Terraria circuit documentation

The most comprehensive **Terraria** circuit reference

Author:putianyi888 Organization:Terraria Circuit Enthusiast Acousse Group (231355279)Time:September4, 2020 Release:3.1

ElegantLaTeX Program

LaTeX 模板:*ElegantBook*

The record

[1]Chapter starts from scratch

1.1 Foreword

注意本书初编写时的游戏是 1.3.5.3 版本。 1.4 After the update, we are working hard to learn new content and update it.

[1] tModLoader、CheatSheet、HERO’s courage

Refer to the https://github.com/putianyi889/TerrariaWiringTut for updates to 1.4 in the documentation

orial/issues/30

This book is positioned as a "document" and is mainly used for systematic ingestion circuit theory, so the text will first talk about the theory and then talk about examples. If the reader finds the theory difficult to understand, he may wish to look at the example first, and then combine the example to see the theory.

After each chapter of thinking, part of the classical circuit analysis and understanding, partly because I am lazy and did not do the circuit, and part of the pure theoretical derive. What they have in common is that it doesn't matter if they don't do it. Compared with the thinking question, the circuit used in the body for example, self-contained section, must be mastered.

After learning the circuit foundation, you can make most of the puzzle maps and circuits in the brush field, including pumpkin gods. After learning logic settlement, you will be able to study the design of complex circuits, such as circuit games. Digital circuits are difficult to read only for capable readers. Don't ignore the appendix, which contains a large number of valuable links and game mechanics that will help you learn the circuit and design some venues.

Since the official information of Tyralia is all in English, and almost all of the English materials are not translated, it is important to prepare a dictionary and english standards above junior high school when conducting in-depth study of Tyralia. At the same time, certain computer or mathematical expertise can help. If you still can't read English with the help of a dictionary, it's recommended to ask someone for help, rather than using a machine flip, which basically doesn't have a word that is accurate. https://github.com/p utianyi889/TMECbackup has compiled some of the posts in the circuit section of the official English forum in Tyraya, where you can turn for translation.

If you're having trouble accepting plain text content, you can also go and watch a video tutorial linked to the appendix. The disadvantage of video relative to text is mainly timeliness, because video is not easy to change, so the technology in video is often out of date technology. We still recommend this book as the main reference after understanding the content of the video tutorial.

All game nouns in this book are preferred for the latest version of Tyraria's Chinese on the Steam platform, followed by Chinese wiki.

In addition, map editors and modules 1 are essential for students who want to make large installations, and they can help you build and back up quickly.

Regarding the game mechanics, if you have a programming foundation, looking at the decompiled c-source code is the most reliable method. Otherwise, refer to wikis and ask someone who understands the source code if you have questions about them. The game mechanisms in the appendix are obtained through the source code.

The body section of this book focuses on circuits, and information other than circuits is discussed in the appendix.

In https://github.com/putianyi889/TerrariaWiringTutorial assisting in writing this book is the only way to support us. You can create proactively or pick up tasks in Issues. If you don't use

GitHub,you can watch tutorial https://zhuanlan.zhihu.com/p/34693871. Follow the book The GitHub project provides instant access to updates.

1.2 Some basic concepts and mechanisms

1.2.1 Entity

An entity is an object that can [2]collide, including but not limited to NPC, person, projectile, object, or diagram.

Slim's contact injury to the player is the collision between Slim(NPC)and theplayer; the player hits Slim witha bow out of the wooden arrow, which is the collision of Slim(NPC)and the wood arrow (projectile); the player is hit by a flesh-and-blood wall laser, which is the collision of the player with the laser (projectile); the player, drop, most NPC,most projectiles cannot penetrate the wall because the player, drop,NPC,projectile and the plot collide.

Collisions are determined by the collision box, such as Alem's collision with the player, because Slim's collision box overlaps the player's collision box. The collision box for all entities in Tyralia is rectangular and has two properties, width and height, which can be found in the source code.

1.2.2 Hard and soft upper limits

Many entities in Tel Rea have a cap on quantity, which is divided into hard and soft caps. A hard cap is a static constant in a game, such as a NPC hard cap of 200, and so on. A soft upper limit is a variable in the game, such as the number of active brush monsters (typically 5 to 15).

From a program perspective, the hard upper limit is determined by the length of the array of the C? To avoid crashes during normal games, the game program has cross-border checks in key functions, such as when NPC reaches the upper limit, the game refuses to generate a new NPC to prevent crashes. The soft upper limit is the developer's balanced control of the game, such as the number of active enemies near the player reaches 15 without brushing the monster.

The known hard cap is shown in Table 1.1.

Table **1.1:** Hard Upper Limit in Tyralia

object	upper limit
buff	22

[2] Strictly speaking, entities are programming terms, and here is simply a simplification without compromising game understanding。

Puppet shadow	100
NPC	200
gamer	255
Drops	400
Projectiles	1000
Cooling organ	1000
Chest	1000
Active liquid	5000
Buffer the liquid	10000

1.2.3 Image Frame / **Physical** Frame

Image Frame refers to each frame screen updated by the computer display during the Terraria game, where pressing F10 shows the current image frame rate in the lower left corner of the game window.

A physical frame (tick) is the smallest unit of time in Tyraya, at 1/60 seconds. At the scale of the physical frame, Tyralia is a turn game, in each physical frame, all the game mechanism except the circuit will be settled in a certain order. Since this book is a discussion of game mechanics, physical frames are represented directly by "frames" without special instructions.

When the game is running, the graphics card is responsible for image processing and theCPU is responsible for the mechanism of settlement. If both processes can be completed in 1/60 seconds, the physical frame rate is 60. If the CPU processing time is greater than 1/60 seconds, the corresponding physical frame rate is reduced. If the graphics card takes longer than 1/60 seconds, the result depends on whether the frame is jumped or not. If jump frames are opened in the settings, the image frame rate is appropriately reduced to maintain the highest physical frame rate that the CPU can achieve. If you turn off jump frames, the image and physical frames are fully synchronized, with real-time game speeds based on the slowest speed in the graphics card and CPU.

1.2.4 coordinates

Coordinates are the positions in the world. Coordinates are not as shown by the depth gauge and compass. The coordinates in the program are (0, 0)in the upper left corner ofthe world, with horizontal coordinates to the right and ordinates down. The world width is maxTilesX grid and the height is maxTilesY

case. The coordinates in the lower right corner of the world are (maxTilesX, maxTilesY).

The central coordinate layering in Tyraya is generally divided into five layers: space, surface, underground, cave, and hell. In the game mechanism, there are only two thresholds, one is the ordinate of the junction of the surface layer and the subsurface, called worldSurface,and the other is the ordinate of the junction of the subsurface and the cave layer, called rockLayer. The height of the space layer is obtained by multiplying rockLayer by one factor; This coefficient is generally 0.35and the constant is generally 200, but may vary in different situations.

1.2.5 Measure

The units of length in Tyralia are: miles, grids, feet, and pixels. The conversion relationship is 1 foot , 8 pixels , 1/2

Grids : 1/5280 miles.

The units of time in Tyralia are frames, seconds, minutes, and days. The conversion relationship is 1 day , 24 minutes,1 minute , 60 seconds,

1 second s 60 frames. Sometimes frames, seconds, minutes are also called game seconds, game minutes, game time.

Speed units are length units / time units, mainly: miles / hour, grid / second, pixels / frames.

The units of length inside the program are pixels, the units of time are frames, and the speed units are pixels/frames.

1.2.6 Drive

A drive (engine) is a device that automatically activates the circuit intermittently. Drives are classified by frequency into low-frequency drives, high-frequency drives, full-frequency drives, and overclocking drives.

- Low-frequency drive refers to a drive with a frequency less than or equal to 4Hz. This type of drive is generally obtained by the timer down frequency.
- High-frequency drives are drives with frequencies greater than 4Hz and less than 60Hz. This type of drive is very rich, the most reliable and stable way is to use full-frequency drive down frequency.
- Full-frequency drive refers to a drive with a frequency equal to 60Hz. Mainstream full-frequency drives have dummy drives and conveyor belt drives. Full-frequency drivers are called because they drive at the same frequency as physical frame rates. Higher frequencies can also be used to get the same effect with full-frequency drives. For example, a120Hz drive has the same output as two full-frequency drivers at the same time.
- Overclocking drives are drives with frequencies greater than 60Hz. Such drives typically operate simultaneously with multiple full-frequency drives, or take advantage of the

supersensitivity of the pressure plate. There are currently no instances of overclocked applications.

1.2.7 Half Brick

When the drop/non-walled creature's collision box coincides with the solid block, the program attempts to push the collision box away from the solid block. Starting with version 1.2, most foreground blocks have six half-brick forms, each with a different mechanism for pushing away from the collision box. Although half-bricks have a place in the circuit, because of their own: not related to the circuit itself; [3] For half-brick-related research and tutorials, readers can refer to the links below, which are sorted randomly.

- Video
  - Do you use half bricks?-Zerogravitas https://www.bilibili.com/video/av22088325
  - Item half brick as well as some interesting apps!-Zerogravitas https://www.bilibili.com/vid eo/av2739847 Ghost Animal Stone Giant! -Zerogravitas https://www.bilibili.com/video/av2208854 7
- essay
  - Experiment: Half-brick nature explores https://tieba.baidu.com/p/3603529198
  - HOIK! [Guide] - Rapid Player/NPC/Etc Transport Using Only Sloped Tiles. https://forums.terraria.org/index.php?threads/hoik-guide-rapid-playernpc-etc-transport-using-only-sloped-tiles.1656/
  - [Early Game] Anti-Monster Wall Defense (Using HOIK! and Stair Glitch) https://forums.terraria.org/index.php?threads/early-game-anti-monster-wa ll-defense-using-hoik-and-stair-glitch.29917/
  - Hoik tracks with pressure plates for mounts https://forums.terraria.org/index.php?threads/hoik-tracks-with-pressure-plates-for-mounts.3711 3/

1.2.8 Projectile Generation and Refresh Mechanism

Projectiles are a large class of entities in Tyralia. These include darts, flames, thrown balls, lasers fired through prisms, beach balls thrown out, waving sun-flaming blades, and even tombstones that bounce after a player's death. This section is mainly for the generation and

[3] Most of its functions can be solved with conveyors or conveyors。

refresh process of all projectiles, for subsequent references to certain content, readers can skip this section directly.

The game uses a list of 1000 lengths to store projectiles. When a projectile is generated, its information is stored in the first empty space of the projectile list. If there is no space in the projectile list, the projectile does not generate .

For each physical frame, the game performs a round of projectile refreshes, which scan the list from start to finish, and perform its refresh function on each projectile. How each projectile is refreshed depends on the idof theprojectile. The first step in firing refresh is to change the position of the projectile, i.e.

New position s original position s speed .

The speed of a projectile that is not moving at a uniform speed in a straight line also needs to be calculated. The second step is to make a collision determination, if the projectile collides with a creature, it may have to be damaged, if it collides with a foreground object, it may have to destroy the projectile, and if it collides with a turquoise pressure pad, it may have to be inserted into the circuit settlement. This last case is the main concern of the book.

The refresh mechanism for a projectile is listed in the appendix.

Chapter 2 Circuit Foundation

The

- power wire /Electrical
- Example of the
- Logical door

2.1 Power / Wire / Appliances

In real life, the three basic components of the circuit are power supply, conductive circuit and electrical appliances, the current generated by the power supply is transmitted through the circuit to electrical appliances and drive electrical appliances. In Tyralia, the three basic components of the circuit are power supplies, wires, and electrical appliances, and the electrical signal generated by the power supply is transmitted by wire to the electrical appliance, which responds to the electrical signal with electrical appliances. This process is described on the wiki as a power activation -- wire activation on the power supply -- electrical activation under the wire. In the next discussion of this book, we use the explanation on the wiki to use activation in uncounted to express the active state of the circuit.

Because wires and tuggers are on different layers, they can overlap. If a wire overlaps a grid, we say that the grid is under the wire, and the wire is on that grid.

A power supply is an item that activates the grid or the corresponding item on its wire, each with its own activation conditions.

An electrical appliance is an item that can be activated by a wire on it or an item corresponding to that grid, and each appliance is activated with its own unique response. All power supplies in Tyralria and their activation conditions are shown in Table 2.1,and all appliances and how they respond can be found in Table 2.2,see Circuit items for details and wikis.

Table **2.1:** Power supply in Tyraria

power supply	The activation condition
Switch / Joystick / Trapped Chest / Dead Man's Chest	Right-click
Gray / Brown / Blue / Jungle Lizard Pressure Plate	Player stampede
Red /Green Pressure Plate	Player /NPC/Enemy Stampede
Yellow pressure plate	NPC/Enemy Stampede
Increase the pressure plate	Players step on or off

Turquoise pressure pad	The projectile touched
Doors / organ doors / high doors	On/ Concern change
timer	Activate every once in a while after turning on
Detonator	Players hit top-down or right-click
Gem lock	The corresponding large gemstone is embedded or removed
Logic sensor (day/night).	Day / Night
Logic sensor (player).	Players enter or out of the blue box
power supply	The activation condition
Liquid sensor	The corresponding liquid enters or leaves
Logic gate	See below for details

Table **2.2:** Appliances used in Tyraria

Electrical appliances	How to respond
Some light sources	On/off switch off
Doors / organ doors / high doors	On/ Concern change
pump	Transfer the liquid from the incoming pump to the outlet pump
Grille	On/ Concern change
organ	Launch a projectile / explode and disappear
fort	Change direction /shot according to the activation point
Fireworks fountains, fireworks boxes	Produce fireworks
Fireworks rockets	launch
Bubble machine / budding balloon machine / party center	On/Off, generate background effects
Fog Machine	On/Off, generates cloud effects
fountain	On / Concern Change, change the color of the water
music box	On / Concern Change, change background music

Part of the statue	Build items / Transfer Town NPC/ Light Off Switch / Generate Enemy Monsters / Generate Small Animals
chimney	Three state switches
Conveyor	Swap players and NPC on two conveyors
The pillar of the sky	On/Off Change, Change Background and Filter
Broadcast box	Displays a text message
conveyor belt	Change direction
brake	Switch the virtual state of the grid
Colored light bulb	Each of the four-color wires controls the lighting of a light bulb
Pixel box	Off when activated from left/right, lit when activated from top/bottom and left/right
Logic light	See below for details
The intersection of the minecart track	Switch between two crossovers

The simplest circuit consists of a power supply, an electrical appliance, and a wire that connects the power supply to the appliance. In this circuit, the moment the power supply is activated, the appliance is automatically activated. Readers can try connecting a wide variety of power and appliances with wires, and then try to activate the power supply to see what happens.

Example **2.1** Put the number "0"(Figure 2.1 (a)with a torchand connect a switch with a wire to the left of "0" (Figure 2.1 (b)). In this circuit, the switch is the power supply, andthe torch on the left side of the"0" is electrical. Because the activation condition of the switch is right-click, we can activate it by right-clicking the switch, and then the red line on the switch is activated, so the left three edges of "0" under the red line are activated. The activation response of the torch is the light-out switch, so the three sides go out, showing the number"1" (Figure 2.1(c).). Repeatedly right-click the switch, the torches will switch between light and extinguish, thus showing the number in

Switchbetween 0 and 1. This is the simplest binary number display, which is short for digital display.

(a) (b) (c)

Figure **2.1**

Note If you want to display a more striking effect, you can paint both the background wall and torch with shadow paint, or you can try other light sources, even a combination of brakes and solid blocks.

Example **2.2** Connects the logic sensor (night) and the party center with wires, as shown in Figure 2.2. In this circuit, the logic sensor (night) is the power supply and the party center is the appliance. At night, the party event ends, the party center automatically shuts down, then the logic sensor (night) is activated, the party center is activated, and the party center is reopened to enter the party event. Repeat this process every day into the night and the map will remain in the party event. Note Note It is important to note that the unit is capable of achieving the desired functions, depending on

Figure **2.2**

Depends on the night-time event settlement before the sensor settlement.

Example **2.3** Conveyor drive

As shown in Figure 2.3, wire the conveyor belt (clockwise) and the pressure plate. In this circuit, the pressure plate is the power supply and the conveyor belt is an electrical appliance. The character stands on the conveyor belt. The conveyor belt makes the character move to the right, the character enters the grid where the pressure plate is located when the pressure plate is activated, and then activates the conveyor belt, the conveyor belt becomes counterclockwise, the character moves to the left, the pressure plate is activated when the pressure plate is removed, the conveyor belt becomes clockwise, the character moves to the right, so repeatedly.

Figure **2.3:** Conveyor drive. The conveyor belt in the figure is clockwise.

This circuit is called "conveyor drive". Each time the pressure plate is activated, the red line is activated, and if left unoperated, the circuit is activated indefinitely, and the circuit that is infinitely activated is called drive. The drive as a whole can act as a broad power supply, releasing signals outwards. The frequency at which the signal is released is the drive frequency. The signal on the red line in Figure 2.3 is the driving signal.

In conveyor drive, the "instant" in which the character enters the pressure plate moves to the left and the "instant" that moves out of the pressure plate is to the right, so the frequency of the drive is equal to the frequency at which the game determines whether the character and the pressure plate overlap, i.e. the physical frame rate.
A driver with an equal frequency and physical frame rate is called a full-frequency driver. Conveyor drive is the simplest and smallest full-frequency drive.

A broad power supply refers to a module that acts as a power supply, such as a driver. In a broad sense, an electrical appliance refers to a module that acts as an electrical appliance, such as a display.

Some appliances switch between two states when activated, such as light-emitting items and functional items switch. When it comes to logic, you can think of two states as 1 and 0.

Example **2.4** Automatic door

When a lot of people first start playing Tyraya, building a house will build a door. Ordinary doors are very inconvenient, not only access needs to operate, but also sometimes can not prevent enemy monsters from entering. After the mechanic was rescued, some people began to change the ordinary door into a block with brakes (Figure 2.4), so that there can be obstacles on both sides of the door, and will not be opened by enemies.

Figure **2.4:** The door is made with brakes and the switch can open and close the door.

Some people don't even bother to get started right, so they invented an automatic door that opens automatically when a player gets started and closes automatically when they get started.

This can be done using two pressure plates. As shown in Figure 2.5, when the player walks from the left to the door, the pressure plate on the left side of the door is applied, and the pressure plate activates the brakes to virtualize the object. When the player passes through the door, step on the pressure plate on the right side of the door, the pressure plate activates the brakes and solidifies the block. The player goes the same way from right to left.

Figure **2.5**

Thoughtful people will soon find a loophole in the door, that is, if people go to the door, and go back, the door will remain open. To solve this problem, the ordinary pressure plate needs to be changed to an aggravating pressure plate (Figure 2.6).

But changing to an increased pressure plate is not a big deal. If you enter a multiplayer game where two people stand on either side of the door while trying to reach each other, the brakes are activated twice and the object remains solid unless one person gives in. This is very inconvenient. This problem can be solved using a logic sensor (player) (Figure 2.7). When there is already a player station

Figure **2.6**

When inside the sensor's blue box, other players enter and leave the sensor state unchanged, and the sensor will not be activated naturally. Therefore, when there is a player in the blue box of the sensor, the door opens or the door closes.

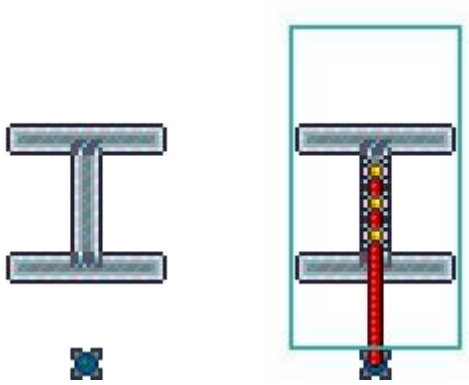


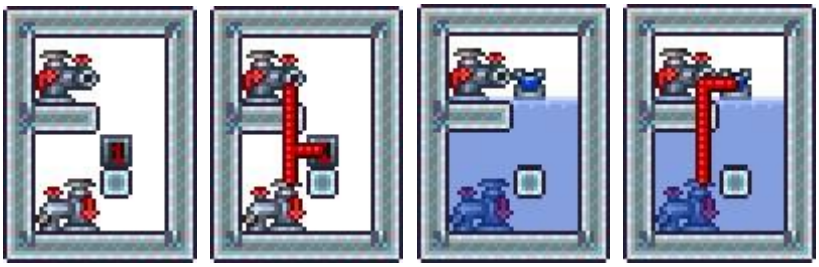
Figure **2.7**

Example **2.5 Printing** liquid

The basic principle of brushing liquids is the "metaphysical" flow mechanism of liquids in Tyraya. Despite **???The Fact That???S Not Going To Be The Same** The flow mechanism of liquid is introduced, but because there are too many liquid grids involved in the game environment and the model is too complex, there is no perfect theory to predict the behavior of liquid simply. In general, diversion leads to an increase in liquid, while excessive diversion leads to a decrease in liquid. Most of the brushing machines use solid blocks to divert liquids. As the mechanism of liquid is not clear, the diversion structure of the brush liquid machine is also based on experience. The brush liquid machine is described here with a focus on how the circuit is activated, so the shunt structure is from Jane.

First we use blocks to build a small pool with a shunt and put the water pump on the bottom, the pump on the top, and set up a 1-second timer (Figure 2.8 (a)). Pour enough water into the pool, then wire the pump, outlet pump, and 1-second timer (Figure 2.8(b)and right-click on the 1-second timer, so every second, the water on the pump is transferred to the pump and the water brush starts to work. A grid wall can be virtualized to fetch water. Brush lava recommends a 3-second timer and a 5-second timer for brush honey due to the different flow rates of different liquids.

One of the small drawbacks of this constructed brush liquid machine is that it needs to turn on the timer manually, as it automatically turns off when exiting the map. On the other hand, when the water is filled with water, the pump does not work, but the timer is still running, which is a blow to obsessive compulsive disorder. One way to solve this problem is to set up a liquid sensor on the edge of the pump (Figure 2.8 (c), Figure 2.8 (d). If the water level is high enough, the fluid sensor remains bright and the pump does not work. If the water intake causes the water level to drop below the liquid sensor, then the liquid sensor goes out, activates the pump, pumps the water, lights up the liquid sensor when the pumped water passes through the liquid sensor, and the liquid sensor activates the water pump until the water level reaches the liquid sensor, which does not activate the circuit because it remains on. Not only does this device allow for intelligent brushing, but the liquid sensor is the driving frequency that perfectly adapts the flow rate of the liquid.



(a) (b) (c) (d) 图 2.8

2.2 Logic door lamp / logic door

(a) Logic Light (b) Logic Gate

Figure 2.9

Logic door lamps are electrical appliances, referred to simply as logic lights. Logic lights can also be divided into normal logic lights and fault logic lights. Switch between on/off when the normal logic light is activated, and the status is set to "Active" (no image effect) when the fault logic light is activated. Logic lights must be stacked on the logic gate, which we say is on the logical door, and the logical door is under these logical lights.

Logic doors are power supplies, and different logic doors have different logic determination rules. If there is no faulty logic light on a logic gate, we call it a normal logic gate. The logic gates discussed in this section are normal logic gates. When the state of the logic lamp on a logic gate changes, the logic gate makes a logical determination and adjusts its own light-out state. If the state of the logic gate changes, the logical gate is activated.

The logical decision rules for the six common logic gates are the item description and wiki.

Example 2.6 Improve the automatic door

In Example 2.4, we use a logic sensor (player) to implement the function of an automatic door. The advantage of this automatic door is that the sensor's sensing range is too large, causing the player to open the door at a distance, even where he or she does not want to open it (above and below the door). The induction range of the pressure plate is appropriate, but it can cause the door to close when people stand on both sides of the door at the same time. What we need to do is open the door when the left or right station people are on the left side of the door. Therefore, the use or door can meet our requirements.

Figure **2.10**

The circuit is shown in Figure 2.10. The circuit is divided into three sub-modules: the accent pressure plate on the left, the blue line, and the logic lamp at the top make up the first submodule: the pressure plate is the power supply and the logic lamp is an electrical appliance. The logic light goes out when there is no stand on the pressure plate. When the character enters/leaves the pressure plate range, the pressure plate is activated to switch the logic light state above. To sum up, the logic light above goes out - no one on the left, the logic light on the top is on the left.

- The accent pressure plate on the right, the red line, and the lowermost logic lamp form a second submodule. Similar to before, it can be analyzed that the logical light below goes out - no one on the right, the logical light on the bottom is on the right.
- or doors, green lines, and brakes to form a third submodule: or doors are power supplies, brakes are electrical appliances. Or the door is on

  When the switch is off, it activates so that the actuator switches between the virtual reals, so or the door goes out, or the door is closed, or the door is lit, and the door is open.

Combined with the definition of a door, we have the following conclusion:

- Someone on the left or right is on the upper or lower logical light , and two logic lights have at least one on , or the door is on , the door is open;
- There are no people on either side of the door - the logic light goes out completely , or the door goes out - the door closes .

In previous circuit analyses, we all used "activation" to analyze. Using activation analysis is always true, but sometimes it is not as simple as state analysis. In Figure 2.10, all power supplies and electrical appliances are in two states: the pressure plate is depressed and bounced, the logic lamp and the logic gate are lit and out, and the brakes are solidified and virtualized. At the same time, all power supplies activate the circuit when the state changes, and all electrical appliances change when the circuit is activated. In this case, the state of the power supply and electrical appliances is always synchronized, such as the logic door when the brakes must be virtual state, logic door when the brakes must be solid state; Note Not all power and appliances are switched in two states. Switches have no state difference; chimneys are three states; pixel boxes have two states, but when activated the behavior is assignment rather than switching; ordinary pressure plates are activated only when pressed, not activated when bounced; and so on. State analysis is not a plpotent, it only applies if both power supply and appliances are two-state switching. In most cases, activation analysis is still required.

2.2.1 Diode / wire changer

Before you begin this section, consider the question: How do you turn on more than 4 1-second timers simultaneously with a switch and make them work independently?

To answer this question, first of all, it's clear that if you're not sure what's going to happen, connecting two timers with a wire is very bad. Because the timer is both a power supply and an appliance, the two timers connected together interfere with each other the next time they are activated, causing one of the timers to turn off the other. Since there are only four colored wires in Tyralia, if you want to turn on more than four timers at the same time with one switch, then there must be two timers activated with lines of the same color. But keeping them from interfering with each other means that the two lines can't be connected. At the same time, you must control these two lines with a switch of only 1 grid in size, so what?

The root of the contradiction is that when a switch activates multiple timers through a single wire, it passes through a timer cycle, and the timer interferes with each other through the wires on the switch. To eliminate this interference, the signal can be transmitted from the switch to the timer, and not from the timer to the switch, that is, to achieve diode-like functions.

As shown in Figure 2.11 (a), a diode is made by placing a logic lamp on the door. Connecting the logical light with a red line, connecting the blue line to the door (Figure 2.11 (b), Figure 2.11 (c),activates the red line to cause the blue line to activate, and activating the blue line does not affect the red line. According to the definition of the door, when the logic light is on, when the door is lit, when the logic light goes out and the door goes out. When the red line is activated, the logic light switches between lights off, so that the logic gate also switches between lights off, activating the blue line once. Because the logic gate is not electrical, activating the blue line does not affect the red line.

