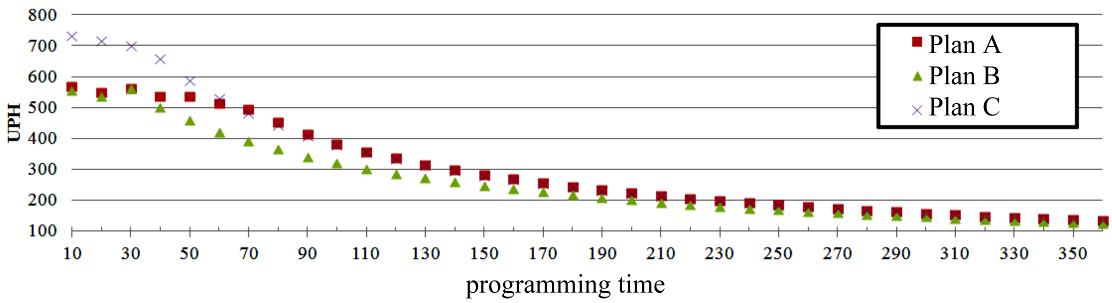


Fig. 17. The production capacity compares between three different configuration types

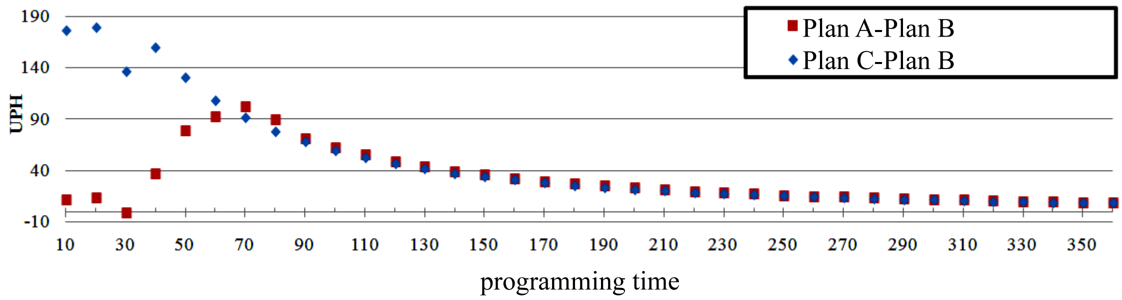


Fig. 18. The difference between three configuration types

3.2 Eight programmers experiment

In this study, there are total eight programmers ($n_p = 8$) designed in equipment. Each programmer is have same programming time ($t_p^1 = t_p^2 = t_p^3 = t_p^4 = t_p^5 = t_p^6 = t_p^7 = t_p^8$), and four sockets ($n_s = 4$). There are also three configuration types of equipment: (1) Plan A, (2) Plan B, and (3) Plan C. They are described as follows:

(1) Plan A

As the Plan A of four programmers experiment, the configuration sequence of Plan A of eight programmers experiment is adding four programmers in right side. As shown in Fig. 18. The experimental results with 38 programming times are tabulated in Table 4 and Fig. 19.

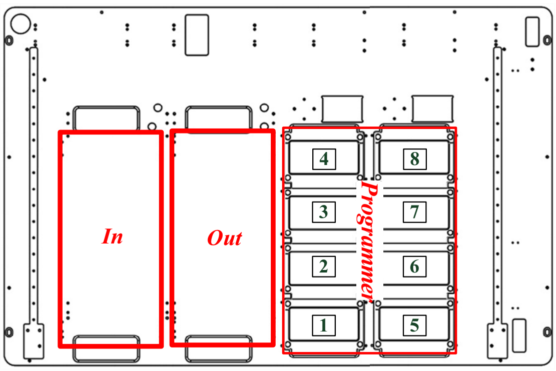


Fig. 19. Configuration of Plan A

Table 4. Experiment results of Plan A with eight programmers

t_p^ℓ	10s	20s	30s	40s	50s	60s	70s	80s	90s
UPH	569.95	550.59	562.76	537.88	537.88	506.51	482.56	476.31	483.34
t_p^ℓ	100s	110s	120s	130s	140s	150s	160s	170s	180s
UPH	440.82	435.51	419.94	403.77	405.42	385.75	371.49	364.30	351.51
t_p^ℓ	190s	200s	210s	220s	230s	240s	250s	260s	270s
UPH	339.40	334.63	324.57	315.10	306.17	297.73	289.74	282.17	274.99
t_p^ℓ	280s	290s	300s	310s	320s	330s	340s	350s	360s
UPH	268.16	261.66	255.47	249.57	243.93	238.54	233.39	228.45	223.72

From the experimental data, the production capacity is decreased gentler than four programmers experiment. Fig. 21 is shown the comparison of experimental data form Table 1 and Table 4. Two capacity data are not much different when programming time less than 50 seconds. Until over 70 seconds, due to the time of programming is much more than the spending time of execution of pick-and-place system, so that eight programmers is holding better production capacity than four.

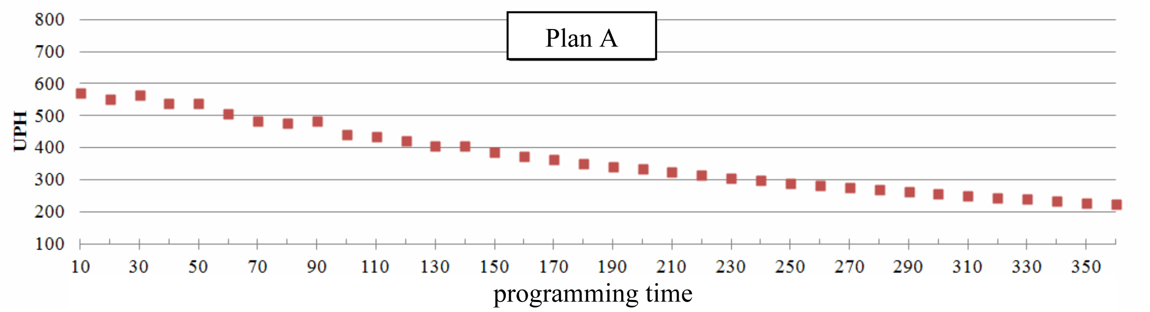


Fig. 20. Experiment line chart of Plan A with eight programmers

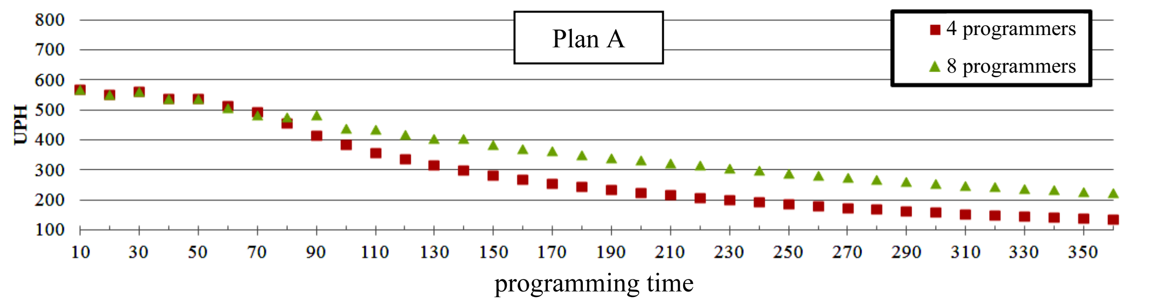


Fig. 21. The comparison of experimental data form Table 1 and Table 4

(2) Plan B

As the Plan B of four programmers experiment, the configuration sequence of Plan B of eight programmers experiment is adding four programmers in right side. As shown in Fig. 22. The experimental results with 36 programming times are tabulated in Table 5 and Fig. 23.