(a) (b) (c)

Figure **2.11:** 2.11 (b) (c):The switch above controls two torches, while the switch below can control only one torch

This principle makes it easy to turn on many 1-second timers with a switch and keep them from interfering with each other (Figure 2.12).

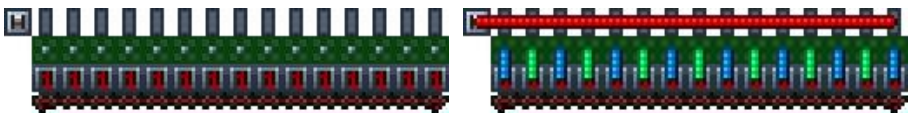


Figure **2.12**

At other times, we use wire changers. For example, we've pre-made a counter (which we'll talk about later), which is entered using a blue line. At the same time we made a drive that used a red line output. To measure the frequency of the drive with a counter, you need to change the counter to a red line input or a drive to a blue line output. This modification can be very difficult when the circuit is more complex, and the wire changer needs to be utilized. Since there is only one-way signal transmission between the driver and the counter, the diodes in Figure 2.11 can be used as wire changers, thus solving the problem of inconsistent wire colors.

Example **2.7** Timer

The timer activates the circuit every 1 second after turning on the 1-second timer. We want to do a 1 second delay and open

After 1 second timer, after 1 second,1 second timer is activated once, and then turned off.

The unit is shown in Figure 2.13. Turn on the 1-second timer, after 1 second, the timer activates the circuit, and then the logic light is activated, and then the logic gate is activated to turn off the 1-second timer.

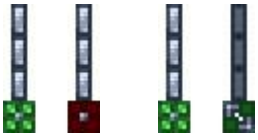
Figure **2.13**

Example **2.8** door drive

We know that if the logic light on the changer is activated, the logic gate is activated. If the logic gate and the logic lamp are connected together, the logic gate activates the logic light again, and the logic light activates the logic gate, thus making a high-frequency drive. Try it in the game and seek an explanation in Logical Settlement.

2.2.2 Absolute equivalent logic gate

There are six common logic gates in Tyraria: and door, and non-door, or door, or non-door, alien or door, different or non-door.



(a) (b) 图 **2.14**

The names of so many doors look so big that you have to be familiar with each door to make a circuit? In fact, many of these six doors are absolutely equivalent. The absolute equivalevalent of the two devices means that if both devices are blacked out, then there is no

way to measure their differences. As shown in Figure 2.14 (a),the left is with the door, and on the right is the non-door, respectively, with the same number and state of the logical lights, the two doors behave exactly the same. The only difference with the door and the non-door is that for the same input, when the door is lit and the non-door must be extinguished, and when the door is out and the non-door must be bright. Because the logic gate is activated when the switch is lit and off, it is always activated at the same time as the door and the non-door. In Tyralia, activation from light to out is no different from activation from out to light, so there is no difference between door and non-door. By the same to say, there is no difference between a door and a non-door, and there is no difference between a different door or a different or non-door.

Now we have narrowed the scope of learning to three doors: with the door, or the door, the difference or the door. Next we look at Figure 2.14 (b),left is with the door, right is or the door, with the door full of lights, or the door is full of lights. Now the two doors are exactly the same. This is because the logic lamps above them have the opposite initial state, and for the same input, the final value is the opposite. So, with the door on⇐⇒ with all the lights on the door⇐⇒ or all the lights on the door⇐⇒ or the door out. This is always the opposite of the light-out state of the door and or the door, as discussed earlier, where the two doors are no different.

There are only two doors left: with doors, different or doors. Is it possible for these two to be absolutely equivalent? Impossible, the reason is in the digital circuit will say. Readers here just need to know that we use the same door and different or door is enough, other ordinary logic door can be replaced with both doors.

2.2.3 Decimal number is displayed

In the previous section, we made a number display that shows binary numbers. With the logic gate, we can do decimal display. There are different approaches to digital visual input, where we use a 4-bit binary input that requires the binary number to display the corresponding decimal number 0 9 at 0000 to 1001,and in other cases the display goes out.

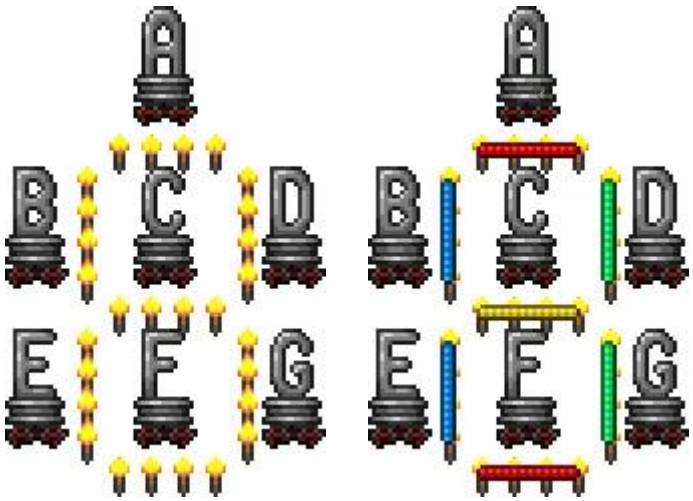
We chose to use a seven-segment line display. First, seven lines are laid out with torches (Figure 2.15 (a)and then each line is controlled separately with seven wires (Figure 2.15 (b)). For 0 9,ayg has a corresponding section to light Table 2.3.

Displays the value	0	1	2	3	4	5	6	7	8	9
Light up the section	abcefg	cf	acdeg	acdfg	bcdf	abdfg	abdefg	acf	abcdefg	abcdfg

Table **2.3**

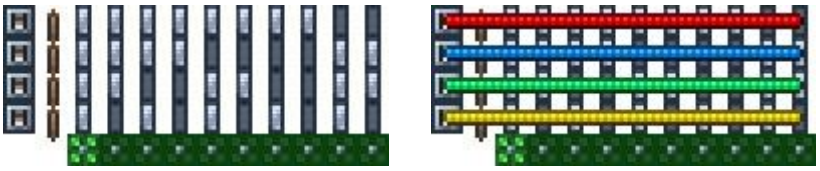
Then we'll process the input. We need to convert the input four-digit binary information into one of the 10 numbers, a process called decoding. We decode the circuit in Figure 2.16.

Next you can wiring. Mark the ten logic doors in turn as 0 9. Place the seven-segment cable display in a state of 0 and wire it according to Table 2.3: connect logic gate 0 to abcefg,connect logic door 1 to cf,and soon. Encountered



(a) (b)

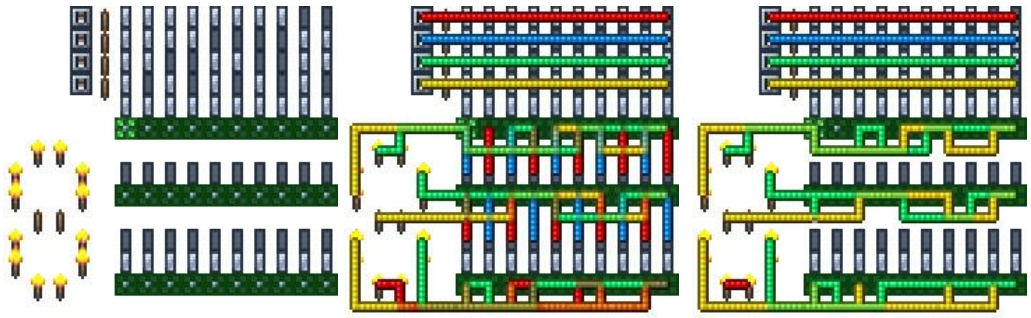
Figure **2.15**



(a) (b)

Figure **2.16:** Ten doors from left to right represent 0 9,and four switches / torches from top to bottom represent 8,4,2,1. For example, torches
When 4,2,1 is lit, the logical gate that represents the number 4 plus 2 plus 1 s7 is lit and the other logic gate is off.

Use a wire changer when the wire color conflicts, and the final circuit is connected as shown in Figure 2.17. By operating the four switches, the display shows the number 0 9 or goes out. This is because every time the number changes: the logical door of the original number goes out, activates once, so that all torches corresponding to the original number go out, and the logic door of the new number lights up, activates once, so that all torches corresponding to the new number light up.

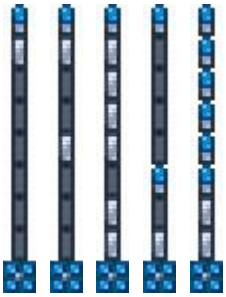


(a) **(b)** Yellow line green wire connection

Figure **2.17:** a connects to 02356789,b to 045689,c to 01234789,d to 2345689,e to 0268,f to 013456789,g to 0235689.

2.3 Fault logic gate

If there is at least one fault logic light on any normal logic gate, the logic gate turns blue, called the fault logic gate. A fault logic gate is a (not a class) special logic door, and the nature of the fault logic door is the same regardless of what the normal logic door below it is. The normal logic light between the fault logic gate and the fault logic lamp at the bottom is a valid logic light. When a fault logic light on the fault logic gate is activated, the fault logic gate is activated according to a certain probability, which is equal to the number of valid logic lights lit divided by the number of all valid logic lights. Ask the reader to carefully consider the description of this passage in comparison to Figure 2.18.



(a) (b) (c) (d) (c)

Figure **2.18:** When activating the fault logic light at the top, the probability of the fault logic gate activating is: (a)1/7;
(c) 1-

Only 1 valid logic light has a special fault logic gate. Because the probability of the logic light goes out is 0,thefault logic gate must not be activated, and when the logic light is lit, the corresponding probability is 1,and the fault logic gate must be activated. Circuit control is possible using a fault logic gate with only 1 valid logic light.

2.3.1 Wire changer

We've talked about wire changers at the normal logic gate. Simply put, a wire changer is a feature that takes advantage of the input required by some logic gate encounters. There are many variations of wire changers in practical applications.

(a) Single-light changer **(b)** dual-light changer

Figure **2.19:** Wire changer

There is only one single-light changer, which is to add a logic lamp to any ordinary logic door (Figure 2.19 (a)). Dual light changer Because there is a middle logic gate separated, input

and output can be used with the same color line. Double light changers can be made with doors, different or doors, and faulty doors, as shown in Figure 2.19 (b).

These three double-light changers may seem equivalent at first glance, but they are not. Because they are not equal in price, we can enrich our design ideas by using their differences in practical applications.

First look at the difference between a door and a fault door. When their topmost logic light is activated, the logic gate is activated. but

(a) Different from door vs fault door **(b)** or door anti-jamming

Figure **2.20:** (a) Right-click switch, red and blue line activated at the same time, left torch does not respond, right torch response;

Yes, if the topmost logic lamp is activated twice at the same time, the door will not be activated, and the fault door will be activated once, the corresponding experimental circuit is shown in Figure 2.20 (a). Readers can experiment on their own and then combine the characteristics of ordinary logic gates with fault logic gates to think about why this is the case. The same differences are made between different or faulty doors and faulty doors.

Then look at the difference between a door and a different or a door. No anti-jamming with the door, different or door anti-jamming. What do you mean? As shown in Figure 2.20 (b), there is a wire that is not related to the changer and wants to cross the wire changer. For different or doors, interference can be avoided as long as the method in the figure is used, and for doors, there is no way to avoid interference from the crossed line in any case. Readers can analyze the reasons for this on their own. Who, you might ask, would bother to get a thread through? In practice, this is sometimes necessary, and then the difference or door provides convenience.

2.3.2 Assignment circuit

The logic circuit in Tyralia has a very big drawback, which is the difficulty of assigning. In a number of electricity, you want to assign a circuit a value of 0 or 1simply by connecting the power supply to the corresponding level. In Tyralia, however, the wire has no level, only activation, and activation can only change the state of the logical light. Assigning a logical light to 0requires activation (or odd number of activations) in the case where the logical lamp would have been 1, and inactivation (or even several activations) if the logical lamp would have been 0. The circuit shown in Figure 2.21 does this: when the logic light goes out, activating the red line activates the fault logic light because the logic light goes out, the logic gate does not activate, and the logic light stays out. When the logic light is on, activating the red line

activates the fault logic light because the logic light is on, the logic gate activates the blue line, and the logic light goes out. In any case, activating the red line turns off the logic light.

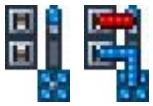


Figure **2.21**

If you synchronize the logic light with a torch, you can activate the red line to turn it out. If you reverse sync the logic light with the torch, you can activate the red line to light the torch (Figure 2.22).

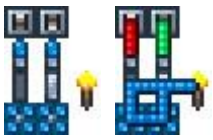


Figure **2.22:** The left switch lights the torch (assignment 1)andthe right switch turns the torch off (assignment 0).

Example **2.9D** trigger

Figure **2.23:** The switch on the left changes the state of the A torch, and the switch on the right updates the status of the A torch to the B torch.

D triggers are the term for power generation and have a simple function of storing and sending states. See Figure 2.23,the switch on the left can control the torch on the left at will, while the switch on the right synchronizes the torch on the right with the torch on the left. In other words, theD trigger stores the value of the torch on the left, and the switch command D trigger on the right emits the value it stores.

Looking back at the assignment circuit, it can set the value of the torch to a constant, which can be adjusted by an effective logic lamp on the fault logic gate. In Figure 2.23, we ask the torch on the left to adjust the effective logic light on the fault logic gate, which acts as an assignment 1 when the torch on the left is 1, and 0 on the left, and 0on theleft. In this way, the function of this fault logic gate is actually to assign the value of the torch on the right to the value of the torch on the left.

2.3.3 Pass circuit

A relay circuit is a circuit that uses a very high frequency. The traditional relay circuit is shown in Figure 2.24. When the green line is activated, a row of fault logic lights is activated. However, since only the first valid logic light is on, only the first faulty logic light is activated, so that the first valid logic light goes out and the second valid logic light is lit. When the green line is activated again, the second fault logic gate is activated, the second valid logic light goes out, and the third valid logic light is lit. In this way, when the green line is activated repeatedly, the six fault logic gates are activated and cycled in turn. By connecting each fault logic gate to a circuit, six circuits can be run and cycled in turn.

Figure **2.24**

The relay circuit in practice is very flexible. Not only the number of fault logic doors can change at will, the effective logic light light on and off, wiring mode can be changed according to the actual needs. Common variants of the pass-through circuit are shown in section 6.1. Use the relay circuit do not be rigid, be good at designing the most reasonable circuit for the needs.

The use of pass-through circuits is very widespread. It can be combined with decimal number display to form a decimal counter inhttps://www.bilibili.c om/video/av22894193, or it can make a https://www.bilibili.com/vid
Neon effects in eo/av21009075 can also be https://www.bilibili.com/video/av639
Blood back in 3957 back to the blue array, and so on and so on. Another typical use of the pass-through circuit is to reduce the frequency. We know that the conveyor belt drives at a frequency of 60Hz. Now we need a 20Hz drive, using only a pass circuit with a cycle length of 3 (Figure 2.25).

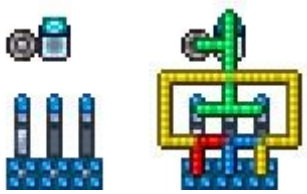


Figure **2.25:** Connecting the yellow line gives you a 20Hz drive because the green line activates 1 time for every 3 activations.

At some point, such as the decimal counter mentioned above, we need a "zero" operation to reset the pass-through circuit to its initial state. This is easy to achieve with the

assignment circuit (Figure 2.26). But what you want to make clear here is that understanding assigned objects needs to be very flexible.

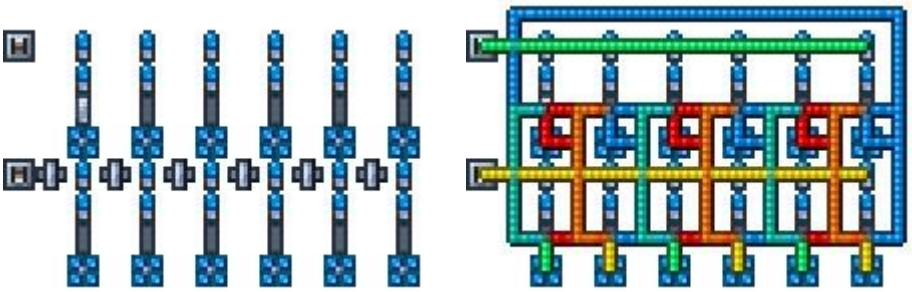


Figure **2.26:** The switch above is used to activate the relay circuit normally, and the switch below is used to reset. In addition to crossing the green and yellow lines of all fault logic lights, red and blue lines are used to cycle through the relay circuit and synchronize the status of the valid logic lights above below, and the green and yellow lines are used to assign values. Add an extra row of fault logic lights to separate the lines that conflict with the color.

Is it only logical lights that have a state? In the relay circuit, the wire can also have a state. If we attach each wire to a torch, we can think of the state of the torch as the state of the wire. Instead of assigning values to logical lights, the circuit is assigned directly to the wire, as shown in Figure 2.27. The wire is assigned on the premise that the wire state determines the state of the logical light.

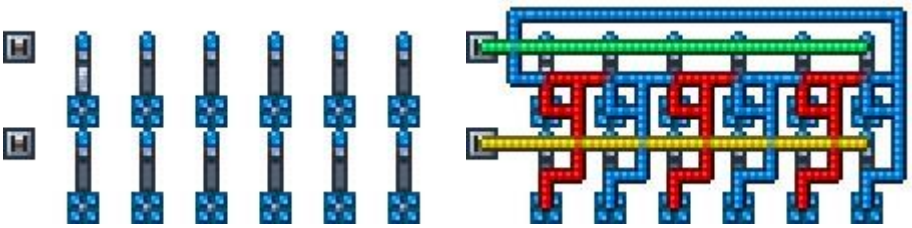


Figure **2.27:** The switch above is used to activate the relay circuit normally, and the switch below is used to reset .

Insert the following row of reset circuits in Figure 2.27 into the gaps in the overreach circuit to get a smaller footprint (Figure 2.28).

In addition, occasionally we may need to use a two-way recircuit, which can be found in section 6.1.2.

Figure **2.28:** The switch above is used to activate the relay circuit normally, and the switch below is used to reset .

2.3.4 Down-frequency circuit

We've talked about how to use the pass-through circuit to reduce the frequency. For example, in Figure 2.29 (a), the green line activates the red line when it activates odd times, and even when the blue line activates several times. In fact, there is an easier way to reduce the frequency by half. In Figure 2.29(b), activating the green line activates the fault logic light and lights up the valid logic light at the same time, when the fault logic gate is activated and the red line is activated, and reactivating the green line activates the fault logic light and turns off the valid logic light at the same time, at which point the fault logic gate is not activated. Figure 2.29(b) can also be activated when the odd number of times the green line is activated, and Figure 2.29 (c) can also be activated when the green line is activated evenly several times. Generally speaking, the down-frequency circuit refers to this circuit that uses a fault logic gate to reduce the frequency by half. When precise frequencies are generally not required, the use of down-frequency circuits is more space-saving than the relay circuits.

(a) (b) (c) 图 2.29

2.3.5 Decimal Counter

In this section we will use the modules we learned earlier to make a four-digit decimal counter with a number display.

The object of the counter count is the driver or a specific signal source. For each activation of the signal source, the number of the counter is added to 1. Because it is decimal, it is full of decimals, which prompts us to use the pass circuit with a loop of 10 as the counting module. Because the counter has to have a zeroing function, the relay circuit should be brought with it a reset circuit. According to the writing habits of numbers, the right is low and the left is high, so the reverse pass circuit is used and the logic gate is slightly misaligned to make the wiring smoother. The input of the last logical gate of the low-bit pass-through circuit is connected to the input of the higher pass circuit to complete the feed function (Figure 2.30), and the lowest-bit pass-through circuit input is then connected to the driven output to achieve the counting function.

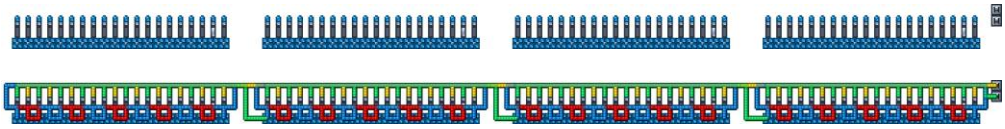
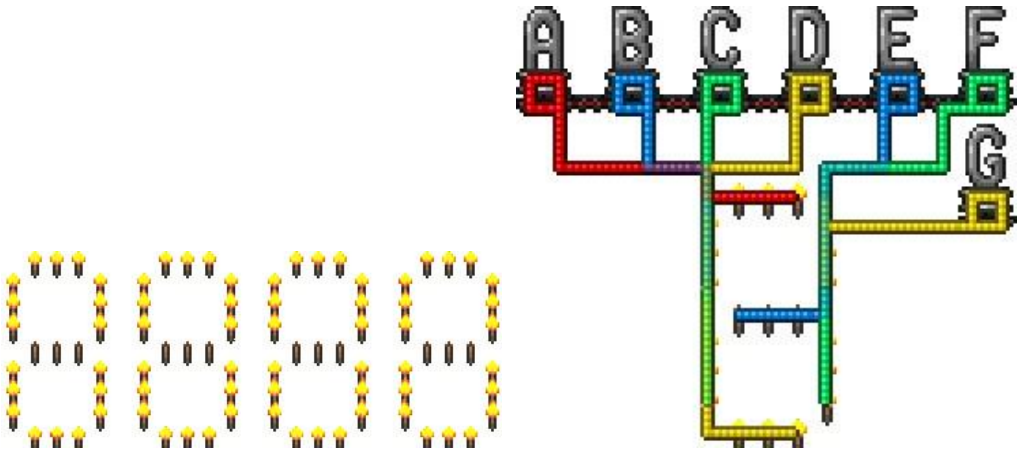


Figure **2.30:** Green line count, yellow line reset. The far right is a bit, and the far left is a thousand bits.

People familiar with the pass circuit can already read the numbers through the logic light on the relay circuit: the valid logic light on the far right of the pass circuit means that this bit is 0,and the valid logic light on the far left indicates that this bit is 9. But since you want to target users who don't understand the circuit, you need to visualize the numbers, that is, add the numbers. We've done a decimal number display before, and if we copy the display portion of that number, it's easy to see that the display isn't right because the inputs for the two numbers are different. The previous number shows that the display receives two signals when the number changes: the first signal turns off the previous number, and the second signal lights up the new number. And the output of our delivery circuit is only the previous number. Because the previous number in the counter can only determine the new number, we can directly activate the signal from the old number to the new number that needs to be activated. In addition, since the display is a number of numbers arranged together, in order to avoid wiring difficulties, the number of the line should be determined, and then connected to the relay circuit. If you select the torch arrangement shown in Figure 2.31 (a), the seven-segment wiring method used previously cannot be used because only one grid is empty between the numbers. So we use another seven-segment join, Figure 2.31 (b), to note that some lines overlap. The wiring can be done with Table 2.4 (Figure 2.32). Notice that a faulty logic gate is used here to change the wire.



(a) **(b)** 图 **2.31**

Digital changes	0→1	1→2	2→3	3→4	4→5	5→6	6→7	7→8	8→9	9→0
Activate the section	to	abdef	bcfg	abcd	acdg	bc	of	you	bc	bceg

Table **2.4**

2.3.6 Downflock technology

Flexible use of pass-through and down-frequency circuits increases the existing drive duration to any integer multiple (Figure 2.33).

Using the above method, the use of a pass-through circuit is too large when a large prime multiple (e.g. 23x) time is required, at which point multiple downfrequency drives can be combined flexibly using the control function of the fault logic gate (Figure 2.34). This is essentially a multi-level recurring.

2.3.7 Dice

In the previous example, we used only the control function of a fault logic gate with a valid logic light. In this example, let's use the probability function of the fault logic gate. Our goal is to make a circuit that has a switch and six torches. When the switch is activated, the six outputs have and only one is lit, and the probability of each torch lighting is 1/6.

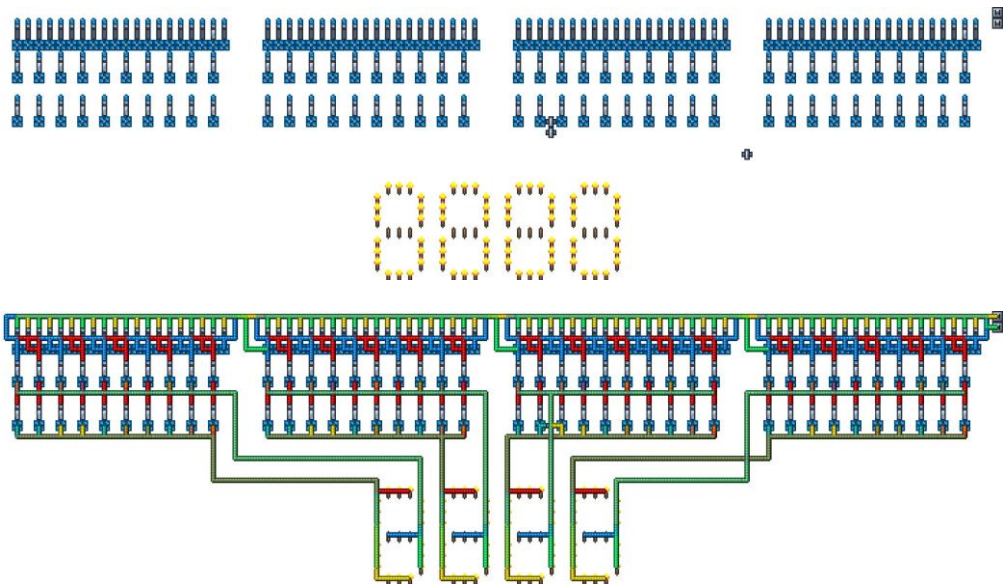
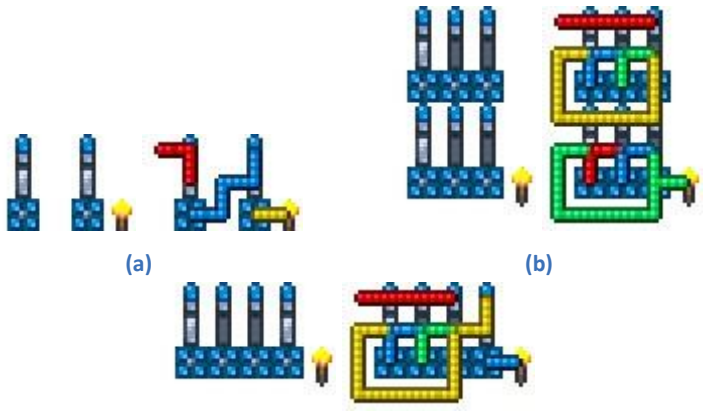


Figure **2.32**



(a) (b)

(c)

Figure **2.33:** (a) Two down-frequency circuit connections, with the red line responding once every 4 torches activated; (b) two relay circuit connections, red lines responding once every 9 torches activated; (c) the down-frequency circuit is connected to the relay circuit and the red line responds to each of the six torches activated.

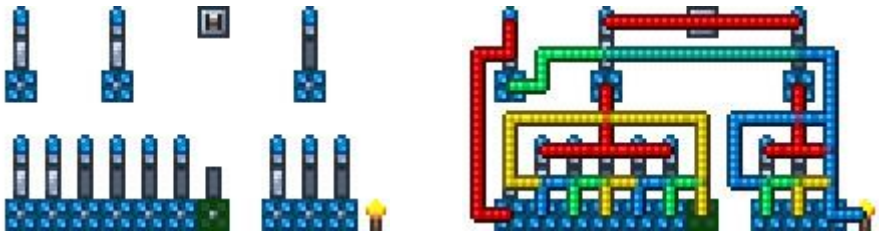


Figure **2.34:** Switch responds 23 times per activated torch. The two fault logic gates on the right above are used for control, the output on the left is connected to a module with a count of 20 (5- the pass circuit is connected to two down-frequency circuits), and the output on the right is connected to a 3-pass circuit. When the switch is activated, the left output is initially left, the right side is not output, and the left count is counted. When the left count to 20, the green line above is activated, changing the effective logic light for control to the right output, the left side not output, and the right count. Activate the torch when the right counts to 3 and change the effective logic light for control back.

The circuit is shown in Figure 2.35. Activate the top yellow line to run. The following is obviously the relay circuit, let's analyze the above section first to see how many times the green line will be activated.

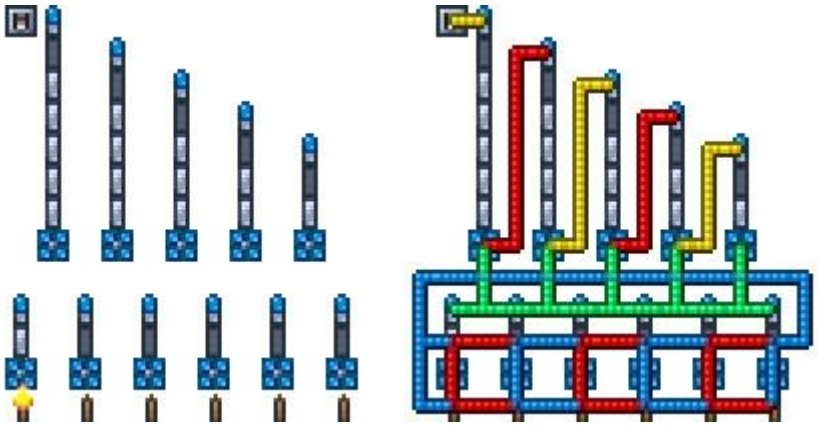


Figure **2.35**

When the top yellow line is activated, the first fault logic gate has a 5/6 probability of activation. Only if the first fault logic gate is activated is the subsequent fault logic gate, that is, the probability of 1-5/6 x 1/6 is not activated once. With the first fault logic gate activated, the second fault logic gate has a 4/5 probability of activation. Only if the second fault logic gate is activated, the subsequent fault logic gate is likely to be activated, that is, there is a 5/6 x (14/5)x1/6 probability that only the first fault logic gate is activated, at which point the green line is activated once. And so on, you can get the green line has 5/6 x 4/5 x (1-3/4) x 1/6 probability activation twice, there is 5/6 x 4/5 x 3/4 x (1-2/3) x 1/6 probability activation three times, there are 5 The probability of activation four times is 5/6 x 4/5 x 3/4 x 2/3 x (1-1/2) x

1/6, and the probability of activation five times is 5/6 x 4/5 x 3/4 x 2/3 x 1/2 x 1/6. That is, when the top yellow line is activated, the green line and so on may be activated 0 to 5 times.

The Green Line, etc., can be activated 0 to 5 times, and passing through the following pass circuit will cause the lit torch to circulate 0 to 5 times to the right. Regardless of which torch is on before the top yellow line is activated, this means that there is an equal probability that the six torches will be lit.

K Thinking question k

1. Why can only player-activated pressure plates choose gray/brown/blue/jungle lizards?
2. Explain why the device shown in Figure 2.36 activates the broadcast box instantaneously when the world turns on. What does the fault logic gate in the unit do?
3. Make a device that displays the message "The eclipse is happening ... every day into the daytime . . ", " The blood moon is rising at night ... . ”。
4. Find the delay device inhttps://www.bilibili.com/video/av5271577.
5. Make a display that shows the current moon phase, as shown in Figure 2.37.
6. Achieves the effect of temple spikes in 2 minutes and 57 seconds https://www.bilibili.com/video/av5271577.
7. The design is illustrated by the binary number shown in Figure 2.38.
8. Complement the decimal number to show the wiring of Figure 2.39.
9. Why does the decimal band number display counter generally not be more than four bits?

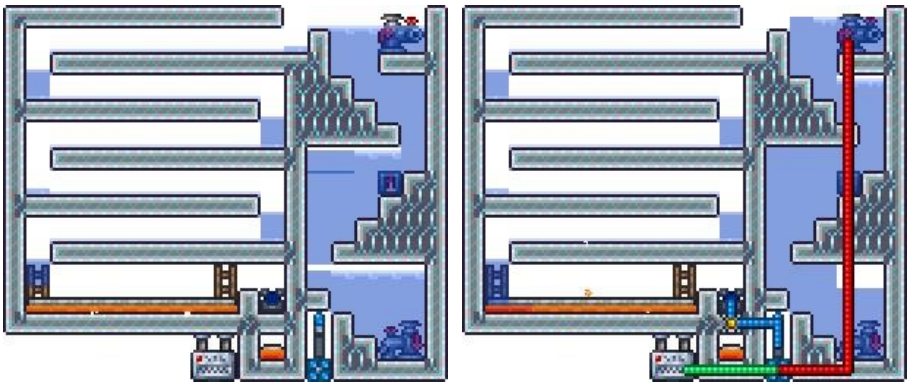
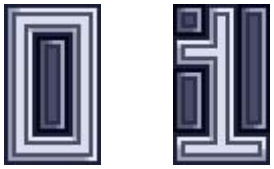


Figure **2.36**

(a) Full moon **(b)** loss-bumping moon **(c)** lower string moon **(d)** residual moon **(e)** new moon **(f)** Emei moon **(g)** upper string moon **(h)** bump moon

Figure **2.37**



(a) 0**(b)** 1 图 **2.38**

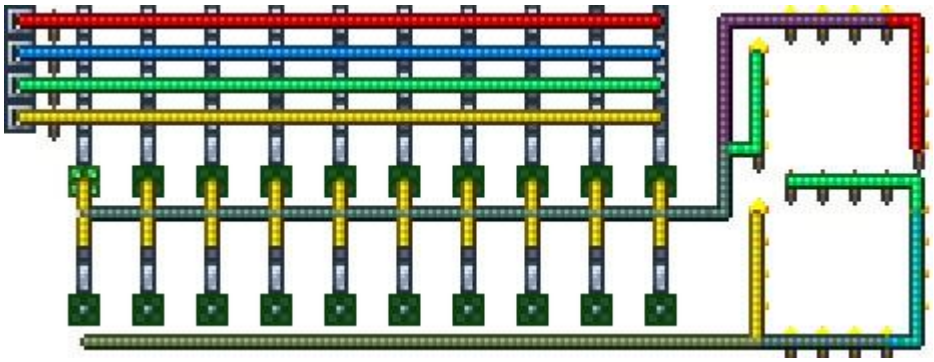


Figure **2.39**

10. Make a multi-part binary counter using the down-frequency circuit and binary number display.

11. Make a dice that shows points, and the six-point effect is shown in Figure 2.40.

(a) 1 **(b)** 2 **(c)** 3 **(d)** 4 **(e)** 5 **(f)** 6

Figure **2.40**

Chapter 3 logical settlement

The

- MechScope Modu
- Logical
- Burst
- Logical
- Logical delay

ex 3.1

In the previous chapter we covered many circuit modules. At this time, if the reader takes these circuit modules together, it is easy to have some unexpected problems, which is why the previous chapter did not introduce too many composite circuits. In this chapter, we will refine all the details of the entire logical settlement to ensure that the behavior of any composite logic circuit can be predicted.

In a logic circuit, each power supply and electrical appliance can be connected to multiple wires, each wire can be connected to multiple power supplies and electrical appliances. When the circuit is activated, all power supplies, wires, and electrical appliances are activated in a certain order.

It is very clear that when the power supply is activated, the wires on the power supply are activated, and then the electrical appliances under the wires are activated. Further, if power *A* is activated before power *B,* the wires on power *supply A* should also be activated before the wires on B, if wire *a* is activated first than wire *b,* then all appliances under *a* should be activated before all appliances under b, and if the logic light on logic gate *A* is activated before the logic light on logic gate B, then the logic gate *A* should also be activated before logical gate *B.*

Some of the above rules are reasonable and easy to understand. The next rule is unique to Tyralia and requires memory.

When a power supply is activated, the four-color wires on it are activated in the order of red, blue, green, and yellow.

When a wire is activated, its upper points are activated in the order from near to far from the power supply. To points where the power supply distance is equal, understanding the activation order requires understanding the graph's Breadth First Search(BFS)algorithm, which is not explained in detail here.

For circuits involving logic gates, circuit settlement is done step by step. Logic gate activation is performed at each step -- wire activation -- logic light activation -- and logic gate judgment is these four small steps. After the logic lamp is activated, the logical judgment is not made for the time being, and then the logical judgment is unified until all the wires have been

settled. Each step is called a logical frame. During the settlement of a logic circuit, four small steps in each logical frame are run repeatedly until no logic gate needs to be activated.

A power source other than a logic gate is called a physical power supply. A logical settlement must be initiated by the activation of the physical power supply.

A logical gate in a logical settlement can only be activated at most once. If it tries to activate a second time, then activation fails and white smoke is emitted from the logic door, called the burst door. The logic gate state still switches normally but does not signal. The door is a normal mechanism, is the developer intentionally for it, to use can also play a role.

3.1 Study the circuit settlement process with **the MechScope** module

The MechScope module visualizes the circuit settlement process. The download link is already given in Section 1.1. After the module is installed, the settlement process of the circuit can be seen in the game.

Figure **3.1:** MechScope module

Once in the game, turn on settings - Control, and you can see the key settings for MechScope. The default key is: Keypad 1 Toggle(Trigger) Keypad 2 Step Keypad 3 Auto StepKeypad 5 Settings

As shown in Figure 3.2 (a), make the simplest circuit first. In this circuit, the switch controls the torch through a red line.

(a) **(b)**

Figure **3.2**

Press keypad 1to see a yellow dot in the upper right corner of the cursor (Figure 3.3 (a)) indicating that MechScope is running.

(a) **(b)**

Figure **3.3**

Right-click on the switch and the circuit is paused. A red box appears on the switch, with a red filter in place to indicate the active power supply and wires (Figure 3.2 (b)). The torch has been extinguished. The yellow dot in the upper right corner of the cursor becomes a red dot (Figure 3.3 (b)), indicating that there is a running circuit. Press Keypad 2 to step, circuit settlement is complete, and the upper right corner of the cursor restores the yellow dot.

With the basics in place, let's look at the Settings menu. Press keypad 5to open the settings menu (Figure 3.4). The first four are single-select, representing the magnitude of each step. Single is step-by-step,Wire is line-by-line,Source is source-by-source,and Stage is logical frame-by-frame. The last six are display settings that affect only the degree of visualization, and are not intersched

Figure **3.4:** MechScope's settings menu

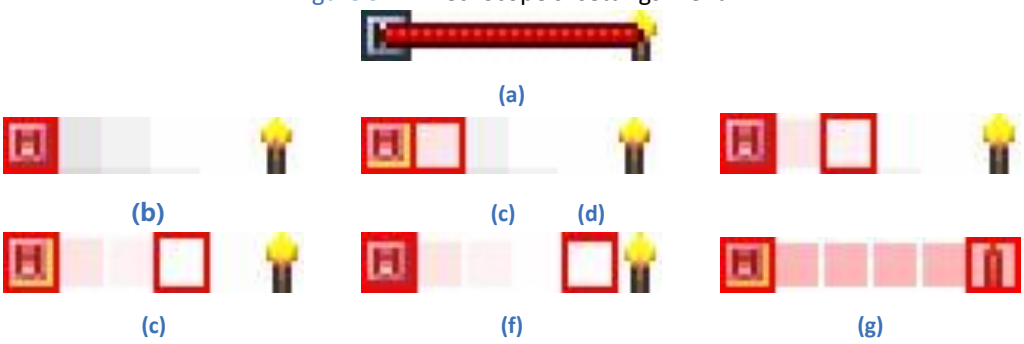


Figure **3.5**

Shao, readers can try it themselves. The last number is the period/physical frame of the auto-step, and the smaller the value, the faster the auto-step.

Our focus is on the first four settings, which are about circuit settlement, where Stage is about logical settlement.

3.1.1 Step by Step

Select step-by-step in the settings menu, and then make the circuit shown in Figure 3.5 (a). Turn on MechScopeand right-click on the switch to see a red box on the switch that indicates that the switch acts as a power source and activates the red line above it (Figure 3.5(b)). Press keypad 2 step, and a red box appears on the right side of the switch, indicating that the grid is active (Figure 3.5 (c). Step again and again, and the red box moves to the right until the torch is on and the torch is closed.

This experiment shows the sequence of activation on a single wire: the wire is activated from near to far. But it shows only one-way propagation. For complex situations, the reader can study themselves using the MechScope module. In general, we only need to know the first activation near the power supply, the distance from the power supply after activation, the distance here refers to the length along the wire to the shortest path to the power supply. Here is a message for computer majors: a single wire is followed from the power supply

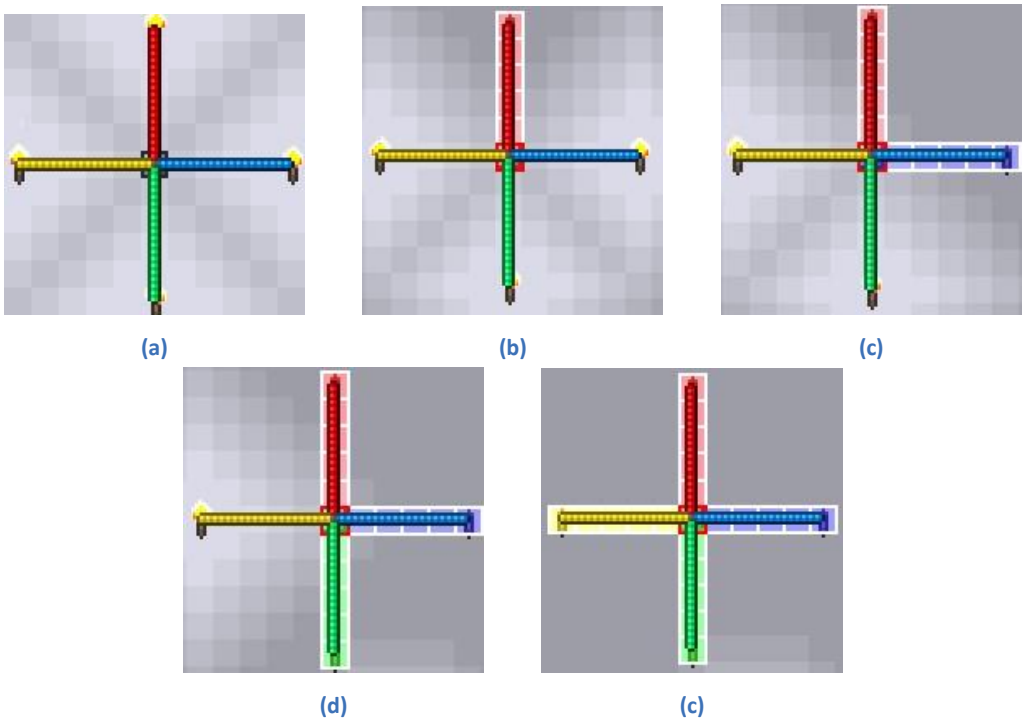


Figure 3.6

The order in which the breadth-first search begins is activated, and the order of the four directions is upper right and left.

3.1.2 Step-by-line

Select step-by-line in the settings menu, and then make the circuit shown in Figure 3.6 (a). In line-by-line step mode, each step advances an entire wire. Right-click on the switch to

see the red line activation (Figure 3.6 (b), step- and blue-line activation (Figure 3.6 (c), step-by-step, green-line activation (Figure 3.6(d),stepping, yellow-line activation (Figure 3.6 (e).

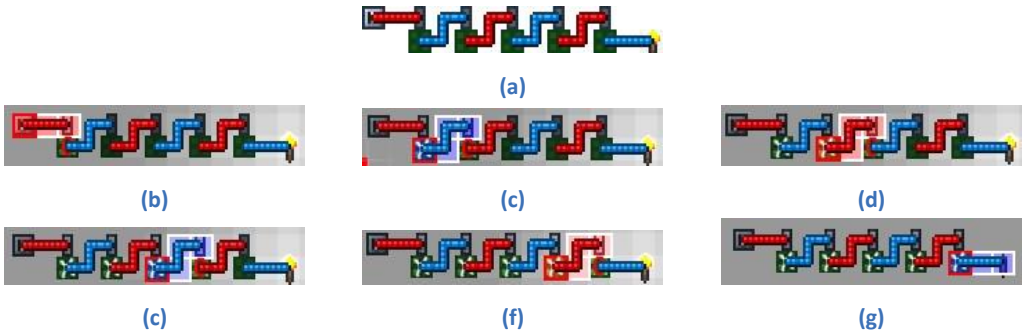
This experiment showed the activation sequence of different wires on a single power supply: red, blue, green, and yellow.

3.1.3 Step by logical frame

Let's skip the source step first and see the logical frame step by step. The circuit is shown in Figure 3.7 (a). In logical frame-by-frame step mode, each step advances one logical frame.

We generally record the logical frame when the physical power supply is activated as the 0th logical frame (Figure 3.7 (b),each logical frame is followed by logical gate activation, wire activation, logic light activation, logic gate judgment. In Figure 3.7, from (b) to (g) the 0th to the 5th logical frames, each with a bold box, the filter marks the activated wire, "Œ"marksthe activated logical door,and"" marks the logical door to be activated in the next logical frame.

Take another look at a slightly more complex circuit (Figure 3.8 (a). Right-click on the switch to enter the 0th logical frame (Figure 3.8(b): switch as power, red line on switch activated, two logic lights activated, two logic gates to determine, and plan to activate (a small red circle is drawn on the logic door).

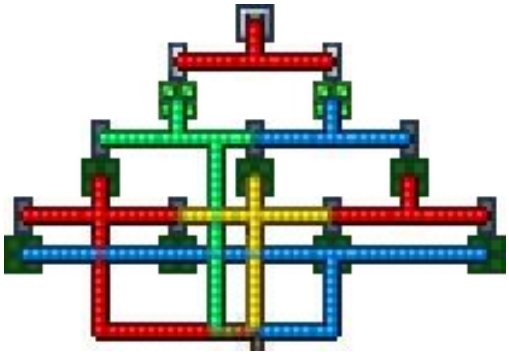


(a)

(b) (c) (d)

(c) (f) (g)

Figure **3.7**



(a)

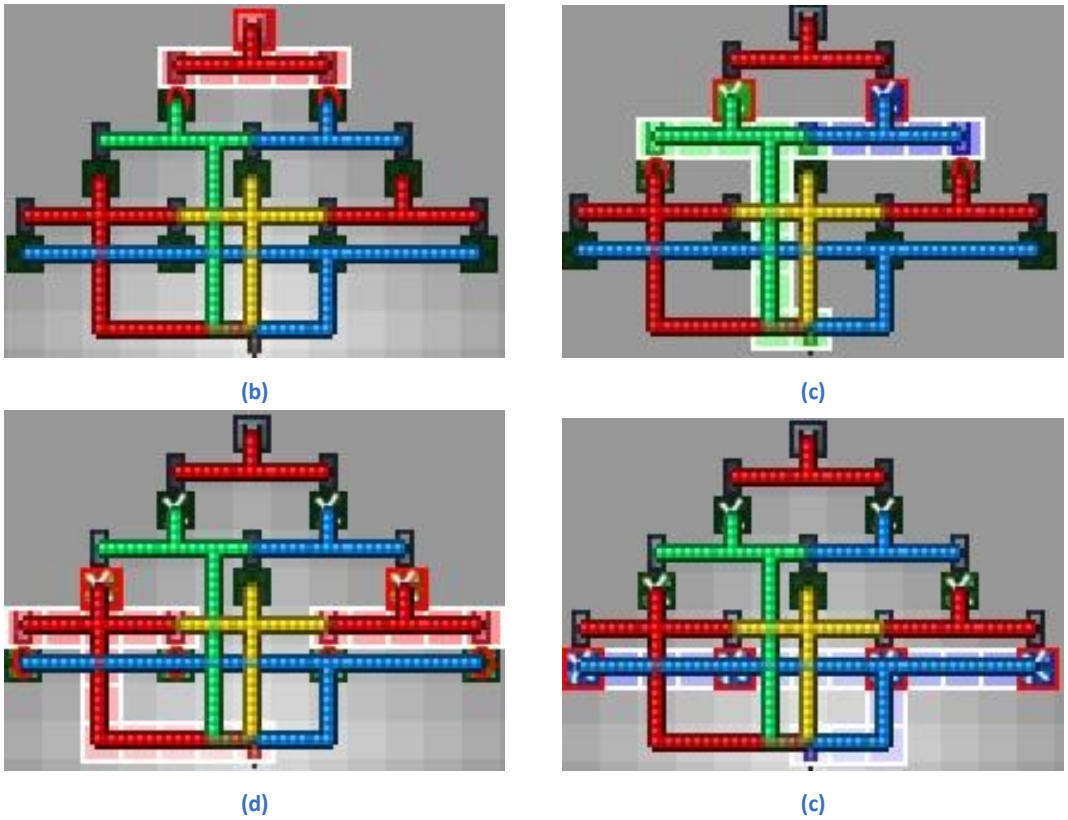


Figure 3.8

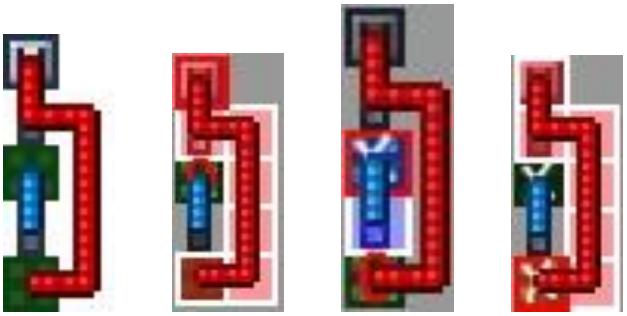


Figure 3.9

The first logical frame (Figure 3.8 (c). Two logic gates are activated as power, blue and green, and three logic lights and torches are activated. Where the torch changes the state, the left and right two logic lights change the state, the middle logic lamp because it is activated twice, the state remains unchanged. The logic gate under the three logic lights is judged, where the left and right logic gates are scheduled to be activated and the logic gate in the middle is not scheduled to be activated.

The 2nd logical frame (Figure 3.8 (d). Two logic gates are activated as power, two red lines are activated, and four logic lights and torches are activated, all changing the state. The four logical door plans to activate .

The fourth logical frame (Figure 3.8 (e). Four logic gates are activated as power, the blue line is activated four times, the torch is activated four times, and the status remains the same. There is no logical gate scheduled to be activated and the logical settlement ends.

3.1.4 Blast door

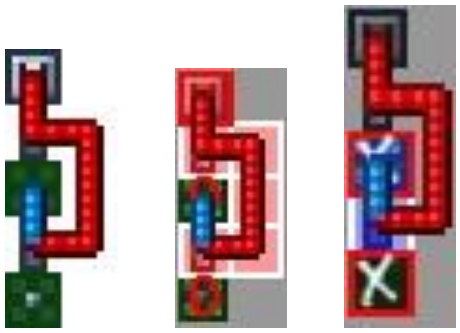
"In Figure 3.9(a), this is a loop circuit that looks like the red line is activated at the 0th logical frame, the blue line is activated at the 1st logical frame, the red line is activated in the 2nd logical frame, and the blue line is activated in the third logical frame ... If that's the case, the game will crash because in Tyraya, no matter how complex the circuit is, it has to be settled within a physical frame. If 1/60 seconds cannot be settled, the physical frames are lengthened accordingly, resulting in a decrease in the frame rate. If it never settles, the game gets stuck in a physical frame and can't do anything.

Tyralia uses a fool's way to avoid this dead cycle. When a logic gate is activated, there is a marker that displays a white "Œ"in MechScope. If the logical gate is then scheduled to activate(MechScope displays the red ""), cancel the plan and play a burst door animation (smoke). In Figure 3.9, the burst door behaves in Figure 3.9 (d), when the logical door above is scheduled to activate again. The moment a screenshot fails to capture the door, the reader can try it on his own.

It is called the "fool's method" because it has a by-product. In other words, a burst door can also occur for circuits that do not cycle dead. In Figure 3.10, the logic gate below is activated at the first logical frame and marked "Œ". In the first logical frame, the blue line is activated and the logical light state below changes, so the logic gate is scheduled to activate again, resulting in the burst door.

From the above two examples, let's summarize the reasons for the explosion of the door. A logic gate is activated at most once in a circuit settlement. If you try to activate again, activation fails and the door bursts. It is important to note that the logical door state will still change as usual.

In practice, burst doors often cause circuit bugsbecause we often want logic doors to be activated multiple times when the state switch is multiple, but not in practice. Learned to predict the door, you can find a way to avoid the door, or even



(a) (b) (c)

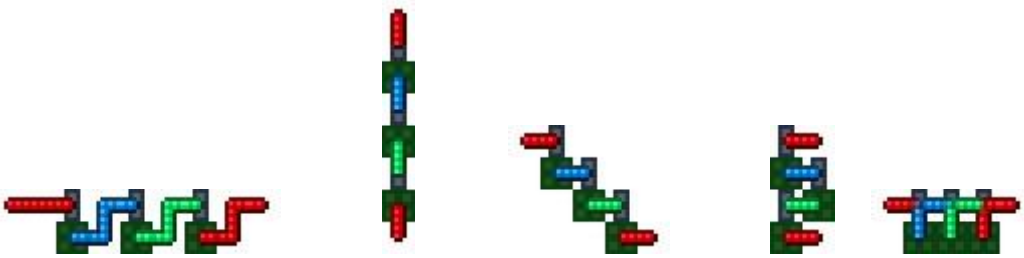
Figure **3.10**

Use a burst door.

3.2 Logical delayer

When connecting multiple logic circuit modules, you need to control the order in which they operate in order for them to work together correctly. The number of logical frames experienced by different circuit modules from input to output is different, so a simple wiring method may cause some modules to work when they should not be working, disrupting the circuit state. Using a logic delayer, you can postpone the running logic frame of a circuit module so that it does not run until it needs to be run.