Fig. 24 shows the comparison of experimental data form Table 2 and Table 5. Two capacity data are not much different when programming time less than 20 seconds. Four programmers configuration is better than eight in 30 and 40 programming time. After that, eight programmers is holding better production capacity than four.

(3) Plan C

As the Plan C of four programmers experiment, the configuration sequence of Plan C of eight programmers experiment is also adding four programming in middle. As shown in Fig. 25. The experimental results with 36 programming times are tabulated in Table 6 and Fig. 26.

Fig. 27 shows the comparison of experimental data form Table 3 and Table 6. The capacity data of four programmers is over than eight programmers 50 UPH when programming time less than 40 seconds.

This is a fairly large value. The difference is caused by short programming time and configuration types of Plan C. The short programming time is only needed four programmers. The configuration of eight programmers is caused longer distance between In Tray and Out Tray. The production capacity is significantly reduced by the moving distance of one row programmers.

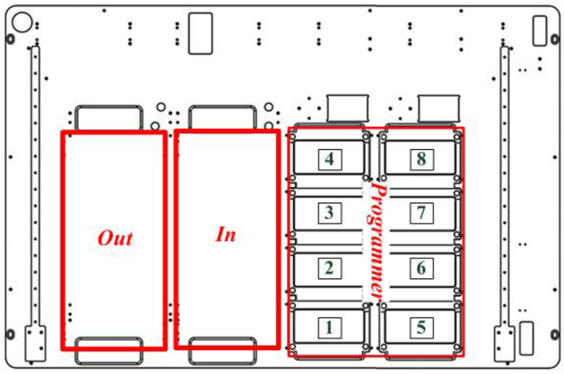


Fig. 22. Configuration of Plan B

Table 5. Experiment results of Plan B with eight programmers

t_p^ℓ	10s	20s	30s	40s	50s	60s	70s	80s	90s
UPH	558.06	537.18	515.96	495.28	506.22	470.79	423.73	393.90	455.59
t_p^ℓ	100s	110s	120s	130s	140s	150s	160s	170s	180s
UPH	430.33	413.84	398.57	384.39	371.17	358.84	347.30	336.48	326.32
t_p^ℓ	190s	200s	210s	220s	230s	240s	250s	260s	270s
UPH	316.75	307.72	299.20	291.13	283.49	276.24	269.35	262.79	256.55
t_p^ℓ	280s	290s	300s	310s	320s	330s	340s	350s	360s
UPH	250.60	244.92	239.48	234.29	229.31	224.55	219.97	215.58	211.36