The line changer mentioned in the previous chapter can act as a logical delayer because the output activates one logical frame later than the input activates. If you connect multiple transwires at the beginning and end, you can control the number of logical frames that are delayed. A common pattern of logical delayers is shown in Figure 3.11.



(a) Horizontal **(b)** vertical **(c)** oblique **(d)** double vertical **(e)** compact horizontal, utilizing burst doors Figure **3.11:** Different swings of the logic delayer, with a logical delay of 3 logical frames.

3.3 Network cable

When designing large circuits, complex signals need to be transmitted between the two places, such as setting up multiple sensors in A, processing and responding in B. In general, each signal needs to be connected to a wire, which can take up a lot of space when the signal is more complex. This is because the signal complexity of a wire is too low: a wire can only choose to activate or not activate two states, so only one binary bit can be transmitted per wire.

3.4 Logical synchronization of ordinary logic gates

Now that we understand the logical settlement mechanism, we can use one wire to pass multiple binary bits. This feature is similar to that of a real-life network cable, so we call it a network cable. Because logic gates are sensitive to logical frames, we can send multiple binary bits over a single wire at different logical frames and parse this information at the receiving end. At the end of the day, we need to do an encoder and a decoder. Encoders are used to translate simple signals on multiple lines into complex signals on one line, and decoders are used to translate complex signals on one line into simple signals on multiple lines.

The design of the encoder and decoder depends on the signal characteristics, so there is no fixed practice here, only a simple example for reference. We place 8 torches in A, using switches to change their status, 8 torches in B,and only 1 wire between ground A and B. We hope that after setting the torch on ground A, we can trigger a switch to update the torch state of ground A to ground B, i.e. pass eight binary bits using a line. In the sender circuit, the reader needs to learn how to use a logic delayer to send signals at a particular logical frame, and in the receiving circuit, the reader needs to learn how to use a faulty logic gate to determine whether a line is activated at a particular logical frame.

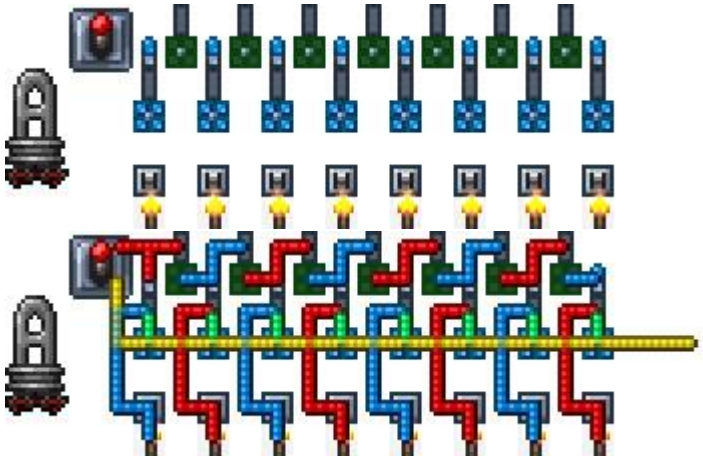


Figure **3.12:** Eight switches control the status of 8 torches, the lever is used to send a signal, and the yellow line is used to output. The 0th logical frame yellow line is activated to make a start signal, followed by the logic delayer above, andthen8 D latches are activated in turn at 1 x 8 logical frames.

In A, we need to trigger the corresponding D trigger in turn in 8 logical frames to send the signal out

(Figure 3.12).

In B, we need to extract the inputs from each logical frame and output them to 8 torches (Figure 3.13).

Although many logic gates are used in the above devices, it is better to use them when AB is far away than to connect ABs with 8 wires.

3.4 Logical synchronization of ordinary logic gates

For players with a number of electrical bases, there are a variety of burst doors when it comes to the Tyralia circuit. And through a variety of teaching videos to get started with the "outsider" is not easy to encounter such a problem.

The reason for the burst is that a logical door is activated multiple times in a logical settlement. Although the faulty logic door can also burst the door (as will be said in the next section), but the ordinary logic door is the worst-hit area of the burst door. When we use ordinary logic doors, we usually have to do simple combination logic operations, such as a few with logic, a few different or logical combination. This

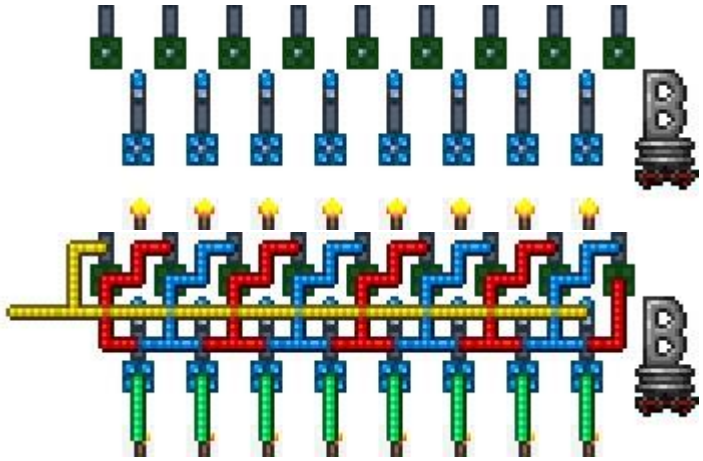
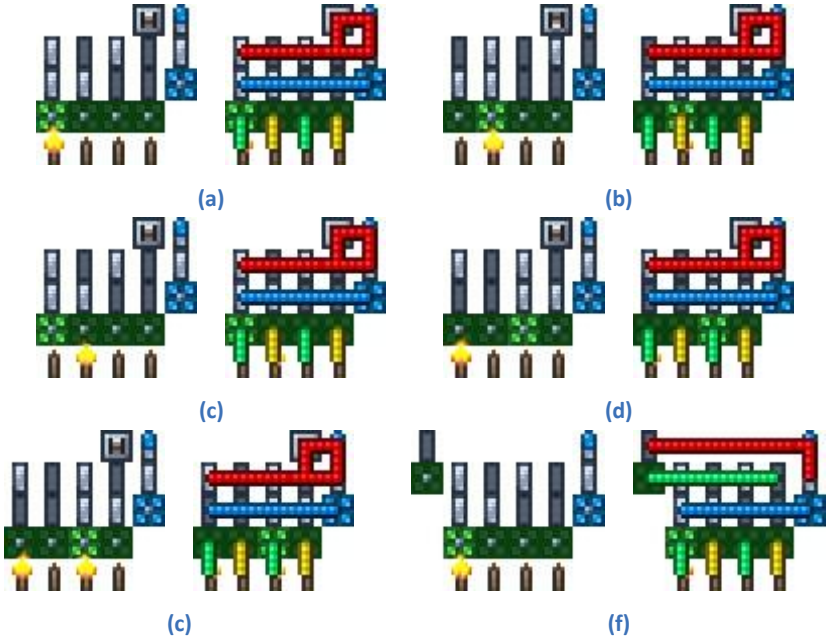


Figure **3.13:** Start working when the yellow line (in the 0th logical frame) is activated. The logic delayer above lights up the corresponding valid logic light below in 1 8 logical frames in turn, and if the yellow line is activated when the corresponding logical frame is activated, the corresponding torch below is activated.
The 9th logical frame resets.

The logical operation of the kind is, of course, a state judgment without the need to use an active judgment. The premise that the state judgment can be made is that the state of all power supplies and electrical appliances connected by a wire should be synchronized, in other words, when the power state changes, and all electrical appliances connected to the power supply also need to change. But when the burst door occurs, the logic door state changes

without signaling, which will cause the electrical state to remain unchanged, the circuit can not make a state judgment. Circuits that cannot make state judgments cannot do combination logic.

Take a simple example. We know that the down-frequency circuit (section 2.3.4)can do binary counting, while section 2. The circuit in 2.3 converts the binary encoding into a decimal display. Then it is natural to think that the two circuits can not be connected to do decimal counting?



Only lite circuits are given here to show that this is not the right thing to do. As shown in Figure 3.14 (a), with only one down-frequency circuit, four numbers can be counted, decoded with four double lights and doors, and output to four torches.

3.5 Status representation and activation representation

The switch is activated for the first time, only the red line is activated, and the status of each logic light and logic gate is as shown in Figure 3.14 (b),so the first door and the second door are activated, the first torch is extinguished, and the second torch is lit. The switch is activated a second time, both the red and blue lines are activated, but the red line is one logical frame earlier than the blue line. After the red line is activated, the logical lights and logic gate states as shown in Figure 3.14(c), at which point the first and second doors are activated, the first torch is lit and the second torch is extinguished. After the blue line is activated, the logical lights and logic gate states are shown in Figure 3.14(d), at which point the first door and the third door are activated, but the first door is activated, so the door bursts, only the third door is activated, so the third torch is lit (Figure 3.14 (e)). As can be seen here, although the state of the logic gate is correct, that is, only the third logical door is bright, but the torch state is not right, because the first torch after the first door burst the door less than one activation.

The reason for this burst is that the red line is one logical frame earlier than the blue line, and the solution is simple: make a logical delay on the red line and activate the red and blue lines at the same logical frame (Figure 3.14 (f)).

3.5 Status representation and activation representation

There are two representations of signals in the Tyralia circuit: state representation and activation representation. Status representation is generally explicit, i.e. the naked eye can read the information it wants to represent directly from the state of the circuit element, such as a bright 1and an out of 0. Activation means implicit, activation means 1,and inactivation means 0. Because activation can only change the status of any item with a display effect (except the pixel box), to read the active information, you need to compare the state before and after the activation of the circuit element, the state change is 1and the state remains unchanged is 0. This kind of reading is obviously inconvenient.

From the point of view of circuit operation, status representation and activation representation often need to be used together. Normal logic gates generally apply to state representations, while fault logic gates are suitable for activation representations. Whether to produce readability output or to transmit information between circuit modules, the interchange between state representation and activation representation is used. This section describes how to interchange state representation with activation representation.

The 3.5.1 status indicates that the status is converted to active

A fault logic gate can be used to turn light into activation and off into inactive functions. It is important to note that with only one fault logic gate, even if the valid logic light switches between lights up and down multiple times in a logical settlement

Logic doors can only be activated once at most. So more subdivision, a little bright immediately activated (Figure 3.14 (b)) and activated afterwards

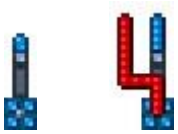
(Figure 3.14 (c)) two methods of creation.

Figure **3.14:** The red line controls the valid logic light state. (b) Activate the valid logic light as soon as it is lit;(c) activate it after the fact, and activate the blue line at some point to convert the light/off of the valid logic light to the activation/inactivity of the fault logic gate.

Lighting instant activation uses a de-frequency circuit, which is not explained in detail here. After-the-fact activation is activation of state 1 at some point (logical frame) and inactivation with a state of 0.

3.5.2 Activation means transition to status

Activate the turn state, you need to turn the activation on, inactive into off. The circuit is designed with special care for the response when "inactive", because according to circuit principles, there is no electrical response when not activated. If you want to respond to Inactivate, you must have the response when the circuit is inactive at some point, as shown in Figure 3.15.



(a) (b)

Figure **3.15:** Whether or not the red line is activated, the active blue line first places the valid logic light at 0,and then the active logic lightbecomes
1, otherwise the valid logic light is 0.

3.5.3 The decimal number that can be displayed randomly is displayed

In the previous chapter we did decimal numbers that weren't free enough: either binary or continuous numbers were needed. A decimal display in a perfect sense should receive 11 inputs, the first ten input activations show a number of 0 to 9, and the last input activation is all off.

The number that shows 0 9 corresponds to the torch status, and the input is active. A circuit is therefore required to convert activation into a state. As shown in Figure 3.16,add apre-0 circuit to the torch. Regardless of which switch is activated, the pre-0 circuit is activated at the 0th logical frame and takes effect at the 1st logical frame. The display circuit is activated at the 0th logical frame, at the
1 to 2 logical frames take effect.

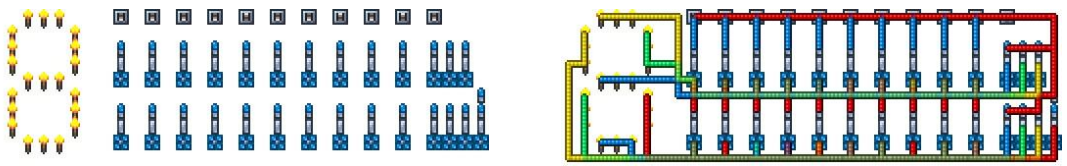


Figure **3.16**

3.6 Dual activation technology

Dual activation technique refers to the practice of activating the same logical light in two adjacent logical frames. Dual activation is typically achieved in Figure 3.17. Dual activation has many uses.

Figure **3.17:** The first time a red line is activated, the red line is activated again in the logical frame that follows because of the influence of the logic gate. The red line does not activate the third time due to the door burst mechanism.

3.6.1 The status of the normal logic gate is activated

There are three common practices for state-to-action activation of normal logic gates, as shown in Figure 3.18.

(a) Activated when the door is lit from the gate, and from **(b)** when the door is lit, the fault logic door can **(c)** the fault logic door can be lit and inactive when the door is lit. Activate, cannot be activated when off. Live, cannot be activated when it is extinguished.

Figure **3.18**

Figure 3.18 (a) Two activations are activated once using a down-frequency circuit. Figure 3.18 (b) activated at some point by the state control with the door. Figure 3.18(c) is the dual activation technique we need to emphasize here, and its function is similar to Figure 3.18(b). When the two logic lights below are fully lit, a double activation of the top logic light causes the door to be activated once and burst.

In other cases, dual activation has no effect. Figure 3.18(c) is less space-efficient than Figure 3.18(b) and does not require logical synchronization because the logic gate is always out before dual activation. "If only the module itself is considered, Figure 3.18(c) performs slightly below Figure 3.18(b),but since In most cases Figure 3.18 (b) requires careful logical synchronization, the performance is not as good as Figure 3.18(c)given the additional circuitry required to achieve logical synchronization."

3.6.2 Signal detection accurate to the logical frame

This content has been encountered in the network cable, where we implement dual activation through a series of logical delayers. If you detect only a single signal, you can use Figure 3.17 to eliminate a wire or a logic gate.

(a) (b)

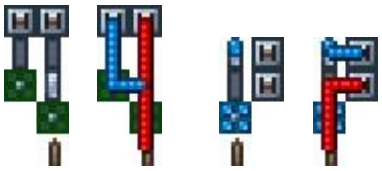
Figure **3.19:** The torch will only respond if the switch at the intersection of the blue and red lines is activated.

For example, two circuits in Figure 3.19 are activated only when the blue and red lines are activated in one logical frame. To put it another way, it can also be thought that one line here achieves signal detection accurate to the logical frame of the other. There are also nuances between the two circuits. The valid logic light on the fault logic gate is off in case the fault logic light is activated and will not burst the door, and the dual activation circuit is a one-time, so the activation condition of Figure 3.19 (a) is actually the first time that the blue line and the red line are activated in the same logical frame;

3.6.3 Embedded assignment circuit

We already know how to use faulty logic gates as assignment circuits. With dual activation technology, we can use ordinary logic doors as assignment circuits.

3.7 Randomly divided into two groups



(a) Embedded assignment 0**(b)** fault door assignment 0

Figure **3.20:** Two ways to assign. The red line changes the torch state and the blue line assigns the torch to 0.

Figures 3.20 (a) and 3.20 (b) are embedded assignment circuits and fault logic gate assignment circuits that we have learned before. In the embedded assignment circuit, nothing happens when the upper and lower lights go out and the lights go down by default, and a double activation of the lower lights is performed. If the red line state switched, then turn on the light, down the light off, the next light to perform a double activation, the first activation

with the door lighting activation, will turn off the up and down lights, the second activation of the downlight light, the circuit returned to the default state.

Figure 6.1 (e)can be achieved with an embedded assignmentcircuit. The Embedded name comes from drKV.

3.7 Randomly divided into two groups

In the previous chapter we learned about the circuit of the dice, i.e. randomly selecting 1 from n elements. This section describes circuits that randomly select m from n elements (or randomly divide n elements into two groups of elements, m and n-m, respectively). Understanding this section requires high school probability knowledge. Let's take random 8 for 4 as an example.

First consider which probability failure logic gates are required.

- The probability that the first element will be divided into group 1 is 4/8.
- Once the grouping of the first element is determined, you can group the second element in two cases. If the first element is assigned to Group 1, there is a 3/7 probability that the second element will be divided into Group 1, otherwise the probability is 4/7。
- "Similarly, if the first two elements are all in Group 1, there is a 2/6 probability that the third element will be divided into Group 1;

  Group 1.
- If the first three elements are divided into Group 1 by 3/2/1/0, then the probability of the fourth element having $\frac{1}{5}/\frac{2}{5}/\frac{3}{5}/$ is $\frac{4}{5}$divided into Group 1.
- If 4/3/2/1/0 of the first four elements are assigned to Group 1, then the probability of the fifth element having $\frac{0}{4}/\frac{1}{4}/\frac{2}{4}/\frac{3}{4}/$ is $\frac{4}{4}$divided into Group 1.
- If the first five elements are divided into Group 1 with 4/3/2/1, then the probability that the sixth element has $\frac{0}{3}/\frac{1}{3}/\frac{2}{3}/\frac{3}{3}$is divided into Group 1.
- If 4/3/2 of the first six elements are assigned to Group 1, then the probability of the seventh element having $\frac{0}{2}/\frac{1}{2}/\frac{2}{2}$is divided into Group 1

1 group.

- If 4/3 of the first seven elements are group 1, then the probability of the eighth element having $\frac{0}{1}/$ is $\frac{1}{1}$divided into group 1.

At first glance, this requires 24 fault logic gates, which are not yet included in the control circuit. If m and n are a little bigger, it's astronomical. A closer look reveals that the grouping probability for each element has many possible values, but

3.7 Randomly divided into two groups

These probabilities have two characteristics: the denominator is the same, and the molecule has a very simple relationship to the grouping of previous elements. This reminds us to use the grouping state of the previous element to directly modify the lighting of the valid logic lamp of the subsequent fault logic gate, thus modifying the grouping probability of subsequent elements.

The group of fault logic doors used for probability is shown in Figure 3.21.

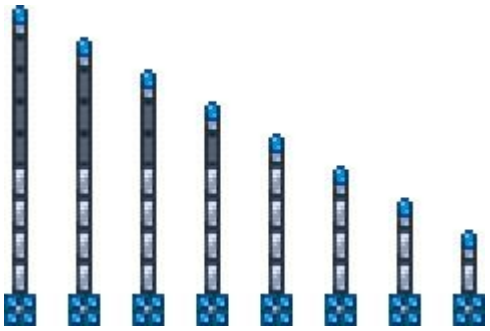


Figure **3.21**

Activate the first fault logic light and the first fault logic gate has a 4/8 probability activation. Let's assume that logical gate activation represents that the corresponding element is assigned to Group 1. When the first fault logic gate is not activated, we want the second fault logic gate to maintain the current 4/7 probability, and when the first fault logic gate is activated, we want the probability of the second fault logic gate to be modified to 3/7,i.e. a lit valid logic light is turned off. All in all, as long as Groups 1 and 2 are not full, each time the previous logical gate is activated, the probability of the back logical gate is reduced. When Group 1 or Group 1

2 When a group has been divided into four elements, the probability of the back logical gate remains unchanged at 0 or 1.

Since the probability reduction of the rear logic gate is determined based on the number of previous logic gate activations, this count function is done using the pass circuit. The pass circuit is entered by all fault logic gates for probability. The pass circuit activates 1 time, indicating that one element is assigned to group 1, and the valid logic light from the bottom to the upper 4th row goes out, leaving the molecules of the back group at most 3; ; The relay circuit is activated 4 times, indicating that four elements have been assigned to Group 1, the bottom row of valid logic lights off, at which point all valid logic lights go out, representing a probability of 0behind, i.e. all remaining elements are assigned to Group 2. For easy wiring, an oblique pass circuit is used (Figure 3.22).

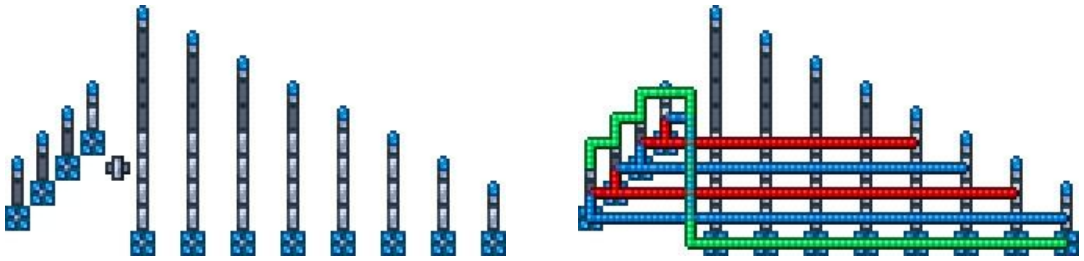


Figure **3.22**

For reuse, a reset circuit is also required to restore the circuit to its original state after all eight fault logic lights for probability have been activated. When fully activated, the relay circuit must have been activated 4 times, so the relay circuit has been reset automatically, the valid logic light must be completely extinguished, and the reset circuit only needs to be fully lit by the required valid logic light (Figure 3.23).

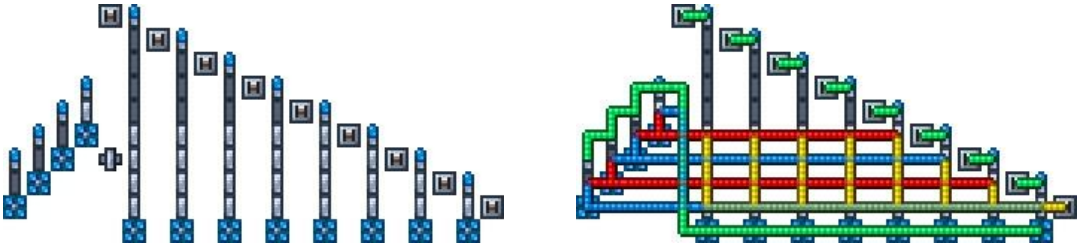


Figure **3.23**

In the circuit above, we need to click on eight switches from left to right, and the corresponding fault logic gate activation represents the corresponding element being divided into [4]Group 1. After the score, you need to click on the ninth switch to reset. The ideal circuit, of course, is to automatically complete all functions with just one switch. The entire circuit should run in the following order: the first fault logic lamp is activated, the first fault logic gate (possible) is activated, the relay circuit is activated, the valid logic light is modified, and then the second fault logic light is activated,... After the eight fault logic lights are activated, the reset circuit is activated. Therefore, a logical delay is required between each of the two adjacent fault logic lights, and there is a logical delay between the last fault logic lamp and the reset circuit. Next, you need to calculate at least a few logical frames that need to be delayed.

When a fault logic lamp for probability is activated at the kth logical frame, the fault logic gate is activated at the k1 logical frame and the relay circuit is activated at the k2 logical frame, and the corresponding valid logic light is turned off. Therefore, the next fault logic light activation cannot be earlier than the k2 logical frame, otherwise the probability has not been modified.

That is, the logical delay between the two adjacent fault logic lights is at least 2 logical frames.

When the last fault logic lamp for probability is activated at the k-frame, the fault logic gate is activated at the k-1 logical frame and the relay circuit is activated; Since both the relay circuit and the reset circuit only operate on the valid logic lamp, so who first who and then on the circuit operation results have no effect, only need to reset the circuit run without interfering with the probability of the fault logic door judgment. Therefore, the reset circuit is activated no earlier than the k1 logic frame, i.e. the reset circuit is activated at least one logical frame later than the last fault logic light.

After you've mastered the logic, you're ready to wi it. As shown in Figure 3.24,two wire breakers are added between each of the two adjacent fault logic lights to construct a delay of 2 logical frames, and a wire changer is added between the last fault logic light and the reset

[4] Note that the grouping is represented here by activation, so if you use torch display, additional set-up is required 0 circuit。

circuit to construct a delay of 1 logical frame. The torch is added to the pre-0 circuit to convert activation to lighting .

Here's just an example of random 8 pick 4. Random m-select n circuits are constructed exactly the same principle and construction except for the number of logic gates and logic lights.

3.8 Randomly divided into three groups

In the previous section, we introduced circuits that are randomly divided into two groups, and this section introduces random groups, the principle of which can be applied to random division of multiple groups. The circuit in this section needs to divide the nine elements into three groups of 3 plus 3 plus 3.

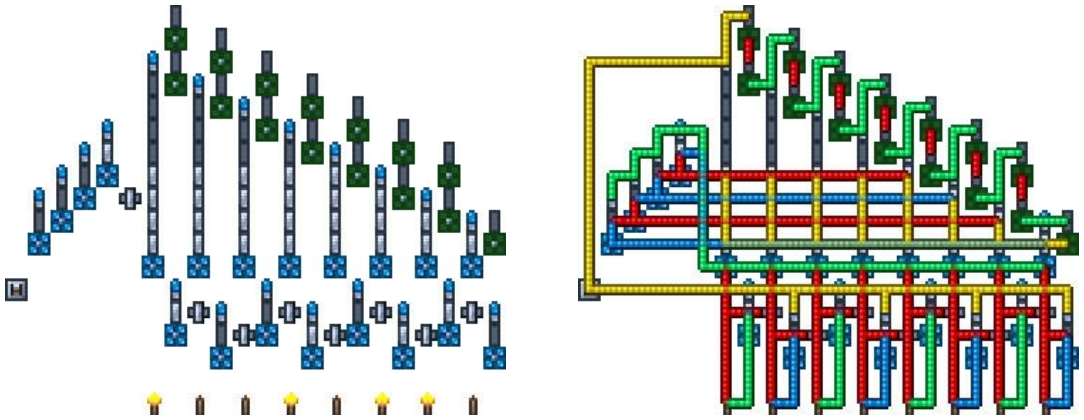


Figure **3.24**

3.8.1 Probability module

To divide 9 elements into three groups, you only need to divide into two groups of 6 plus 3 and then into two groups of 6 elements into 3 plus 3 groups. The circuit is shown in Figure 3.25. The grouping circuit above will be assigned to a set of 6 activations. Each element is grouped into 6, and the relay circuit activates the corresponding fault logic light for the circuit of the following 6 to 3. If this logical gate is activated again, the element is assigned to Group 1, otherwise it is assigned to Group 2. If pick 9 and 6 are not activated at the beginning, they are assigned to Group 3. Because the delay of 2 logical frames has been set for the two adjacent fault logic gates in 9-choice 6, the delay between the next 6-choice 3 adjacent fault logic gates is already at least 2 logical frames, so adjustment is no longer required. Due to the color of the wire,the reset circuit of 6 pick 3 is slightly different from the 9 pick 6.