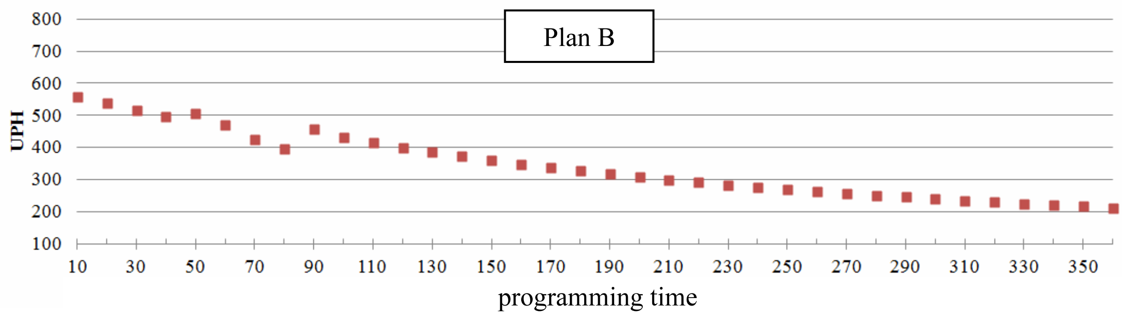


Fig. 23. Experiment line chart of Plan B with eight programmers

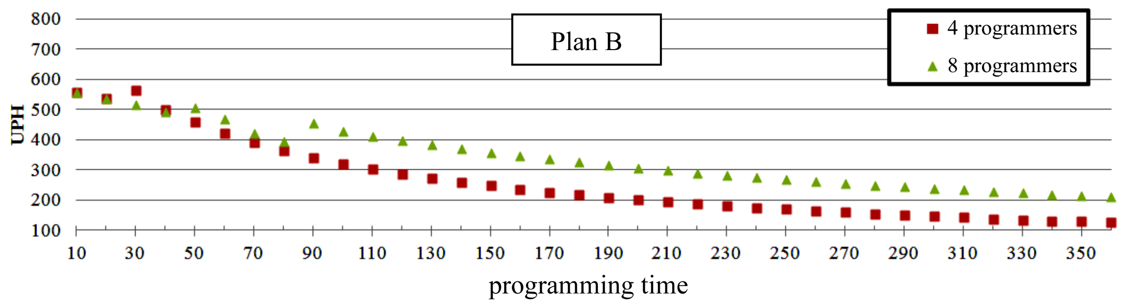


Fig. 24. The comparison of experimental data form Table 2 and Table 5

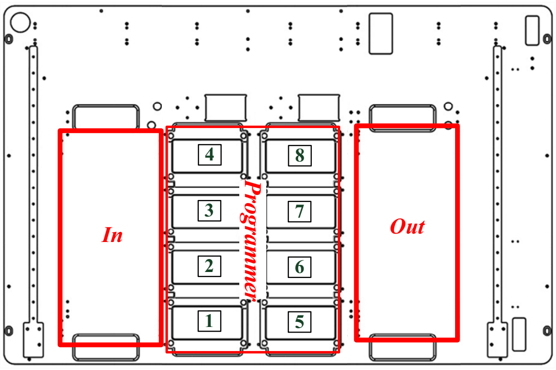


Fig. 25. Configuration of Plan C

Table 6. Experiment results of Plan C with eight programmers

t_p^ℓ	10s	20s	30s	40s	50s	60s	70s	80s	90s
UPH	588.03	576.16	556.15	551.96	578.10	554.27	511.32	496.93	584.77
t_p^ℓ	100s	110s	120s	130s	140s	150s	160s	170s	180s
UPH	560.48	532.83	507.78	484.98	464.13	445.01	427.40	411.13	396.05
t_p^ℓ	190s	200s	210s	220s	230s	240s	250s	260s	270s
UPH	382.04	368.99	356.80	345.39	334.69	324.63	315.15	306.22	297.77
t_p^ℓ	280s	290s	300s	310s	320s	330s	340s	350s	360s
UPH	289.78	282.21	275.03	268.20	261.70	255.51	249.60	243.96	238.57

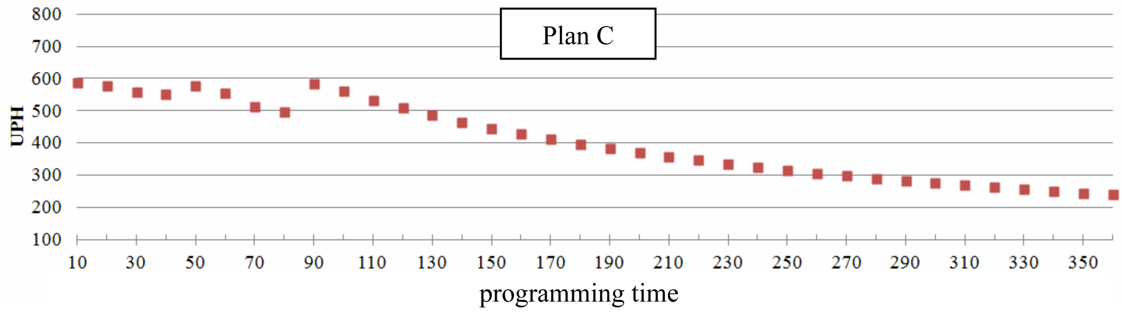


Fig. 26. Experiment line chart of Plan C with eight programmers

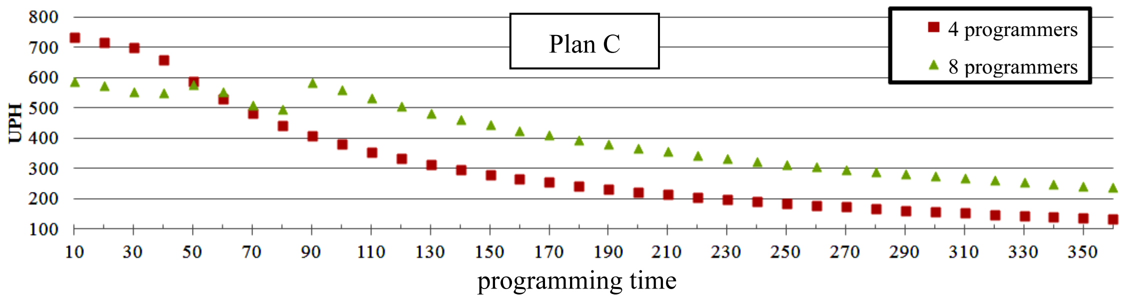


Fig. 27. The comparison of experimental data form Table 3 and Table 6

Fig. 28 and Fig. 29 are comparison chars of three experiments. Three methods are quite different in the 210 seconds or less. Based on these data, the following information is observed: (1) Plan C gets better production capacity, (2) The excessive programmer actually helped to reduce production capacity, and (3) The production capacity in 80 seconds is less than 90 seconds. The configuration type will affect the production capacity up to 100 UPH.

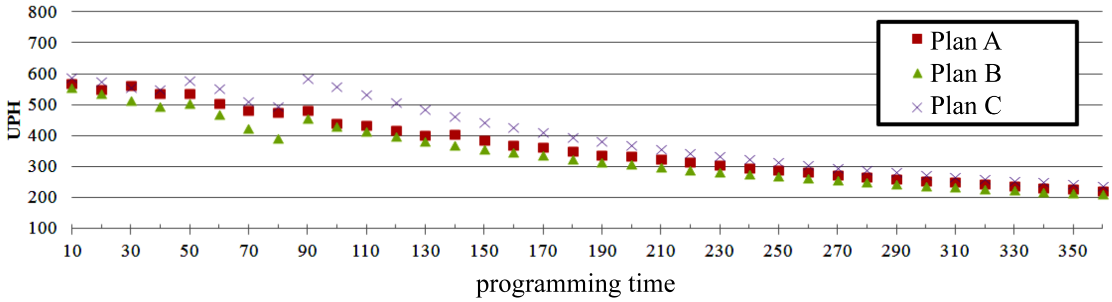


Fig. 28. The production capacity compares between three different configuration types

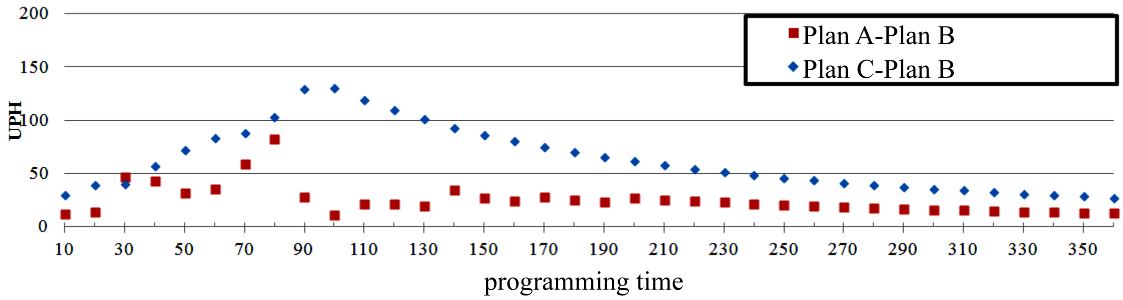


Fig. 29. The difference between three configuration types

3.3 Discussion of experiments

The following two conditions are observed through the experimental data.