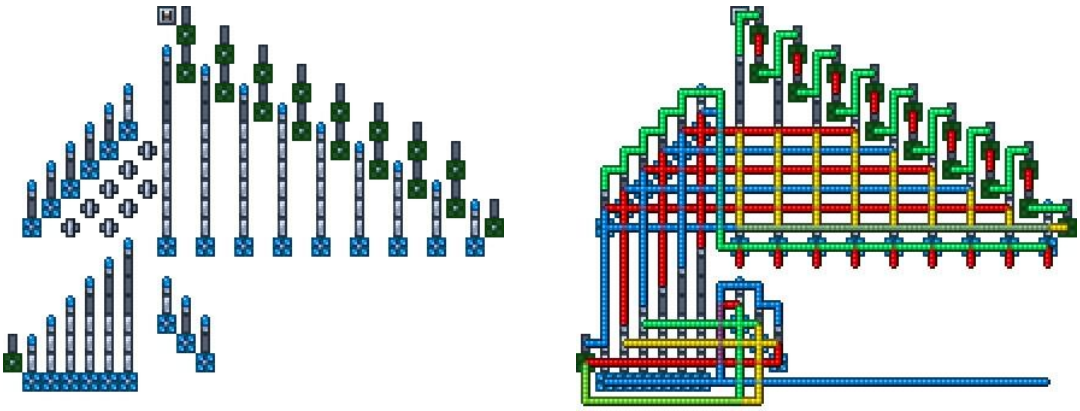


Figure **3.25**

By identifying the activation signal from the core module, the grouping can already be determined: if one of the fault logic lights in 9 option 6 is not activated, the element is assigned to Group 3, and if the fault logic light is activated, but after two logical frames,6 The blue line below option 3 is not activated, indicating that the element is assigned to Group 2, and if the two logical frames after the fault logic light is activated, andthe blue line below option6 3 is activated, the element is assigned to Group 1.

If you want to visualize grouping, you need to make a judgment:6 Choose 3 Is the blue line below 9 select 6 after a fault logic light 2 logical frame activated? This precision to the logical frame needs to be achieved through the following activation and door.

3.8.2 Letter processing

Now let's go back to the circuit of random grouping. We already know that when a fault logic gate of 9 pick 6 is activated, if theoutput of 6 pick 3 happens to be activated after two logical frames, the corresponding element is assigned to Group 1. We can do this by accurately detecting signals to logical frames.

Select the signal processing method, and then choose the visualization method. Torches have only two states and cannot represent a split into three groups, so only multiple torches or colored light bulbs can be used. Colored light bulbs are more intuitive, so here's the colored light bulb: with 9 colored light bulbs, the bulbs show red, blue, and green for 3 groups.

Since the signal processing circuits of the nine bulbs are identical, only one principle is mentioned here. We do this signal processing circuit with two inputs (red and blue). If the red line is not entered, the bulb is red, if the red line is entered and the second logical frame after the blue line is not entered, and if the red line is entered and the blue line is entered by the second logical frame after it, the bulb is green. In addition, the red line is entered up to once in the logical settlement, and the blue line is bound to be entered three times. Since the bulb is red when there is no output, the bulb can be set to red using the set 0/set 1 circuit. When the

red line is activated, turn off the bulb's red and the blue line on. When the blue line is activated after two logical frames, the blue goes out and the green lights up. The circuit is shown in Figure 3.26. Connect two wire changers on the red line so that the decision becomes that the blue goes out when the red and blue lines are activated in the same logical frame, and the green lights up, thus applying activation and the door. The left and right fault logic doors are used to place the colored light bulb in red in advance. [5]

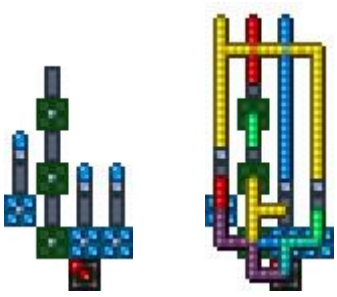


Figure **3.26:** The red, blue, and yellow lines above are red line input, blue line input, reset.

3.8.3 Circuit optimization

There is a problem with attaching the signal processing device to the bulb and the core module: the space spacing of the core module output is narrow (2width peroutput); However, the signal processing unit above is 4in width, so either the wiring will be ugly (Figure 3.27) or the distance between the core module and the colored light bulb will only be forcibly widened. Therefore, space compression for the signal processing module is necessary. Figures 3.28 and 3.29 are solutions that take up widths of 2 and , respectively, and are not sure [6]whether they are the optimal solutions for the corresponding widths. The smaller the footprint width, the greater the occupancy height 3. Circuit optimization has no fixed circuit, need to sum up experience through a lot of practice and try, so here omit the optimization process 10,000 words, only release the circuit diagram for reference.

[5] Note that although two colors of wires are attached to the fault logic gate, only one color wire is received with a valid logic light。

[6] If you have a choice, no one will use a circuit that takes up a lot of width and height。

The computational speed of the 3.9 circuit

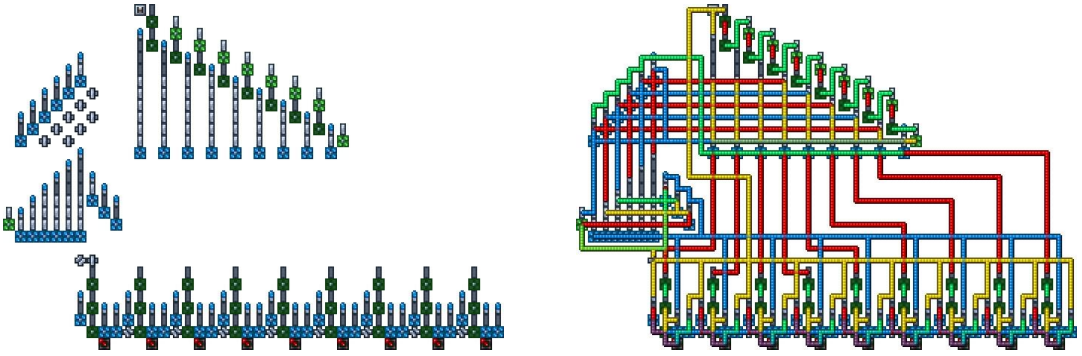


Figure **3.27**

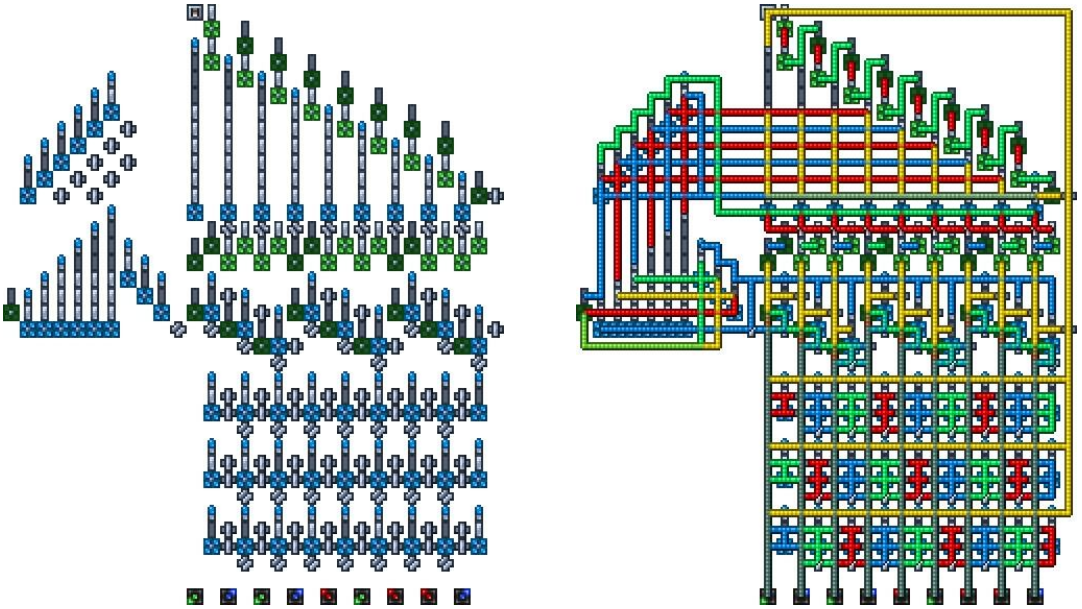


Figure **3.28**

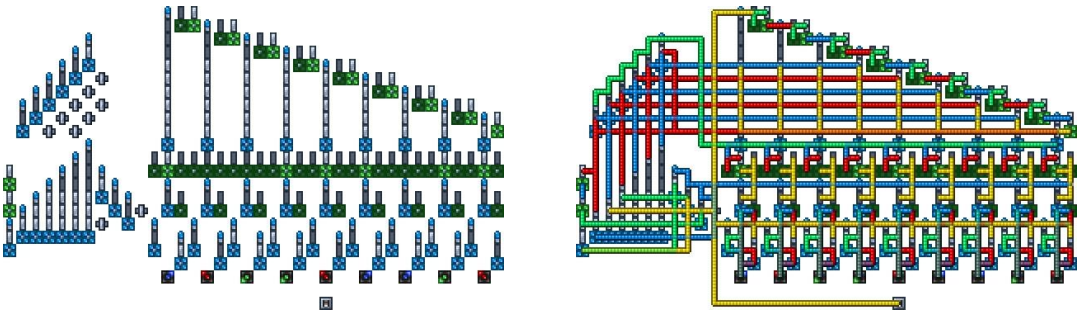


Figure **3.29**

The computational speed of the 3.9 circuit

Some simple studies have shown that the complexity of a circuit depends largely on the grid of the activated wire. So when the circuit is more complex, find a way to reduce the length of the wire, or reduce the number of activations of a wire can reduce the complexity of the circuit.

If the complexity of the circuit is too high, it can cause the host CPU to be slow enough, limiting the physical frame rate. A simple example is connecting 40,000 torches(100 x 400)using wires and then activating them with a full-frequency drive. On the author's computer, when the driver is turned on, you can see that the physical frame rate has dropped to about 32. This speed is affected by the performance of the computer.

3.9.1 Multi-stage relay circuit

Suppose we're going to make an ultra-large-scale pass circuit:1000-pass circuit. If we open 1000 logic doors directly, such a relay circuit activates once, activating the line connecting all the top lights (approximately 2000 blocks) and the corresponding line (4 blocks) on the logical door, and we ignore the line on the logic gate. Then the pass circuit runs for a cycle and the wires are activated

The grid is approximately 2000 x 1000 x 200000.

If we split the pass circuit flat into two segments, A and *B,* and start with *A,*and switch to *B*after *A* has run a cycle, this switch can be done through a pair of fault logic gates. We call this pair of fault logic doors a first-level pass,and*A* and *B* a-level pass." Then the entire circuit runs for a cycle, with 1000 first-stage passes and thousands of grids, and the second-level recidivations 500 times each, with grids of about 1000 x 500 x 2 x 1000000. As a result, the overall complexity is cut in half. If you continue to increase the number of stages, complexity will continue to decrease.

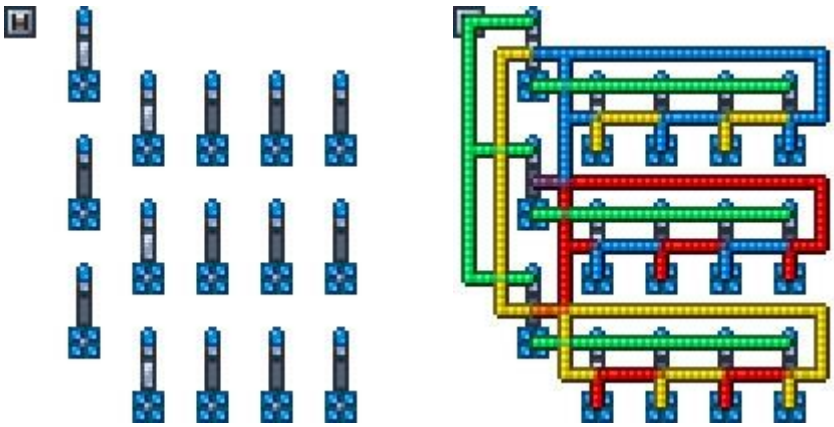


Figure **3.30:** Secondary pass for period 12

Traditional relay circuit cycle complexity is $O(n^2)$and the average single activation complexity is $O(n)$. With multi-stage pass-through circuits, the cycle complexity is minimal at $O(n \log n)$and the average single activation complexity is minimal at $O(\log n)$. Examples of multi-stage recurring circuits are shown in Figure 3.30 and Figure 3.31.

The ultra-large-scale pass circuit can be used for transmitter calibration animations, with one instance beinghttps://www.bilibili.com /video/av46694445

3.10 Experimental determination of the activation sequence

See Figure 3.32,a wire connection switch and twodarts organs, right-clicking the switch when it is clear that the two dart organs in the same physical frame shot darts. If you put two turquoise pressure pads at the same distance, it is clear that the two turquoise pressure pads are activated in the same physical frame, which is the two physical events that occur at the same logical frame.

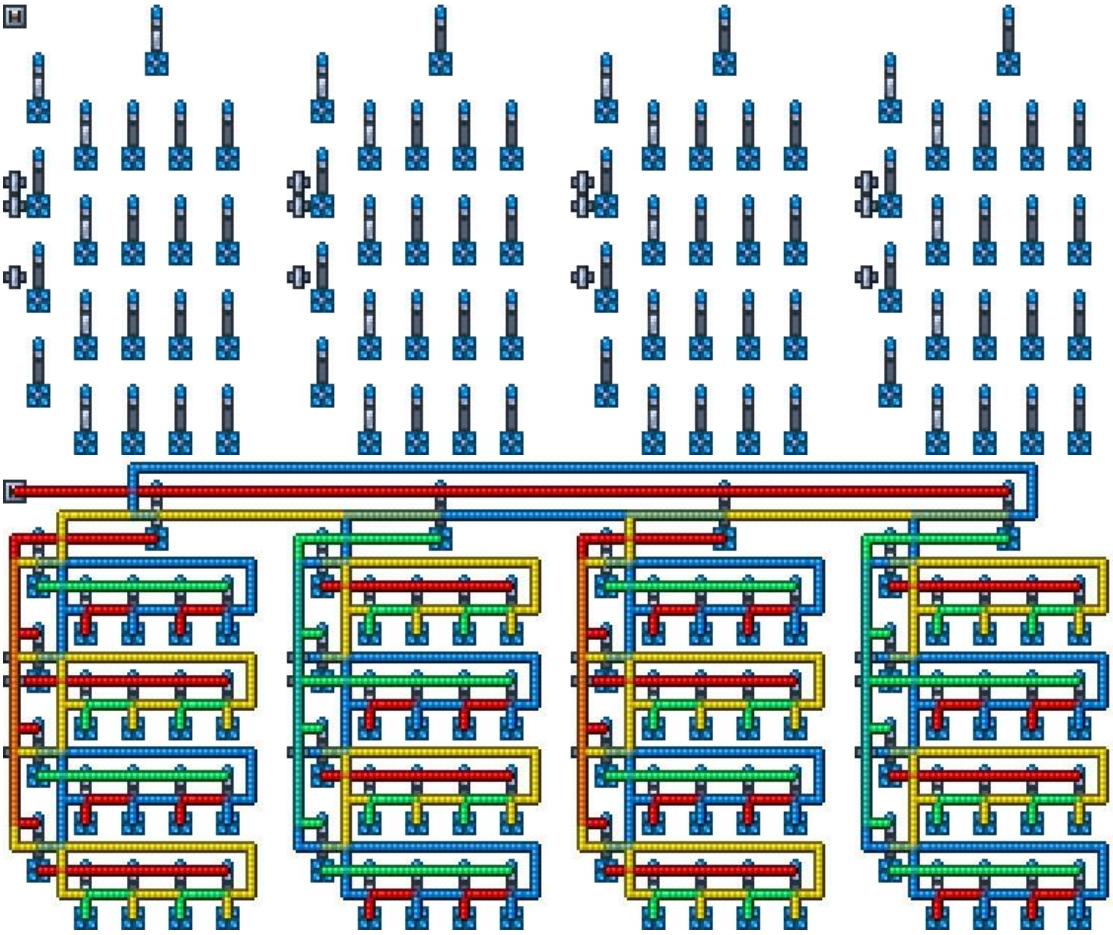


Figure **3.31:** Three-level recurring period of 64

Figure **3.32**

According to inertial thinking, since two darts are fired from the same logical frame, then the two turquoise pressure pads should also be activated in the same logical frame, then as shown in Figure 3.33, the first right-click switch should make the left torch respond. In practice, however, the first right-click switch will cause the torch on the right to respond, which means that the two turquoise pressure pads are not activated in the same logical frame, and the green pressure pads on the left are activated earlier.

So are the two turquoise pressure pads activated by different logical frames in the same logical settlement? Look at Figure 3.34,if the two turquoise pressure pads are activated by different logical frames in the same logical settlement, and the turquoise pressure pads on the left are activated earlier, then the door will burst with the door. In practice, however, the door is not opened, but activated twice, which means that the two turquoise pressure pads are not activated in the same logical settlement. This means that different physical power supplies do not participate in the same logical settlement, but wait for the logical settlement of the physical power supply-related front to be completed before performing the logical settlement of the subsequent physical power supply.

Figure **3.33**

Figure **3.34:** On the far right is the down-frequency circuit, where the torch response represents two activations with the door at a time.

From the above experiment you can see that the left side of the turquoise pressure pad settles in front and the right side settles in the back, which is due to the different order in which the darts that activate them are generated. According to Rule 5, the darts of the darts organ on

the left are generated after the darts of the dart organ on the right. From the beginning of dart generation, each logical frame needs to judge the collision of darts, because the left darts into, so judge the collision of the left dart first to judge the collision. So when two darts collide with the green pressure pad at the same time, the program puts the collision event on the left first, which is to arrange the trigger sequence of the two collision events to the left first right and then.

It is not very useful to know so much about the settlement mechanism because the most powerful logic circuits are not sensitive to these orders.

K Thinking question k

1. Make a signal processing circuit with two inputs a, b,which requires output activation when b is activated within 3 logical frames after a is activated. Where a is activated only once in the entire logical settlement.
2. To do a signal processing circuit, there are three inputs a, b,c,which requires output activation when a, b, c is activated at the same logical frame. where a, b, c is activated only once in the entire logical settlement.
3. To do a signal processing circuit, there are two inputs a, b,which requires that when a, b is activated at least once in a logical settlement within the same logical frame, the output is activated, whereas in the entire logical settlement, at most, it is activated within the interval of the two logical frames.
4. Prove that there is no signal processing circuit, and that there are two inputs a, b,which require output activation when a, b is activated in at least once in a logical settlement within the same logical frame. Where a, b may be activated any number of times in the logical settlement.

Chapter 4 Digital Circuits

There are logic gates in Tyraria, and naturally there will be digital circuit theory. And because of the nature of the game, the digital circuit system in Tyralia is almost completely different from that in real life. So don't have two illusions: I've learned (digital) circuits in reality, so I can easily handle the Tyralia circuits.

- I'm familiar with the Tyralia circuit, so I'm sure I'll understand the real-world (digital) circuit.

The content of this chapter is to establish the digital circuit system in Tylaria, and the second is to optimize the limits of the circuit by using the knowledge of digital circuit. This chapter is highly professional and recommends that people with the ability read it.

4.1 Combine logic with timing logic

There are three different functions in Tyraria: the door, the different or the door, and the fault logic gate. There are also two ways to transmit signals in a circuit: state transmission and activation transmission. State transfer means that if a torch is placed on a line, then the torch light indicates that the state of the line is 1and the torch extinguishing indicates that the state of the line is 0,and activation delivery means that in a particular logical frame, wire activation means 1and inactivation means 0. A circuit in which all wires are all state-passed is called a combined logic circuit, and a circuit where at least one wire is activated for transmission is called a timing logic circuit.

The characteristic of a combined logic circuit is that only the door and the different or different door in the circuit, and each line is activated only at a fixed logical frame (except for the burst door in the logic delayer). Each output of the combined logic should have a true value table.

Timing logic circuits use activation or not to transmit information, the same wire may be activated at different logical frames, and different logical frame activations may represent different signals. The relay circuit is a typical pure timing logic circuit.

Unlike real-world digital power, there are no explicit functional limitations to the combined logic and timing logic in Tyralia. There are many circuits that can be implemented either with combined logic or with timing logic. The combination logic circuit and the timing logic circuit each have advantages and disadvantages, a circuit chooses to use combination or timing, depending on the characteristics of the circuit itself.

- In the combined logic circuit, the operating state of the whole circuit can be obtained directly through the reading lamp, if there is a bug,directly look at which lamp state is not right, you can find the problem. In contrast, if a bug occurs in the timing logic circuit, it is difficult to debug unless MechScopeisused.
- Due to the problem of logic light wiring orientation, reverse winding which violates the orientation of logic lamp often occurs in the timing logic circuit, and wiring is more difficult than combining logic.
- For logical synchronization, logic delayers are often used in complex circuits. Each data in a composite logic circuit requires a separate logic delayer, and a logic delayer can be shared with all data in the timing logic circuit. Store all the data to be transferred temporarily and then send a signal by the central logic delayer to release the data. This is hard to see now, and it will be clear later in the divider section.

Many circuit concepts are divided into state and activation, and can also be classified as combined logic and timing logic.

For example, the pressure plate belongs to the combined logic power supply, while the ordinary pressure plate belongs to the timing logic power supply, the torch belongs to the combination logic appliance, and the pixel box belongs to the timing logic appliance.

The design of combined logic circuits requires Boolean algebra knowledge.

4.2 Boolean Algebra

4.2.1 Boolean operation

A number is a Boolean value, which means that the value of this number can only be 0 and 1. Boolean is an operation on the Boolean value. We refer to boolean values in expressions and Boolean constant 1 simply as letters.

There are four primary Boolean operations in Tyralia, the rules of which are as follows, and the meaning in the circuit is shown in Figure 4.1:

Take back 0 s 1, s1 s 0

Connections 00 s 0, 01 s 1, 10 s 1, 11 s 0

与 0&0 = 0, 0&1 = 0, 1&0 = 0, 1&1 = 1

XOR ˆ(a_1, a_2, · · · , a_n) = · ·$_1$, · · · , a_nThere happens to be one for1

1010 situation if a situation
If a other other

(a) 取反：A = ˜B, B = ˜A

(b) Connections:B - AC

(c) 与：C = A&B

(d)异或：D = ˆ(A, B,C)

Figure **4.1:** Circuits corresponding to primary Boolean operations. Note that the logic gate is not required for reverse operations and connection operations.

In addition, in order to understand, we also use some other Boolean operations, which can be derived from the primary Boolean operation. Figure 4.2 shows how to export or do operations using the same operations. Table 4.1 shows the commonly used non-primary Boolean operations.

Figure **4.2:** Exported or operated with an operation. C = A| B

The name of the operation	mark	Export the formula
or	$a \mid b$	$\sim(\sim a \& \sim b)$
Less than	$(a) < b$	$\sim a \& b$
Greater than	$a > b$	$a \& \sim b$
amount	$a == b$	$\sim ab$
Less than or equal	$a \le b$	$\sim(a \& \sim b)$
Is greater than or equal to	$a \ge b$	$\sim(\sim a \& b)$
Not equal to	$a! = b$	ab

Table **4.1:** Commonly used non-primary Boolean operations.

4.2.2 The nature of the primary Boolean operation

The law of exchange, the law of distribution, the law of binding, the law of connection, and, of, of, or all satisfies the law of exchange, changes the alphabetical order of these operations, and the result

Unchanged. Connections, and all satisfy the law of union, but are different or not satisfied with the law of union, because 1 is ˆ(ˆ (1, 1), 1) ≠ ˆ (1, 1, 1) is 0.

This condition is essential when *a*is*b* to 0!), and the connection meets the distribution law *abs*c s (as*c)*(bsc).

Other commonly used operating laws

	connect	and	XOR
The operation with 0	$0a = a$	$0\&a = 0$	$\hat{}(0, a_1, \cdots, a_n) = \hat{}(a_1, \cdots, a_n)$
The operation with 1	$1a = \tilde{}a$	$1\&a = a$	$\hat{}(1, a_1, \cdots, a_n) = \tilde{}a_1\& \cdots \&\tilde{}a_n$
with its own operations	$aa = 0$	$a\&a = a$ $\tilde{}a\&a = 0$	
Conversion operations	$\hat{}(a, b) = ab$	$a\&b = \tilde{}ab\&b$	$\hat{}(a_1, a_2, a_3, \cdots, a_n) =$ $\hat{}\tilde{}a_1a_3 \quad a_n, \tilde{}a_2a_3 \quad a_n, a_3, \quad , a_n$ $(\quad \cdots \quad \cdots \quad \cdots \quad)$

Here we prove the difference or conversion operation.

证明 令 $b = a_1a_2 \cdots a_n$，则 $a_1a_3 \cdots a_n = ba_2$，$a_2a_3 \cdots a_n = ba_1$。

Suppose $\hat{}(a_1, a_2, a_3), \ , a_n)$,then $a_1, \ . . , a_n$ has and only 1 1, so b is 1. Thus

$$\hat{}(\tilde{}a_1a_3 \cdots a_n, \tilde{}a_2a_3 \cdots a_n, a_3, \cdots, a_n)$$

$$=\hat{}(\tilde{}ba_2, \tilde{}ba_1, a_3, \cdots, a_n)$$

$$=\hat{}(a_1, a_2, a_3, \cdots, a_n) =1.$$

反过来，假设$\hat{}(\tilde{}a_1a_3 \cdots a_n, \tilde{}a_2a_3 \cdots a_n, a_3, \cdots, a_n) =1$。令 $b_1 = \tilde{}a_1a_3 \cdots a_n$，$b_2 = \tilde{}a_2a_3 \cdots a_n$，then $\hat{}(b_1, b_2, a_3, . , a_n) =1$。As before, $\hat{}$ can be introduced ($b_1a_3 . . a_n, \tilde{}b_2a_3 \cdots a_n, a_3, \cdots, a_n) =1$。A simple calculation provides that you cantell the number of$b_1a_3 \ a_n = a_1, \ \tilde{}b_2a_3 \cdots a_n$ is a_2, so $\hat{}(a_1, a_2, a_3), \ , a_n) =1$。 ■

Other marks When we're dealing with a set of Boolean values, we can number them. For example, an octal binary number can be expressed as $a_7, a_6, a_5, a_4, a_3, a_3, a_1, a_0$,in short $a_{7:0}$。 aThe highest bit of $_{7:0}$ is $a_7$and the lowest bit is a_0.

$a_{7:0}$ can be divided into two four-digit binary numbers $a_{3:0}$(which can be called a lower four bits) and $a_{7:4}$(which can be called a high four)."