(1) When the programming time is bigger than the spending time of execution of pick-and-place system, the equipment is almost waited the programming finish. If the number of programmer is not enough, it gets lower productivity.

(2) When the programming time is less than the spending time of execution, the equipment is almost moving the finished chips. If the number of programmer is too much, that would also cause a lower productivity.

Therefore, in order to design better capacity equipment, the following two directions is summarized.

(1) The optimum configuration of common programming equipment.

Based on the simulation results in Section 3.1 and Section 3.2, a good configuration of common programming equipment is that the programmers are installed in the one side (Plan A or Plan B). Although the production capacity is not the best, but it is a better choice for all programming time.

(2) The appropriate configuration of special programming equipment.

When the last programmer starts to work and the first one finishes the programming at the same time, it is an appropriate configuration of special programming equipment. Such situation would make pick-and-place system working continued. The equipment will have continuing output, so the productivity is the best. Less number of programmers is sufficient for using in the case of lower programming time. Gradually increase the programming time can achieve better capacity when placed more programmers. The case of four programmers is discussed as follows.

If the programming time is $t_p^\ell = 60$ seconds, then the number of programmer is selected by:

$$n_p = \frac{t_p^\ell}{t_m n_s} \tag{18}$$

where t_m is the spending time of maximum moving distance of equipment. The mathematical relationship of t is set by:

$$t_m = \frac{2d_{\max}}{v_a} \quad \textbf{(19)}$$

where $d_{\max}$ is the maximum moving distance of equipment. Considering four programmers and Plan A configuration as shown in Fig. 30, $d_{\max}$ is set by

$$d_{\max} = \sqrt{(x_{56}^{in} - x_1^4)^2 + (y_{56}^{in} - y_1^4)^2} \quad \textbf{(20)}$$

Then n_p could get a scope by

$$n_p = \frac{t_p^{\ell} v_a}{2n_s\sqrt{(x_{56}^{in} - x_1^4)^2 + (y_{56}^{in} - y_1^4)^2}} \quad \textbf{(21)}$$

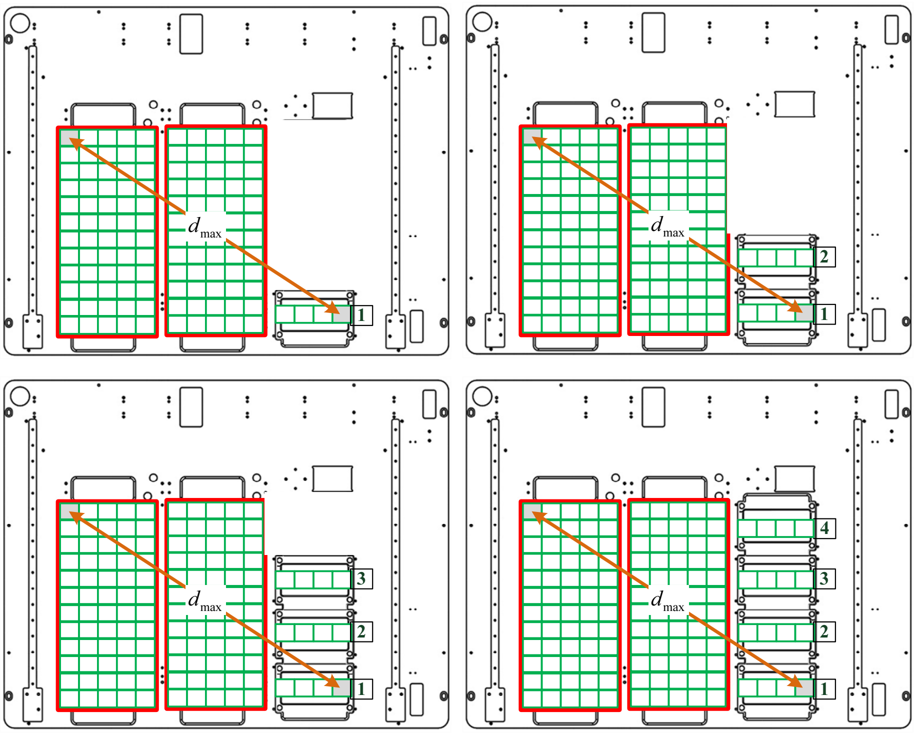


Fig. 30. Description of the maximum moving distance of equipment with four programmers and Plan A configuration

For example, if v_a=165, and $n_s = 4$, and the coordination of Plan A of four programmers are considered, the result of the calculation is shown in Table 7. The right number of programmer is 2.53 units. Table 8 is shown the result of the appropriate maximum number through the simulator. The production capacity of two programmers is more than one programmer 127.87 UPH and promotion is 0.72. The promotion of three programmers is more than one 0.88. If the programmer number increases to four, the production capacity is only rise up 53.24 UPH and promotion in only 0.30. Obviously, four programmers not only don't help the capacity promotion, but also rise up the cost of equipment. Two or three programmers are the appropriate number of the equipment with 60 seconds programming time.

Table 7. The result of the calculation

x_{56}^{in}	y_{56}^{in}	x_1^4	y_1^4	$d_{\max}$	t_m	$t_m n_s$	$t_p^{\ell} / t_m n_s$
68.77	338.30	468.8	60	487.30	5.90	23.62	2.53

Table 8. Production capacity compare table

n_p	a_{UPH}	Differences $a_{UPH}(n_p) - a_{UPH}(n_p - 1)$	Proportion of promotion (differences/ $a_{UPH}(1)$)
1	177.20	--	--
2	305.07	127.87	0.72
3	462.03	156.96	0.88
4	515.27	53.24	0.30

4. Conclusions

In this paper, an equipment simulator is designed for a small-sized chip automatic programming equipment to verify different configuration of equipment and different programming time. The chip automatic programming equipment is combined with a pick-and-place system, two trays, and several programmers. In this paper, an equipment simulator is designed to simulate different configuration of programming equipment. Two cases with four or eight programmers are presented in the experiment, and each one with three configuration types. The highest priority behavior is to pick up the chip from the programmer and place it into the Out Tray. From the experimental result, different programming time is suitable different number of programmers and different configuration. Therefore, the equation is presented the relationship between the programming time and programmer number in this paper. The results illustrate that the proposed equipment simulator can help to find the balance between the programming performance and the cost of the equipment.

Acknowledgment

This paper was partly supported by the Ministry of Science and Technology (MOST) of the Republic of China under contract MOST 103-2622-E-032-001-CC3.

References

[1] P. Pellati, Automated test equipment for research on nonvolatile memories, IEEE Transactions on Instrumentation and Measurement 50(5)(2001) 1162-1166.

[2] S. Chand, H. Schneeberger, Single machine scheduling to minimize weighted earliness subject earliness subject to no tardy jobs, Theory and Methodology 34(2)(1986) 221-230.

[3] D. Biskup, Single-machine scheduling with learning considerations, European Journal of Operational Research Society 115(1)(1999) 173-178.

[4] A. Bachman, A. Janiak, Scheduling jobs with position-dependent processing times, Journal of the Operational Research Society 55(2004) 257-264.

[5] Topline IC Tray, BGA Tray: JEDEC Matrix IC Trays, <http://www.topline.tv/drawings/pdf/trays/Tray_JEDEC_Matric_IC_Tray_ALL.pdf>.