4.2.3 True Value Table

For slightly more complex logical expressions,such as the one that isknownto be*in*theworld, it is difficult to see at a glance what they mean. The expressions in the example contain only three letters, and they have only eight possibilities for value, so we can just take these eight cases out of the picture and see what the result of the expression is in each case.

a	*b*	*c*	*˜ab&˜bc&abc*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

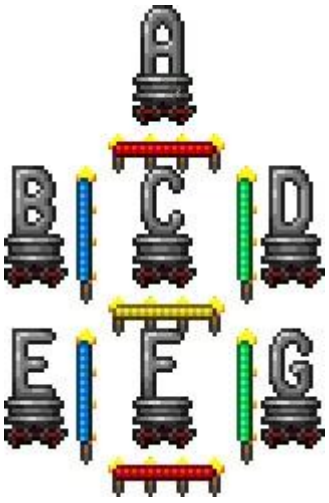
Table **4.2:** Thetrue value table of the ab*s/bc-abc.*

As you can see from Table 4.2, isn't that logic when and only when *a, b, c* is all 1, and the value of ab s bcs*bcsis* 1? So we get the logic of abs/bc-abc-a-b-c, and that's a good idea. Table 4.2 is called the true value table of the ab*s/bc-abc.*

More often, we get a real value table and then we want to get its logical expression. For example, for a display (Figure 4.3),we started by knowing only which inputs each display unit should be on and which should be dark, rather than how to build this logic. What we encounter here is the process of solving logical expressions with a known real value table. In the example given at the beginning of this section, because the true value table is so special, we get its logical expression just by observation. For simple observations where logical expressions are not available, subsequent sections give a solution.

input	*A*	*B*	*C*	*D*	*E*	*F*	*G*
0	1	1	0	1	1	1	1
1	0	0	0	1	0	0	1
2	1	0	1	1	1	1	0
3	1	0	1	1	0	1	1
4	0	1	1	1	0	0	1
5	1	1	1	0	0	1	1
6	1	1	1	0	1	1	1
7	1	0	0	1	0	0	1
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

(a)



(b)

Figure **4.3:** (a) A table of true values for decimal numbers;

The linear correlation of the 4.2.4 string is related to rank

Recall that we refer to the Boolean value and the Boolean constant in the logical expression [7]simply as letters. We refer to expressions that are connected by several letters as strings, or strings. Expressions connected by 0 letters are recorded as 0,called empty strings. Two strings of 0 and 1 are called ordinary strings. If two strings are constant, then the two strings are the same.

Note that by the nature of the connection 1*a* s*a*, boolean constant 1 plays a role in this string is reversed. For example, there are 8 different strings of 1,a,b: 0,1,a,b,ab,ab,a,ab,a,b,ab,ab.) A string can contain duplicate letters, but due to the nature of the connection, *aa* is 0 and 0*a* s *a,* a pair of duplicate letters can be directly offset.

Given a set of *strings* $s_1, \cdots, s_r$,selects several of the strings (at least one) to do the connection operation, and the resulting string is called a linear combination of the strings. If a group of non-ordinary strings has an ordinary linear combination, then we call this group of non-ordinary string linear correlation. Otherwise, if all linear combinations of a set of non-ordinary strings are non-ordinary strings, then we call this set of non-ordinary string linear irrelevance. The size of the largest linearly independent subset of a set of non-ordinary strings is called the rank of this set of non-ordinary strings.

theor 4.1

Remember the rank of n, then this group of strings has different non-ordinary linear $2^n - 1$. On the The number of different non-ordinary linear combinations and in this form this group of strings The r。 ♡

deduc 4.1

The r A different set of non-ordinary strings of $2^r - 1$ piece, a rank that is not the same as the At least $\lceil \log_2(n+1) \rceil$。 ♡

Example **4.1** considers three string abs,*abc, abcd.*

"Makes *A* s *ab, B* s *abc,C* s *abcd,* then all the different linear combinations of *A, B,C* are *A* s *ab, B* s *abc."* ,*C* = *abcd, AB* = *c, AC* = *cd, BC* = *d, ABC* = *abd*。 All linear combinations are non-ordinary strings, so ab, abc, abcd linear independent, thelargest linear independent subset ofab, abc, abcd is itself, so ab, abc, abcd rank is 3.

Example **4.2** Consider three strings of bc, ac,ab.

"So that *all* the different linear combinations of *A,* B, and*C* are *A, B,b, B,* B." *AC* = *ac,C* = *AB* = *ab, ABC* = 0。 *ABC* - 0 is an ordinary string, so bc, ac, ab linear correlation.

[7] 由 n How many different strings of letters are there??

All linear combinations of A and B are *A* s *bc, B* s *ac, AB* s ab, are non-ordinary strings, so bc, ac linearly independent, so bc, ac, ab rank is 2.

4.2.5 Simple logic and its complexity

Simple logic refers to logic that can be achieved with only one door or different or door. The concept is well understood, but it is much more complex than we thought.

Consider a simple example. Suppose we now make the logic of *a*and*b* with a two lights and the door, and in the game we connect two logic lights with two lines of different colors. In practice, a and *b* can sometimes be trans-positioned using the law of exchange with logic for wiring reasons. But the logic of equivaleity goes far beyond these two. In the conversion operation of section 4.2.2, you see that *a*s*b* s*ab*s*b,* synonymous *with a*s*b* s*ab*s*a,* reuse the exchange law, and *a*s *b* = *b*&˜*ab* = *a*&˜*ab*。 "Thus, there are 6 equivalent two-lamp simple logics for*a*and*b,* and 3 regardless of the exchange law." Each of these equivalent logics represents a different wiring method, and a variety of wiring methods can help us to swing the wire in some small spaces.

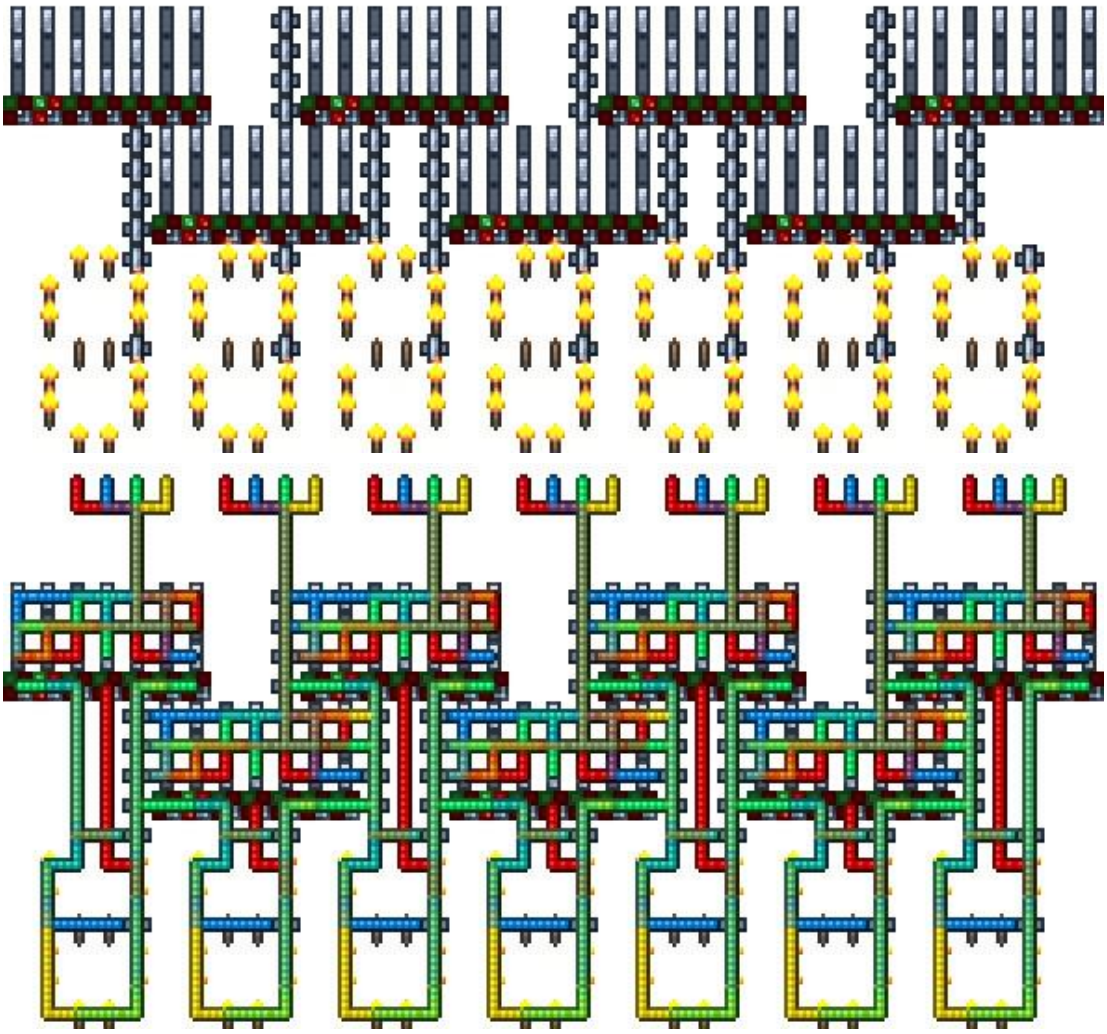
Similarly,there are 168 equivalent three-light simple logics for*a*and*b*, and 28 for the exchange law. For heterogeneous or logical, it is possible to obtain its equivalent logic with another formula in the conversion operation. Each logic corresponds to a wiring, so many equivalent logic will bring many kinds of wiring methods, which will increase the complexity of the problem we consider. In general, there are always some of these wiring methods that meet the requirements, but if not, we need to introduce a much more complex step logic.

How do 168 equivalent logics work out? Such a workload is unacceptable to the labor, even if there is no more skill, let alone the situation that the labor may be miscalculated. For simple logic, we developed the software TCLC(Terraria Groupal Logic Calculator),and you just need to enter the true value table of thislogic, and it will use the exhaustion method to find out which logic meets the requirements. The program downloads https://www.bbstr.net/threads/154/. This software has only been initially developed because there is not much actual demand yet. If you have more needs or ideas, you can contact me.

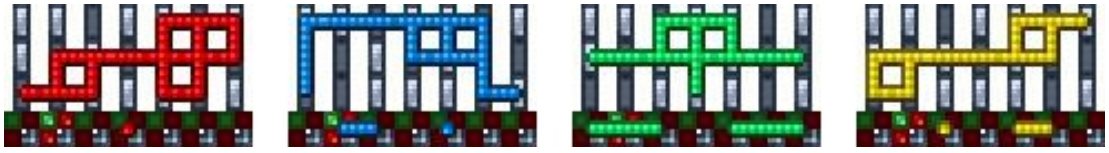
4.2.6 Step logic and its complexity

Step logic is the result of several simple logical connections. "Step" means that this logic can be settled within a logical frame, or that the input and output are only one logical frame apart. Similarly, we have k-step logic.

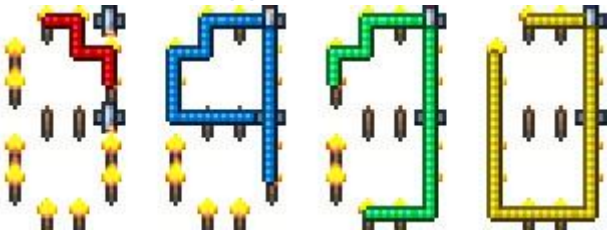
What's the use of step logic? If you want to execute multiple simple logics at the same time in a circuit, you can connect several of them for possible simplification. This description is very abstract, let's look at an example.



(a) The number of stackables can be displayed



(b) Control section



(c) Show section Figure **4.4**

Figure 4.4 shows a BCD number. This number is the only BCD number that can be secretly stacked (Figure 4.4 (a)) found so far. The requirements for dense stacking are extremely demanding. First, because the width of the lower display is small, the width of the control circuit

is very limited; These conditions may seem incredible, but the numbers can be made thanks to the extra complexity of step logic.

This number shows a total of 7 outputs, and because they are ranked 7,at least 7 logical gates arerequired. When the input is connected, there can be various wiring possibilities on the logic lamp, the logic gate can be selected and different or, the number of logic lights can also be changed, the output, multiple logic doors can be output to the same wire to achieve connection operations, wires can also be connected to the display several segments. In this way, regardless of correctness, a simple number of apparent wiring arrangement combinations can have thousands of kinds. Finding a viable solution from such a complex situation is not too difficult, but it also requires considerable observation and experience.

Example **4.3** A combination logic has three inputs *a, b,c,* and two *outputs, d, e,* whose true value table is shown in Table 4.3, to try to design the circuit.

a	*b*	*c*	*d*	*e*	*de*
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	1	0

Table **4.3**

Using TCLC,we can calculate that *d* has four lights and *e* has a single light:

$$d = \hat{}(abc, \tilde{}c, \tilde{}b, \tilde{}a) = \tilde{}\hat{}(c, b, a, \tilde{}abc)$$

$$e = abc$$

Both *d* and *e* are expressed in simple logic. If we think the expression of *d* is too complex, [8] we might consider using a connection operation that does not use a logical gate.

Considering the true value table of *de* (the far right column of Table 4.3), we can calculate that *de* has a double light for :

$$de = \tilde{}ac\&\tilde{}ab = \tilde{}bc\&\tilde{}ab = \tilde{}bc\&\tilde{}ac$$

"Thus,*e* is still represented by a single lamp, and *d* uses a dual light to connect to a single lamp: *d* s *dee* s*(de) e,* and we get the circuit shown in Figure 4.5." In this example, we reduce

[8] In fact, it is n input logic, our expectation is to use n -1A light can be done, and the results here are clearly not satisfactory.

the number of lights by considering the connection between the outputs. On the other hand, considering the logic of connectivity broadens our thinking, increases the possibilities, and gives us more hope of designing circuits that meet the harsh conditions.

Similar examples include logical units of binary and decimal interoperability using the plus-three shift method, which are covered later.

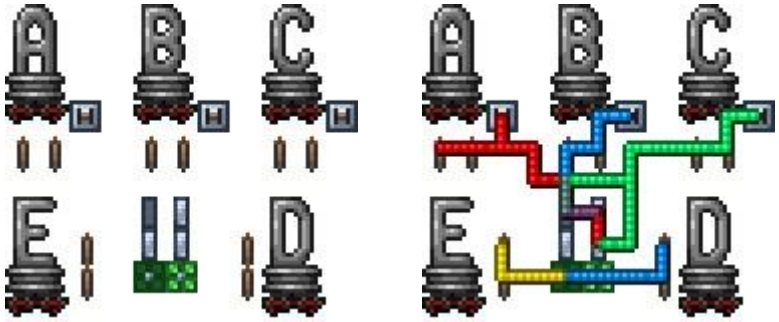


Figure 4.5

4.3 Arithmetic circuit

4.3.1 Adder

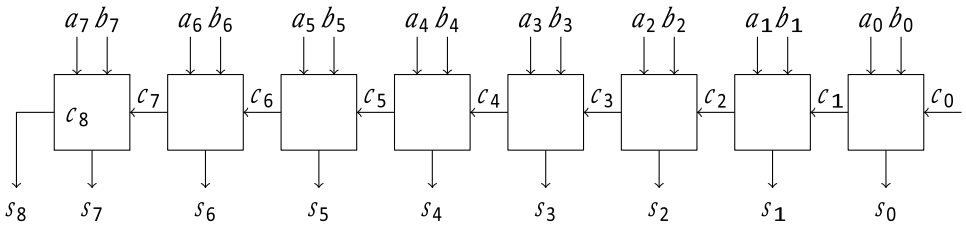
The input of the adder is two integers, and the output is the sum of the two inputs. There are many solutions for adders, and one of the most direct ways is to series all-adders.

What is a full adder? Let's look at the vertical calculation of binary addition. When calculating $a_{7:0}$and$b_{7:0,}$ first list the verticals shown in Figure 4.6 (a) and fill in the two additions. Then calculate a_0 plus $b_{0,}$which maybe 00, 01, 10. The low of the result is directly the lowest bit of the sum, written at the position *of* s_0. The high of the result is the entry of the lowest bit forward, written at the position *of* c_1. Then calculate a_1 plus b_1 plus $c_{1,}$fill the low of the resultwith $s_{1,}$fill in $c_2$as the highposition, and soon. This entire addition process can be simulated using the circuit shown in Figure 4.6 (b), which consists of eight small modules, each of which calculates a_i and b_i and $c_{i,}$ outputs the results low to $s_{i,}$and highs to the next small module, $c_{i.}$ Each of these small modules is called a full adder.

加数 1 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 加数 2 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 进位 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0

和 s_8 s_7 s_6 s_5 s_4 s_3 s_2 s_1 s_0

(a) Added vertical diagram. In general, c_0 is 0



(b) The circuit performs addition operations figure **4.6**

The all-in-one receives three *inputs,a, b,* c_i, whereas *a, b* is two additions, andc_i is the input entry. The all-in-one produces two *outputs, s,* c_o, where *s* is and c $_o$ is output feed. The true value table for the all-adder is table 4.4. This logic is discussed in Example 4.3.

With the all-in-one, connecting multiple all-in-ones directly to the end gives you an adder (Figure 4.7). This adder requires two additions to delay a logical frame input from low to high.

a	*b*	c_i	c_o	*s*
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Table **4.4:** The true value table of the all-adder

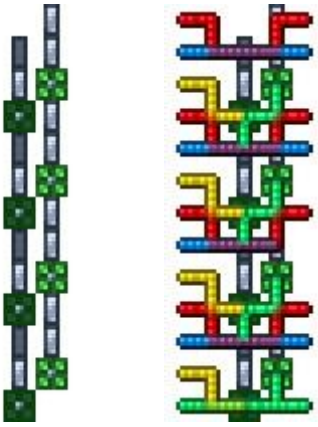


Figure **4.7:** Four adders. The red and blue lines represent two additions, the green line is the entry, the yellow line is the sum, and the bottom green line is the overflow position.
High below the upper low.

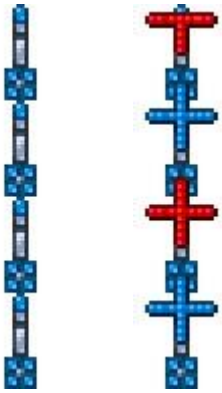


Figure **4.8:** Four-bit accumulator. The two additions are entered in turn into the red and blue lines.
High below the upper low.

In addition to combining logic, we can also use timing logic as adders. The down-frequency circuit has the characteristics of two-in-one, so you can use the down-frequency circuit directly as an adder (Figure 4.8). Unlike a combined logical adder, a timing logic adder is essentially a

accumulator, and if it is not reset, each input is added directly to the previous result. In a timing logic adder, the second addition is entered in the same logical frame, or the low delay is entered, as opposed to the combined logical adder.

Another advantage of timing logic in adders is that by changing the down-frequency circuit to a pass-through circuit, you can simply add up to any feed. The feed conversion in https://www.bilibili.com/video/av40474377/ is actually a decimal accumulator, adding a value or subtracting a value to the accumulator for every binary digit modified.

4.3.2 Subtractor

With the complement tool, subtractors and adders can share a circuit, and we just need to add some parts to reverse one of the additions. The reverse operation is easy, and the plus one operation is just to borrow *c*0 that we didn't use in Figure 4.6 (a). The combined logic addition and subtraction machine is shown in Figure 4.9, it is important to note that the adder's input delays a logical frame from low to high, so for each digit to reverse, also from low to high delay a logical frame, the work is done in Figure 4.9 the right column of logic gates.

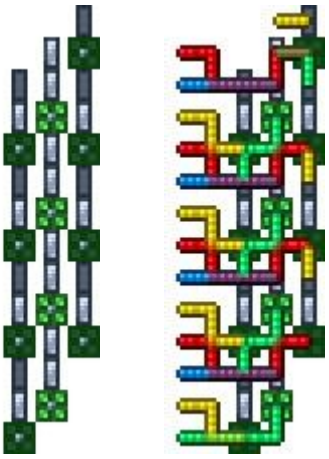


Figure **4.9:** Four-bit add-subtractor. The top yellow line controls addition and subtraction,0 is addition, and1 is subtraction. The red and blue lines represent two additions, respectively, when adding is calculated. When subtraction is calculated, the blue line is subtracted and the red line is the minus. The yellow line is the sum and the bottom green line is the overflow bit. High above low below.

The subtraction of timing logic is also using complement operations, and no specific implementation is given here.

4.3.3 Comparator

Comparing the size of two numbers is the simplest way to compare from high to low. For example, compare $a_{s7:0}$ and $b_{s7:0}$

Two 8-bit binary numbers, first comparing the highest bits a_7 and b_7,and then *comparing* $a6$ and b_6 if they areequal, and soon. Finally, we can get a logical expression:

$a_{[7:0]}>b_{[7:0]} = (a_7>b_7)(a_7==b_7\&a_6>b_6)(a_7==b_7\&a_6==b_6\&a_5>b_5)$

$(a_7==b_7\&a_6==b_6\&a_5==b_5\&a_4>b_4)$

$(a_7==b_7\&a_6==b_6\&a_5==b_5\&a_4==b_4\&a_3>b_3)$

$(a_7==b_7\&a_6==b_6\&a_5==b_5\&a_4==b_4\&a_3==b_3\&a_2>b_2)$

$(a_7==b_7\&a_6==b_6\&a_5==b_5\&a_4==b_4\&a_3==b_3\&a_2==b_2\&a_1>b_1)$

$(a_7==b_7\&a_6==b_6\&a_5==b_5\&a_4==b_4\&a_3==b_3\&a_2==b_2\&a_1==b_1\&a_0>b_0)$

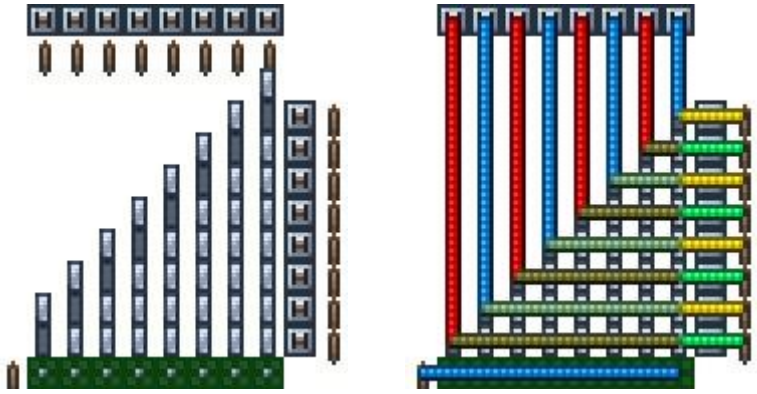


Figure **4.10:** Bit-by-bit comparator. The torch above represents b, the high on the left, the low on the right, the torch on the right, a, the lower high above, and the torch on the lower left, $a_{7:0}>b$, and $7:0$.

Using the export formula in section 4.2.1, we get a logical expression that contains only the primary Boolean operation:

$a_{[7:0]} > b_{[7:0]} = (a_7\&b_7)(\tilde{}a_7b_7\&a_6\&b_6)(\tilde{}a_7b_7\&a_6b_6\&a_5\&b_5)$

$(\tilde{}a_7b_7\&a_6b_6\&a_5b_5\&a_4\&b_4)$

$(\tilde{}a_7b_7\&a_6b_6\&a_5b_5\&a_4b_4\&a_3\&b_3)$

$(\tilde{}a_7b_7\&\tilde{}a_6b_6\&\tilde{}a_5b_5\&\tilde{}a_4b_4\&\tilde{}a_3b_3\&a_2\&\tilde{}b_2)$

$(\tilde{}a_7b_7\&a_6b_6\&a_5b_5\&a_4b_4\&a_3b_3\&a_2b_2\&a_1\&b_1)$

$(\tilde{}a_7b_7\&\tilde{}a_6b_6\&\tilde{}a_5b_5\&\tilde{}a_4b_4\&\tilde{}a_3b_3\&\tilde{}a_2b_2sa_1b_1sa_0sb_0)$ Convert this logical expression to a circuit as shown in Figure 4.10.

Comparing the size of two numbers, in addition to bit-by-bit comparison, you can subtract the two numbers directly and judge the positive and negative results. Subtractors we've done,

and in the complement representation, the plus or minus of a number can be derived directly from the highest bit. Here are two details. Unsigned integers have no signed bits, how do I compare the size of two unsigned integers? The difference between the two numbers of signed integers can be 127 ,or less than 0 ,and this number as a signed integer can be1,that is , less than 0. How are these special cases handled? In fact, our comparator here can only compare unsigned integers, whereas the so-called "highest bit" actually refers to the overflow bit *s*8 in Figure 4.6 (a). There are two ways to handle signed integers: the first is to judge the symbol first, and then make a comparison of the unsigned integer if the symbols are equal, and the second is to add both numbers to 128,which turns the range of 128 to 127 into 0 to 255,and then to make acomparison of the unsigned integers.

Comparing whether two numbers are equal also has two solutions, one is to compare directly bit by bit, and the other is to determine whether their difference is 0. These two methods do not have absolute advantages and disadvantages, although bit-by-bit comparison of the circuit will be relatively simple, but the difference can be judged directly on other functions, such as adder.

4.3.4 Multiplier

Multiplication is the promotion of addition, and multiplication is also the promotion of addition. Let's first look at the vertical calculation of binary multiplication. Because of the scale of multiplication, here is an example of a 4-bit multiplication (Figure 4.11).

As you can see from Figure 4.11, doing a 4-bit multiplication is actually adding four numbers. The addition of 4 numbers can be split into three times, 2 numbers, in order of operation, as shown in Figure 4.12.

				a_3	a_2	a_1	a_0
				b_3	b_2	b_1	b_0
				b_0&a_3	b_0&a_2	b_0&a_1	b_0&a_0
			b_1&a_3	b_1&a_2	b_1&a_1	b_1&a_0	
		b_2&a_3	b_2&a_2	b_2&a_1	b_2&a_0		
	b_3&a_3	b_3&a_2	b_3&a_1	b_3&a_0			
H_3	H_2	H_1	H_0	l_3	l_2	l_1	l_0

Figure **4.11:** Multiplication vertical calculation

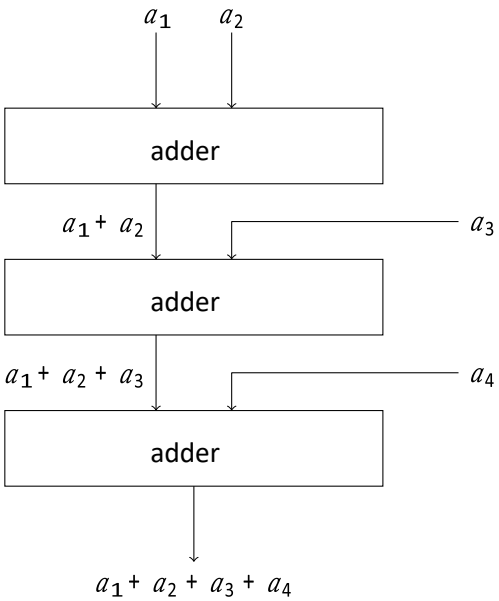


fig 4.12: Four a_1, a_2, a_3, a_4 Even

By expanding the adder in Figure 4.12 into a series of all-adders, a structural diagram of the four multipliers is obtained (Figure 4.13).

The individual unit circuits of the multiplier are shown in Figure 4.14,which marks areas and lines with background walls of different colors. The orange area is the all-in-one. The yellow area is with the door. The blue line is a multiplier entered from the right, delaying 1 logical frame for each cell passed. The red line is a multiplier entered from the top, delaying 2 logical frames for each cell passing through. The purple line is a horizontal inbound pass. The green line is from top left to bottom right and pass.

The effect of cell stacking is shown in Figure 4.15,where each cell size is 3 x 7. The optimization process uses the following wiring techniques: the left and right direction of the

- red and blue wire cycle, easy wiring.
- Adjacent two columns are misaligned to facilitate the transfer of entry and lateral multipliers.
- Vertical logical delay uses a different or door so that the door does not need to make way for the passing green line.

With the door, the equation *asb* s*absb*iseasily wired.

4.3.5 Divider

1

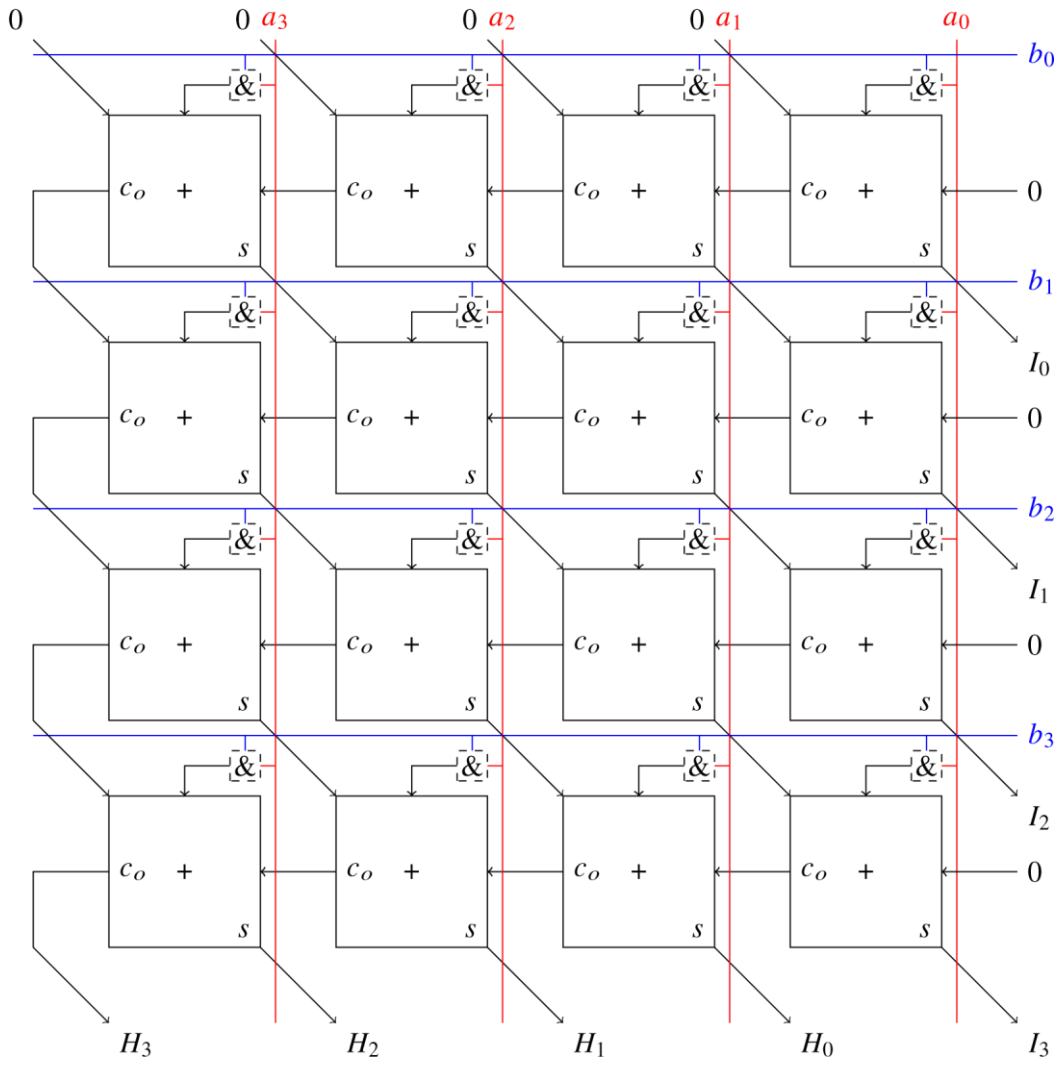


Figure **4.13:** A diagram of the structure of the four-digit multiplier. A large solid box represents a full adder, c_o represents a feed output, *and s* represents and output. Small dashed box represents output with door, red and blue line input, black line. The two multipliers are entered from the top and right, the low of the product is output from the right and the high is output from the bottom. The input missing from the all-adder is zeroed. Two left and right all-in-ones, one earlier logical frame on the right, two up-and-down all-adders, and two earlier logical frames on the top.

Figure **4.14: The** structure of a single unit of a multiplier

4.3.6 Shifter

Shift is divided into logical shift, arithmetic shift, circular shift, in which logical shift and arithmetic shift are widely used.

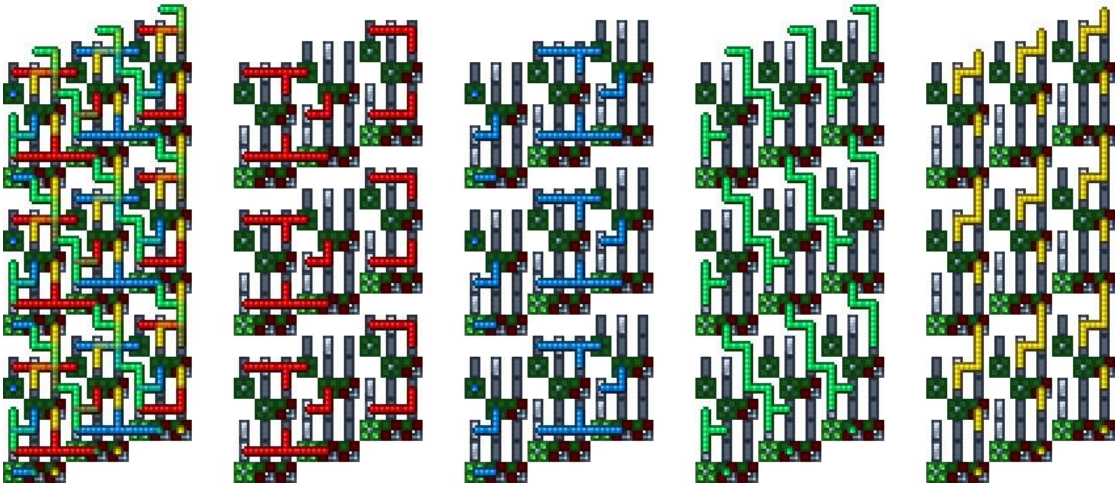
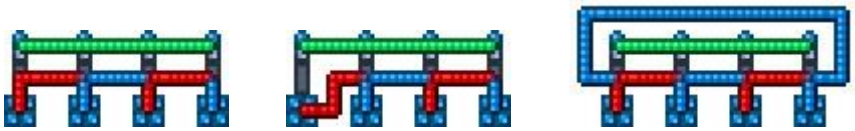


Figure **4.15:** Stacked optimized effect



(a) Logical right shift one **(b)** Arithmetic move one bit to the right **(c)** Loop to the right One Figure **4.16:** The passing circuit shifts

If only one bit is moved, all three shifts can be achieved through the pass circuit, as shown in Figure 4.16.

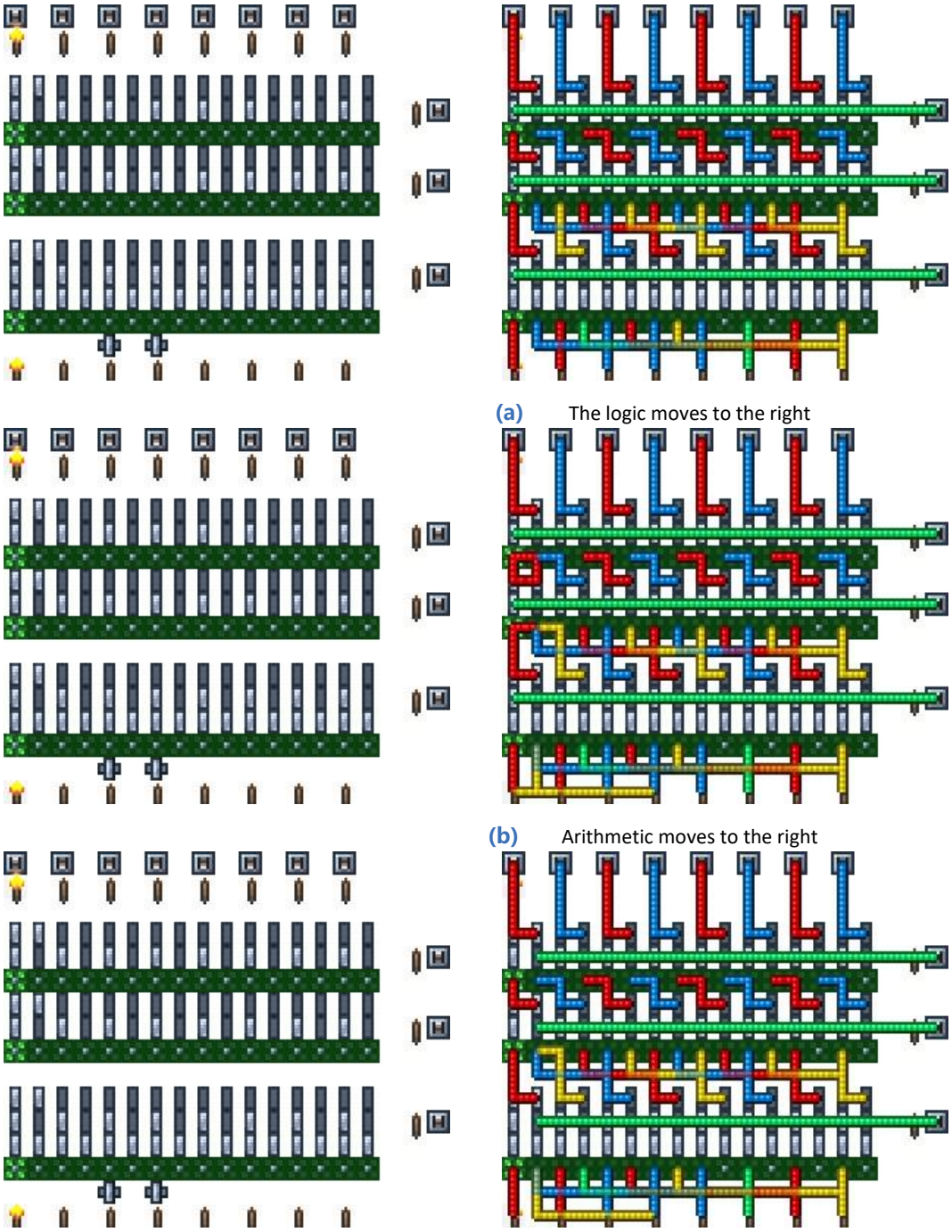
If you want to move any bit, the pass circuit probability will burst the door, and other scenarios need to be considered. A simple idea is to use the combined logic shown in Figure 4.17. The disadvantage of this combination logic is that all digit inputs of the shifted bits are to be entered in the same logical frame, while the digits of the shift are entered in turn at each logical frame, and adjusting the logic synchronization is more complex. In addition, due to the orientation of the logic gate, this method is not suitable for circular shifting.

In fact, because the structure of digital circuits is relatively complex, in Terralia want to do a certain size of digital circuits, we must consider the use of network cable technology to transmit data. Network wire technology can be used to design a specially compatible network cable shifter by sending different digital propagation data at different logical frames. Figure 4.18 Shifts by selecting the starting digit of the network cable output.

4.3.7 Random Number**Generator (Random Number Generator)**

In reality, the functions that need to be implemented by complex algorithms are very simple in Terraria. Use 8 directly

With a 1/2 probability of a fault logic gate, we can randomly generate 8-bit binary numbers.



(a) The logic moves to the right

(b) Arithmetic moves to the right

(c) Simplely optimized arithmetic right shift Figure **4.17:** Shifter that combines logical principles. The three torches on the right act from top to bottom by moving 1 bit, 2 digits, and 4 bits.

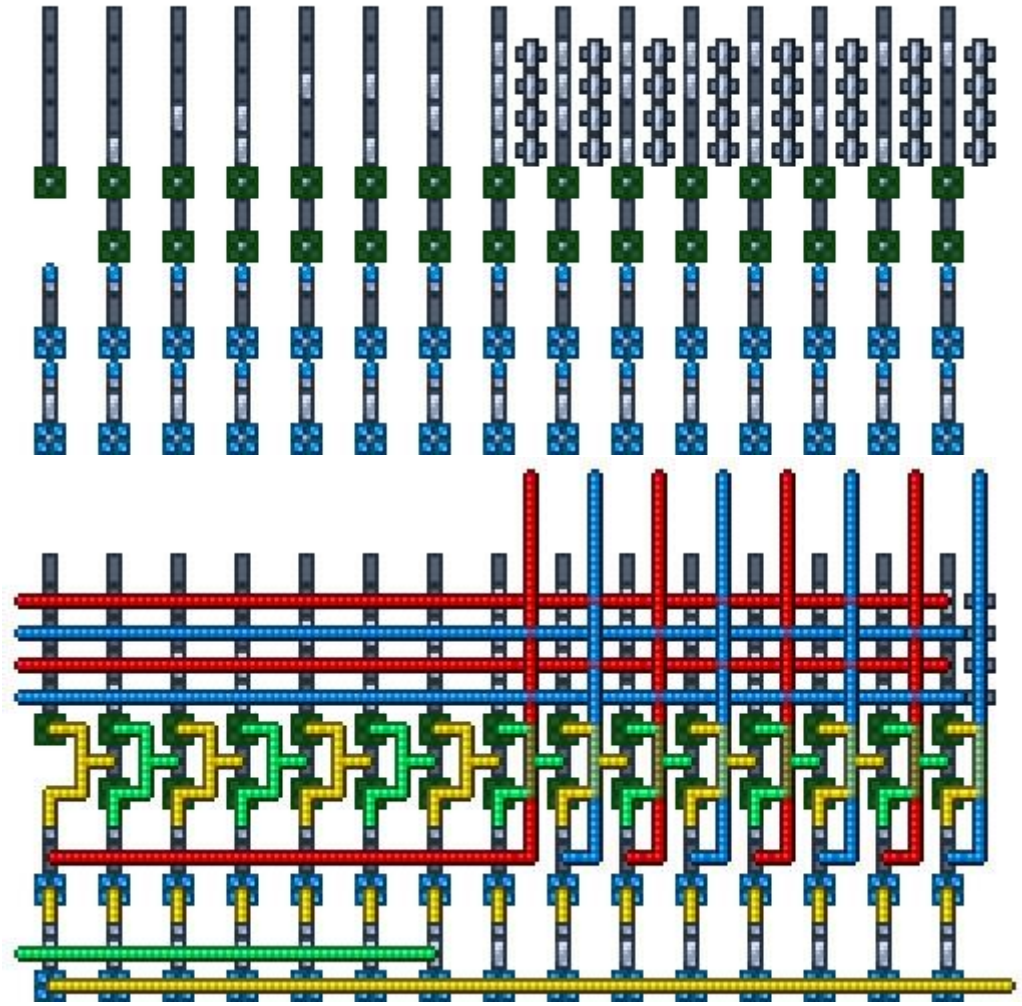


Figure **4.18:** Shifter. The vertical red and blue line enters the number of shifts to be shifted. The horizontal red and blue line enters the number of digits shifted. Activate the corresponding output through the logic light at the top. The bottom green line is used to switch logical shifts and arithmetic shifts.

It's gone.

4.3.8 Feed Conversion

Method one uses the definition $a_{n:0}$ directly $= a_n 2^n + a_{n-1} 2^{n-1} + \cdots + a_0 2^0$。 To convert a binary to a decimal using this formula, you need to make a decimal adder and then store the decimal values of the powers of 2. https: www.bilibili.com/video/av40474377/ is the way to use it.

The decimal turn binary can use the same principle to make a binary adder, and then store the binary values of the powers of 10.

Method 2 Uses the methods described in sections C.4 and C.5. In each step of the vertical, we need to do a check table operation and then the wrong one. The operation of this check table is actually a combination logic, as shown in Table 4.5.

Binary-to-BCD and BCD-to-binary correspond to different true value tables, and we need to design them separately.

Binary	BCD
0000	0000
0001	0001
0010	0010
0011	0011
0100	0100
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100

Table **4.5:** Binary-to-BCD checks from left to right, andBCD-to-binary checks from right to left.

4.4 Storage circuit

A storage circuit always has an array for data and a multiple selector for selecting data.

4.4.1 Multiplexer

The multiple selector receives a binary address that activates the location that the address represents. A classic four-bit multiple selector is shown in Figure 4.19.

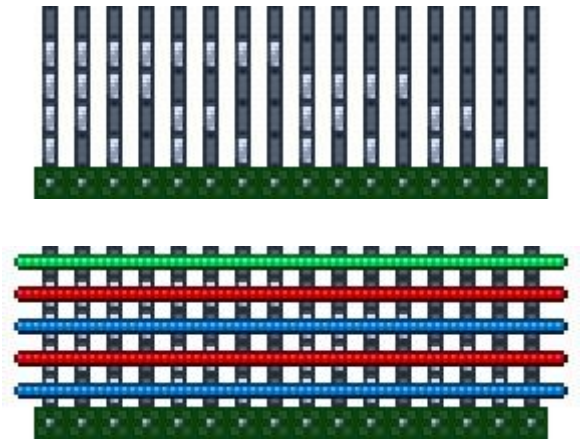


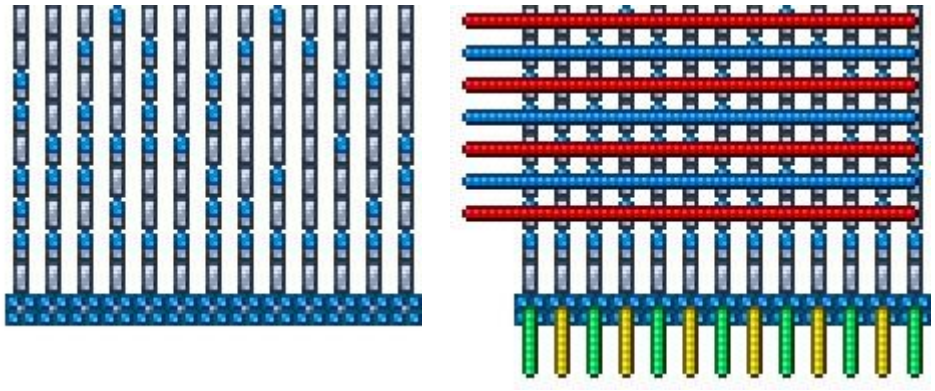
Figure **4.19:** Classic four-bit multiple selector. The red and blue lines represent the address, which is low below the high above. The green line doubles the active output. The 16 logical doors represent 0000 to 1111from left to right.

We've seen similar circuits in decimal numbers, and in fact that's the state version of the multiple-selector. In this section, we implement the activation version of the multiplex selector with dual activation technology.

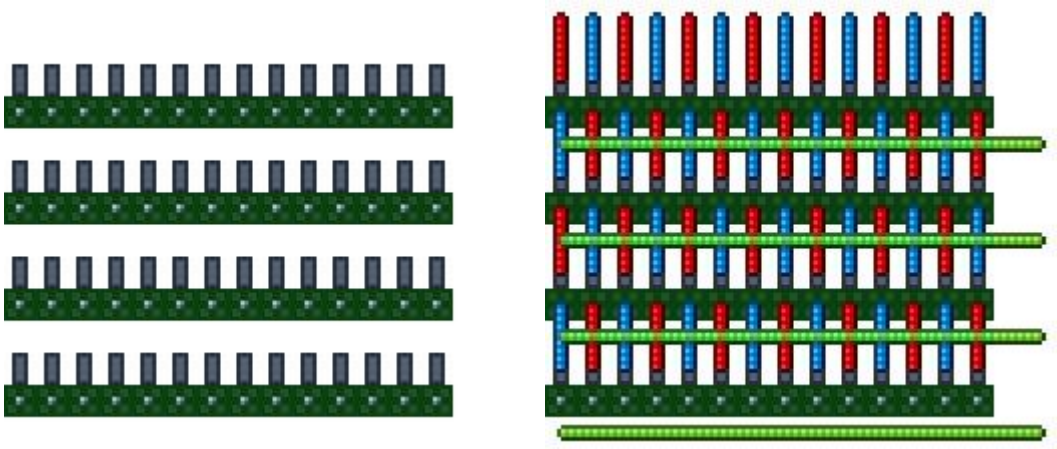
4.4.2 Read-Only Memory (**Read-Memory**).

Read-only memory is used to store pre-designed, fixed data. This data does not need to be modified at the time of use, so it is physically done directly in the circuit. Figure 4.20 shows three ROM designs. Figure 4.20 (a) Store data using fault logic lights. When a horizontal line is activated, only the logic gate under the passing fault logic light is activated. Figure 4.20 (a) is not currently of practical value because the memory is to work with the multiple selector, which is an upturned output. Figures 4.20 (b) and 4.20 (c) are both wired data storage and are two types of ROMs that are widely used. Two circuits

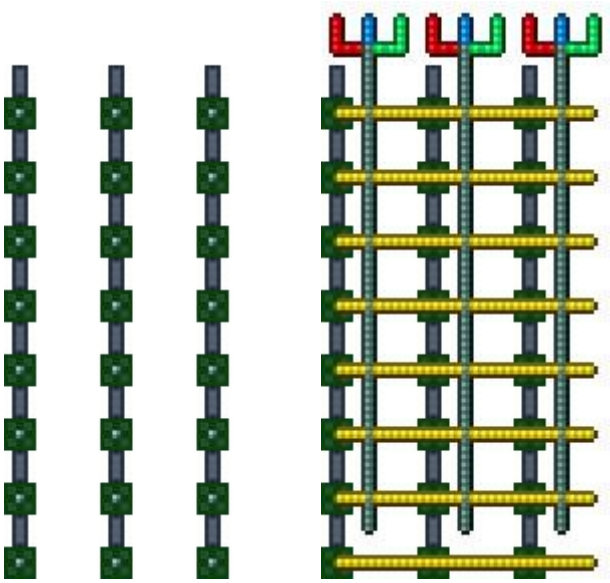
4.4 Storage circuit



(a) The landscape input is vertically output



(b) Vertical input landscape output



(c) Vertical input landscape output

Figure **4.20:** Three designs for ROMs.

The difference in how it is implemented is that Figure 4.20(b) is stored by connecting the output line to the logic gate, while Figure 4.20 (c) is stored by connecting the input line to the logic light. In terms of footprint, Figure 4.20(b) occupies a lower height (1.5 grids/bit)and Figure 4.20 (c) accounts for

With a high height(2 grids /bit). In terms of logical synchronization, the output of Figure 4.20(c) is synchronized, while Figure 4.20 (b) is out of sync and is better in scenarios that already require out-of-sync scenes (e.g. many arithmetic circuits and network wires), figure 4.20 (b).

4.5 The theory of segmented display simplification

In the decimal number display, Figure 2.16 is the state-style multiple selector, and the lower half of Figure 2.17 circuit is the ROM of Figure 4.20(b). Only displays with a limited output can be implemented through roM.

4.4.3 Write-Only Memory (Write-Memory Only).

1

4.4.4 Random Access **Memory**

1

4.4.5 Register

1

4.4.6 栈（Stack）

1

4.5 The theory of segmented display simplification

1

4.5.1 Segmented display main structure

1

4.5.2 Displays the matrix, segmented matrix, and number matrix

1

The primary transformation of the 4.5.3 matrix

1

4.5.4 Simplification principles and details

1

4.6 Processor structure

1

4.6 Processor structure

4.6.1 Assembly language

1

4.6.2 Machine language

1

The topology of the 4.6.3 processor

1

4.6.4 Data **Path**

1

Chapter 5 Circuit Items Are Detailed

The

5.1 Switch / Joystick

(a) Switch (b) Lever

Figure 5.1

The activation condition of the switch and lever is a right-click. As with other interactive items /NPC, you can only right-click when the switch/lever is within reach.

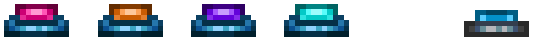
Switches are generally used when the circuit is dense due to their smaller size compared to the lever. On the other hand, the disadvantage of small volume is that switches are not easy to find and easy to point wrong.

The switch can be placed on the side of the wooden beam and on the side of the solid block and the lever cannot be placed on a flat surface and the switch cannot (Figure 5.2). It is important to note that the lever can only be placed on the crucible if there is a sufficient area of the background wall on the crucible, and the background wall will not fall after it has been placed on the pickaxe, which may be a determined bug.

Figure 5.2: The lever placed on a flat surface

5.2 Pressure plate

(a) Grey / Brown / Blue / Jungle Lizard Pressure Plate (b) Red / Green Pressure Plate (c) Yellow Pressure Plate (d) Orange Pressure Plate



(e) Increase pressure plate (f) turquoise pressure pad Figure 5.3: Pressure plate

Pressure plate is divided into ordinary pressure plate, increase pressure plate, turquoise pressure pad.

5.2 Pressure plate

Normal pressure plates include player-triggered gray/brown/blue/orange/jungle lizard pressure plates, NPC-triggered yellow pressure plates, and player-triggered red/green pressure plates. The pressure plate is mistranslated and the correct translation should be "gravity pressure plate". The four colors of the pressure plate function exactly the same.

Note Methods for distinguishing between ordinary pressure plates of different colors: the player can only trigger a variety of traps in the map, the color is very hidden, such as the gray pressure plate on the stone, the brown pressure plate on the earth, the blue pressure plate on the ice block, the jungle lizard pressure plate on the jungle lizard brick is very illegible; Red/yellow/green pressure plates are very eye-catching and can be triggered by NPC.

The ordinary pressure plate has its own collision box, and the size of the collision box is the border size of the map when the pressure plate bounces. The decision of normal pressure plate activation is based on each collision chamber, i.e. each frame determines whether the impact chamber enters the pressure plate from the side or drops from above onto the pressure plate. Because the collision box is based on the collision box, a collision box triggers two normal pressure plates in a frame, activating only once, while different collision boxes trigger the same normal pressure plate within a frame, each of which is activated once (Figure 5.4). Note that "entry" or "drop" is a process, so direct transmission to a normal pressure plate does not trigger the pressure plate. At the same time,"drop" requires the character to have a suspension process, which determines that the suspension can be through the character's action (jumping action when suspended) or wings (the winged character opens when the wing is suspended). So walking straight down from [9]the block is not a drop, and the collision box is not coming in from the side, so it does not trigger the normal pressure plate (Figure 5.5).

(a) (b)

Figure 5.4: (a) Step on two red pressure plates at the same time, with only the torch response on the left;

[9] The judgment of the other case is very complex and will be in 传 The drop-off sectionTalked about。

NPC also drops onto the red pressure plate and the torch is still on.

(a) (b)

Figure **5.5:** (a) Go straight down from a grid height, wings do not open, no jumping action, the pressure plate is not triggered;
"Down" arrow key, wings open, jump action, pressure plate will be triggered.

The pressure plate does not have a collision box, which determines whether 1 person is in the grid where the pressure plate is located per frame, whichever is based on the pressure plate itself.

5.3 Timer

The collision box for the turquoise pressure pad is 16 x 10 or 10 x 16,depending on its orientation. After the projectile is generated, each update position activates the collided turquoise pressure pad, which is defined as the collision box that did not intersect at the time of the last update, but intersects at this update. If the projectile collides with multiple turquoise pressure pads at the same time, only the one with the highest priority (the green pressure pad on different rows, the highest priority above; the green pressure pad on the same line, high priority on the left); and multiple projectiles collide with the same turquoise pressure pad at the same time, each projectile is activated once (Figure 5.6).

(a) (b)

Figure **5.6:** (a) Cannon fire, only the torch response on the left; (b) activate the left or right switch torch to respond, activate the middle switch torch does not respond (actually twice).

Ordinary pressure plates and pressure plates can be placed on platforms and ingots and turquoise pressure pads cannot be, turquoise pressure pads can be placed on the sides and below solid blocks and ordinary pressure plates and accent pressure plates cannot.

The pressure plate track is a track with a pressure plate that can only be triggered by a minecart.

5.3 Timer

Figure **5.7:** Timer

The timer is affected by the cooldown time mechanism, but when the countdown reaches 0, the countdown is reset instead of being deleted. [10] When you use the right button or circuit to activate the shutdown timer, the timer is not removed directly from the cooling list, but is left to be deleted the next time it is refreshed to the timer.

5.3.1 Series timer

Figure **5.8:** The torch is activated 8 seconds after the red line is activated. Turn on the 5-second timer when the red line is activated, turn on the 3-second timer after5-second, 5-second timer activation, turn on the 3-second timer after 3-second, 3-second timer activation, turn off the 5-second timer, activate the torch, and turn itself off through the wire changer.

Connecting timers at different times gives you the sum of their times (Figure 5.8). How about connecting timers at the same time?

5.4 Detonator

Figure **5.9:** The left red line is activated after 5 seconds of torch activation. Turn on the left one 1 second timer when the red line is activated, turn on the left 2 1 second timer after 1 second, turn on the left 1 second timer after 1 second, the left 2 1 second timer (before the left 1 second timer), turn off the left 1 second timer, turn on the left 3 1 second timer, and so on. Turn off the right 2 1 second timer when the right 1 second timer is activated, activate the torch, and turn itself off through the changer.

Because adding a cooling list is front-to-back, and refreshing a cooling list is backward-forward, this results in a timer that adds a cooling list later to activate first. So timers at the same time can also get the sum of their time when connected together (Figure 5.9). Flexible combination of this technology and frequency reduction technology can get a timer that takes up a fairly small footprint, with a period of 1/4 second integer times.

[10] In fact, there are differences in program handling, and the argument here is only easy to understand without changing the characteristics。

5.4 Detonator

Figure 5.10: Detonator

The detonator is a relatively special item, which has two states of pressure and bounce. Right-clicking or the character passing two grids on the detonator at a vertical speed of at least 3 pixels/frame causes the detonator to press down and activate as a power source, and then after 1 second, the detonator automatically bounces. For determining reasons, the detonator is also pressed down when the character falls at a certain speed (neither fast nor slow) on the edge of the support next to the detonator (Figure 5.11). A right-click or a character stampede while the detonator is under pressure is invalid (Sage mode).

(a) (b)

Figure 5.11: (a) Jump in place at the edge of the platform, and when the jump height is within a certain range, the detonator is pressed down;

The detonator is also an electrical appliance that switches between pressing and bouncing when activated. If the detonator is activated under pressure, it will not be activated as a power source and will not automatically bounce until it is activated again.

Detonators can be placed on almost any flat surface. Unlike the joystick, the detonator is not hanging furniture, so it cannot be placed on a pickaxe.

5.5 Trapped Chest / Dead Man's Chest

(a) (b)

(a) Non-themed trapped chests. The subject is trapped chest see Appendix the treasure chest of the dead

5.5 Trapped Chest / Dead Man's Chest

Trapped chests are mistranslated, and the correct translation should be "organ chests" or "trap chests". Trapped chests can be dug up directly by pickaxes, thus distinguishing them

from ordinary chests. The dead's chest is more hidden than the trapped chest, because it can contain items. Without wires attached, the dead man's chest and the ordinary gold box are completely indentable.

Trapped chests and dead chests are power supplies that activate when the mouse right-clicks. Trapped chests can't hold anything, right-clicking just plays the box open and close animation, and the dead treasure chest is no different from the gold box except that it will be activated.

The chest of the dead is "right-click" activation instead of "on/concern change" activation. Automatic closing of the chest caused by the character's distance does not activate the dead man's chest.

5.6 Sensor

(a) Logic sensor (day/night) **(b)** logic sensor (player) **(c)** liquid sensor Figure **5.12:** Sensor

There are 3 logic sensors and 4 liquid sensors in Tyralria. The logic sensor (day) lights up during the day and goes out at night, the logic sensor (night) goes out during the day and lights up at night, and the logic sensor (player) is placed with a blue box with a circuit layer that is slightly wider than 5 squares wide and slightly larger than 10 blocks higher, and lights up when there is a player in the box, and turns off when there is no player in the box. The liquid sensor lights up when there is a corresponding liquid in the grid and goes out when there is no corresponding liquid.

The logic sensor (day) and logic sensor (night) are activated when lit by off, and the other sensors are activated when the switch is off.

The logic sensor (player) can be thought of as an accent pressure plate with a wider range of sensing, but insensitive to transmission and only judgment per frame. The activation signal of the logic sensor (player) and the activation signal generated by the logical settlement triggered by that signal do not trigger the transmitter.

All sensors are free to be placed without support blocks or background walls.

Sensors that use the map editor or Mod copy-paste do not work and need to be removed and placed manually. Therefore, the circuit should be designed to avoid the use of sensors on a large scale.

5.7 Partial light source

5.7 Partial light source

The light source here does not refer to a valid light source, but to all objects that can glow. All light sources that can be controlled by the circuit are listed below.

(a) Torch (b) candle (c) Plasma Lamp (d) 灯笼 (e) 烛台 (f) Lamppost (g) Bonfire (h) fireplace (i) Chinese lantern (j) chandelier (k) disco lamp (l) Volcano (m) Crystal Gem Block (n) Christmas Lights Figure 5.13: Light source in circuit. Light sources in themed furniture are listed in the Appendix.

- Torch: Size 1 x 1,placed on the platform, above and on both sides, background wall. is a
- valid light source.

Candles: Size 1 x 1,placed on a flat surface other than a pickaxe, can be placed on the ingot but will drop immediately. Water candles and peace candles do not provide gain when extinguished. is a valid light source.

- Lantern: Size 1 x 2,suspended under solid blocks. 3 worm bottles,6 soul bottles without circuit function. Star bottles and hearts do not provide gain when extinguished. is a valid light source.
- Lamp: size 1 x 3,placed on platform, ingot, solid block. is a valid light source.
- Bonfire: size 3 x 2,placed on platform, ingot, solid block. No gain is provided when off. is a valid light source.

- Candlestick: Size 2 x 2,placed on a flat surface, solid block other than a pickaxe. is a valid light source.
- Chandelier: Size 3 x 3,suspended under solid blocks. is a valid light source.
- Crystal Gem Block: Belongs to a solid block. Is not a valid light source.
- Chinese lanterns: size 2 x 2,suspended under solid blocks. is a valid light source.

Lamp post: size 1 x 6,placed on platform, ingot, solid block. Is not a valid light source.

5.8 doors / door / high door

- Disco lights: size 2 x 2,suspended under solid blocks. Is not a valid light source.
- Christmas lights: Size 1 x 1,placed around the solid block. is a valid light source.
- Fireplace: Size 3 x 2,placed on platform, ingot, solid block. is a valid light source.
- Plasma Lamp:Size 2 x 2,placed on a flat surface, solid block other than a pickaxe. Is not a valid light source.

: Volcanois 1x1 and 2x2, respectively, and is placed on flat surfaces and solid blocks other than pickaxes. Is not a valid light source.

Here's an emphasis on two commonly used display light sources: gem blocks and torches. In general, gem blocks appear better, and their lighting out appears in small maps. However, since torch activation is only a simple state change, and each gem block is activated also needs to update the block and the surrounding 8 pieces of maps, each update of the map requires a lot of judgment, which results in large-scale use of gem block circuits than the use of torch circuits more card.

5.8 doors / door / high door

(a) Non-subject doors. The subject is shown in the Appendix. (b)

Organ Doors (c) High Gate Figure 5.14

Doors / door / high door is both a power supply and an electrical appliance.

Locked jungle lizard doors have no circuit function (nonsense). The door is placed between the upper and lower solid blocks. The door does not open to the door when it appears within the opening range(1 x 3)on one side of the door, and the door does not open when it appears within the open range on both sides of the door. When the door can be opened to both sides, if you right-click to open the door, the door opens in the direction facing the player, and if the circuit opens, the door appears to open randomly to both sides.

Compared with the door, the opening direction of the door is up and down. When the door can be opened to both sides, use the right-click door to open the door, according to the

player and the relative position of the door to determine the direction of opening the door: the player at a relatively high point when the door opens down; Open the door using the circuit and secure it down.

High doors have no direction and open doors are unobstised.

When the circuit switch door is not used, the door is activated as a power source. If you open the automatic switch door in the settings, the door acts as a player sensor.

5.9 Fog Machine

图 **5.15:** Fog Machine

Fog Machine is an electrical appliance that is switched on/off when activated. When Fog Machine is turned on, a white dynamic cloud is generated in a narrow landscape area. This cloud must be turned on in the Settings - video to open the Blood Splash effect to be visible. In the setting

5.10 Toilet

Set - Turn lighting mode to Color in the video to make the clouds more visible.

5.10 Toilet

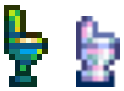


Figure **5.16:** Non-themed toilet. See appendix for the theme toilet.

The toilet is made of electrical appliances. When the toilet is activated, it is shown in Figure 5.17. The toilet has a cooling time of 60 frames.

Figure **5.17:** Many toilets are connected to the driving scene. Photo Source: Sleeping

5.11 Pump

(a) Incoming pump **(b)** outlet pump Figure **5.18:** Pump

Connect an incoming pump and an outlet pump with a wire, and when the wire is activated, the liquid on the incoming pump will transfer as much as possible to the outlet pump. Virtualize the solid blocks under the pump and the pump will not break down.

To understand the liquid transfer settlement in some strange cases, the internal operation mechanism of the pump is described here.

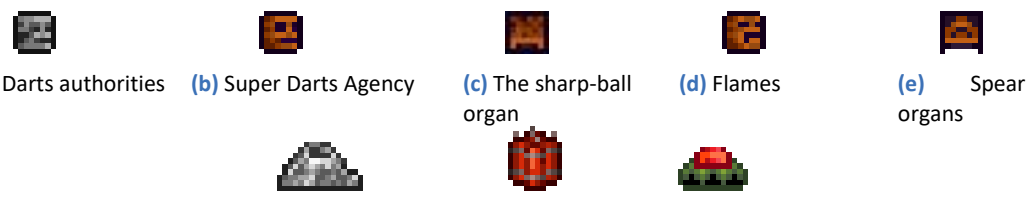
The game uses two 19-length lists to store the coordinates of the incoming and out-of-water pumps, respectively. In this list, each pump is a single grid. The pump in the game contains four tugs, which are added to the list in turn when the pump is activated, in order of bottom left - bottom right - top left - top left - top right. Add to the list when it is full and do not continue to join.

Pump settlement is carried out after each wire is settled, the list of incoming pumps is scanned from the end of the trip (except for incoming pumps without liquid), for each incoming pump, the list of pumps is scanned from front to back (except for pumps full of liquid), and the liquid in and out of the pump is transferred.

The liquid transfer between the single-entry pump and the single-grid outlet pump must first follow the principle of liquid consistency, that is, the transfer of liquid will not cause different liquids to appear on the outlet pump. Second, the amount transferred is the minimum amount of liquid on the incoming pump and the amount of residual liquid over the outlet pump (0 to 255fluids percell).

5.12 Organs

5.12 Organs



(a) Darts authorities **(b)** Super Darts Agency **(c)** The sharp-ball organ **(d)** Flames **(e)** Spear organs

(f) Fountains (organs) **(g)** explosives **(h)** mines Figure **5.19:** Organs

The organ referred to here refers to the activation of electrical appliances that will cause harm to the player.

Super darts organs, sharp ball organs, flame organs, spear organs are generated in the jungle lizard temple. Dart organs are generated in other locations underground. These 5 organs are solid blocks, and hammering can change the direction of shooting;

The projectiles of the darts organ, the super dart organ, the sharp ball organ and the flame organ are generated in the 2nd block in front of the organ, so a solid block close to the front of the four organs does not block the projectile. A solid block that clings to the front of the spear organ blocks the spear.

Darts travel in a vacuum at a speed of 45 grids per second and a lifetime of 60 seconds. The cooling time for darts, super darts and flame organs is 200 frames.

The lance organ has a range of 19.5 grid 3and a cooldown of 90 frames.

The flame organ has a range of 20 frames and a cooldown of 200 frames. Despite the wide special effects of the flame organ, its projectiles are generated only in front of the organ.

The sharpball organ cools at 300 frames with a generation limit. Its restriction rules are more complex, see wikis.

Fountains (organs) cannot be found in Chinese wikis for translation reasons because of conflicts with fountains (decorations). Check out the Geyser entry in the English wiki.

A fountain (organ) can be placed on or down a solid block, and its orientation is divided into up and down depending on the placement. Cooldown 200 frames. The turquoise pressure pad is not triggered. The range of 20 blocks starts with the first 2x4 open space (no solid blocks and liquids) encountered in the injection direction of 29 blocks and is then blocked by obstacles.

The explosive explodes when activated, with an explosion radius of 10 d.c. Explosives are disposable and therefore have limited application in circuits. Use the map editor or Mod to place explosives under a chest that can be detonated indefinitely because the objects under the chest are invincible.

The mine explodes when activated. The difference with explosives is that mines do not damage the tugger and do less damage. On the other hand, mines can explode when trampled by a player or NPC, which allows the detonation of mines to be made completely invisible without wires, so they can be painted completely invisible.

5.13 Turret

(a) Cannon

(b) Rabbit Gun

(c) Colored paper cannon

(d) transmission gun station

(e) snowball launcher

Figure **5.20:** Turret

5.13 Turret

The turrets include cannons, rabbit guns, colored paper cannons, delivery gun stations and snowball launchers.

The turret size is 4 x 3,but can be placed on two-block-wide solid blocks, platforms, or ingots. The turret can use the conveyor float 5.

The left/right part of the right-hand turret can be turned and the corresponding shell can be fired with the corresponding shell left- and left-click. When activated with wires, different positions of the activation turret have different effects (Figure 5.21). Use wires to activate the projectile fired without injury. Manual activation of fired projectiles causes damage without invincibility frame 6.

The position of the barrel to generate the projectile is across the intersection of the two columns in the middle, and the ordinate coordinates are at the junction of the lower grid and the middle grid. For the conveyor, the build position is also moved down by 5 pixels, and if the conveyor is vertically up, the build position is also moved 5 pixels to the right.

The turret generates the initial speed of the projectile, with the conveyor station at 93 pixels / frame 7and theother 14 pixels / frame. The initial speed direction is only related to the turret orientation, except for the special angles of 0,π/2,π/4, which are arctan (1/3) and arctan (3) instead of π/8 and 3π/8.

After the artillery and rabbit gun shells are fired, the first 17 frames of speed remain the same, starting from frame 18, the vertical speed per frame increases by 0.28 pixels per frame (up to 16 pixels / frame), the lateral speed multiplied by 0.99. The projectile of the conveyor station moves in a straight line at a constant speed and activates all turquoise pressure pads on the path in sequence. The projectile fired by the color paper gun runs in a straight line, with a lifetime of 2 frames, and explodes at the end of the lifetime to produce special effects. In these 2 frames, the projectile can only advance by 28 pixels, cannot leave the size of the turret 4x3, and cannot trigger the turquoise pressure pad.

The shells of the Rabbit Cannon bounce on the solid block. When a horizontally fired rabbit shell stops on a horizontal solid block plane at the same height, it is 93 blocks 8from the gun.

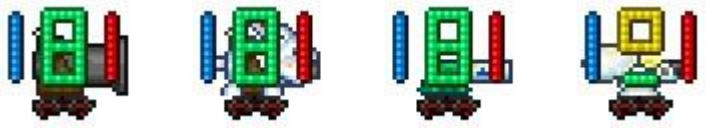


Figure **5.21:** The red line turns right, the blue line turns left, the green line fires, and the yellow line changes the projectile color.

The mouse operation of the conveyor station is different from that of other turrets. The mouse right-clicking gun station has the same effect in different positions as the wire activation.

3 Verified by @dcfhft.

4 The fountain (Geyser) is a natural landscape, the fountain is an artificial landscape, the official does not distinguish between translations.

5 Verified by @WIAADC

6 verified by @WIAADC

7 3 pixels forward each time, 31 times per frame

8 Verified by @WIAADC

5.13 Turret

Projectiles fired from turrets other than colored paper cannons can trigger turquoise pressure pads. Cooldown 30 frames.

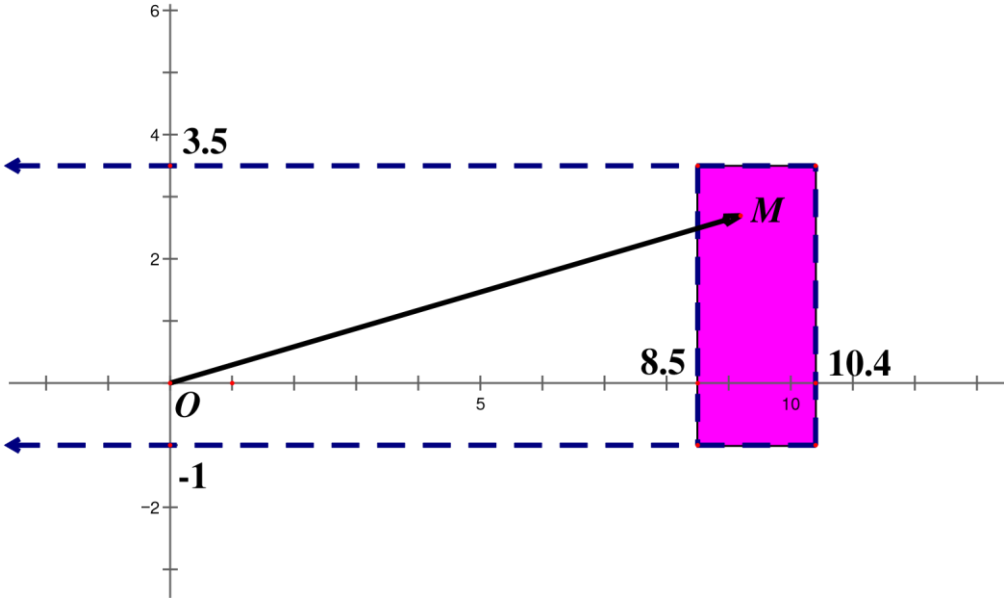


Figure **5.22:** Direction of launch of the snowball transmitter. O is the emission point, and the OM is the direction of emission if a little Mis randomly taken withinthe rectangle shown in the figure.

Snowball transmitters are special and have almost identical characteristics to those described above. Snowball transmitters are 3 x 3 insizeand can only be placed on solid blocks, platforms, or ingots three blocks wide. Snowball launchers can not be manually turned, snowballs in the backpack when the right-click can fire snowballs, can be sent in succession. The snowballs emitted trigger the turquoise pressure pads. Activating one of its left three blocks with a circuit causes it to face left, activating one of its right three blocks makes it face to the right, and activating one of the middle three blocks emits an unsophageable snowball. There are only two types of snowball launchers facing: horizontally to the left and horizontally to the right. The launch point is 12 pixels to the left (if facing left) or 12 pixels to the right (if

right) in the center of the snowball launcher; the launch speed is random in the center of the snowball launcher (see Figure 5.22). After the snowball is emitted, the first 19 frames of speed remain the same, starting at frame 20, increasing the vertical speed per frame by 0.3(up to 16), and the lateral speed by 0.98. Cooldown 10 frames.

5.13.1 Turret control circuit

Although the multiple interfaces of the turret have allowed us to operate at will, we are not satisfied, we hope to be able to turn into place at once, rather than a few turns. How many times does it take to activate a turret from one angle to another? Obviously we need to know what angle is before turning, what angle it is after turning, and then subtract the two angles.

It's too complicated to think so. In fact, we just need to know what angle it is after turning, not what angle it is before turning. This is because, unlike other appliances that switch between different states when activated, the steering of the turret is limited. For example, when the turret goes to the far left, if you activate the left side, the turret does not respond, not to the far right. This allows us to reset the turret 8 times by activating the turret on one side and then calculate the number of activations for the second turn at the angle of rotation. Using this approach, we designed a turret control circuit that can be turned and fired at the click of a button (Figure 5.23).

To understand this circuit, you need to understand logical settlement. When a switch is activated, two lines on it are activated: the red line and the blue/green/yellow line. The red line activates the reset circuit on the left consisting of three fault logic gates, each of which activates blue-green

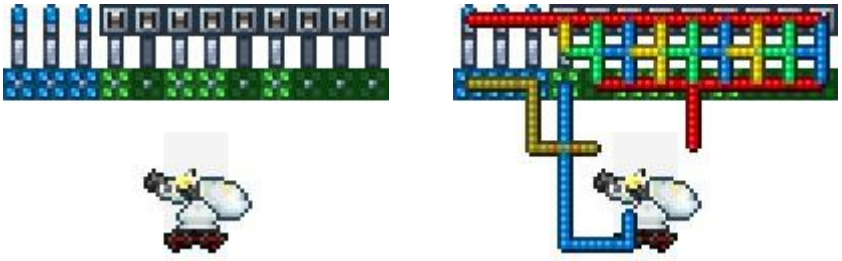


Figure **5.23:** Turret control circuit. [11]each switch corresponds to nine directions.

The yellow tricolor line, which emits a total of 9 activation signals, rotates the turret to the far left. The blue/green/yellow line activates the logic delayer below, generating a different number of signals by initially activating doors at different locations to rotate the turret to the right to the desired position. The tail end of the logic delayer outputs a signal to fire a projectile. In general, each switch is activated in three steps: reset, steer, and transmit. Where the order of reset and steering is controlled by the color of the wires on the switch, because the red lines settle first, and the order of steering and firing is controlled by logical delay.

5.14 Conveyor

Figure **5.24:** Conveyor

A conveyor is divided into three tugs. The conveyor is a solid block that can be knocked into half a brick. In the working mechanism of the conveyor, the three diagrams have their own transmission areas (Figure 5.25).

Figure **5.25:** The transmission area of the three tugs. If the Tug is knocked into the lower half of the brick, the transfer area moves down half a block and the other half-brick shape transfer area remains unchanged.

When a wire is activated, record that the conveyor Tug (hereafter A)under the wire is settled with thelast settled conveyor Tug (hereafter referred to as B)9,and then swap the transferable targets within the two grid transfer areas, their speed remains the same after the swap, and their position remains the same relative to the transfer area. Cannot be delivered when the transfer area of the two graphs coincides and the transfer area of A is higher than the delivery area of B, and when the delivery area of A and B coincides and A is below

[11] See the order of settlement on the same wireLogical settlement

B, where the overlapping portions of the two transfer areas belong to the delivery area of A (Figure 5.26).

In order to be fluent, the above description omits a key process. If A is not the lower half brick, the other two graphs of the conveyor are ignored in subsequent settlements to avoid self-transfer. Conversely, if A is the lower half brick,

Then it will be possible to implement its own transmission (**?** . ） 。

Note Interchange! Interchange! Interchange! Always thinking about sending a few NCs together and copying the word 100 times!

How did the others do that? Put half a brick on the target conveyor to push the NPC away, setting a delay between several transfers.

Using the characteristics of the transmitter, we can design transmission arrays with different principles. Bidirectional 1 pass 8 takes advantage of the size of the conveyor itself; bidirectional 1 pass 20 takes advantage of the size of the transfer area; One-way multi-pass 1 takes advantage of a line passing through multiple

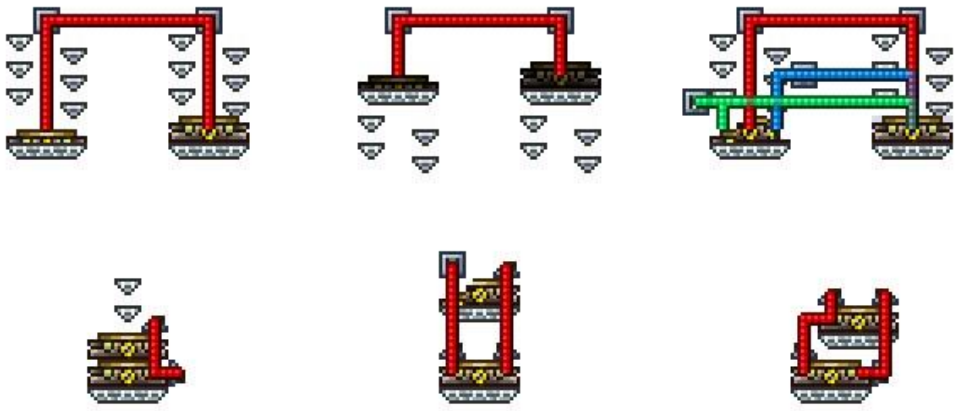


Figure **5.26:** Upper left device: The right conveyor is half a grid higher than the left conveyor delivery area, so the player is half a grid higher from the left to the right. Upper-middle device: As long as the collision box and the transmission area have a little overlap, it can be transmitted. Top right device: The three tugs of the conveyor each have a transmission area, from the left to the right, activating the green line will be half a grid higher, and activating the blue line red line will not. The following three devices: activate the above switch will not transmit, activate the following switch will be transmitted, when the transfer as long as the player in the transmission area of the lower conveyor, will be transmitted from the bottom to the top, the transfer time only the player in the transmission area of the upper conveyor and not in the transmission area of the lower conveyor, will be uploaded from to the bottom.

rules for the first and last settlement transfers when selecting the conveyor.

5.14.1 Bidirectional 1 Pass 8

The size of the conveyor is 3 x 1,so two wires of the same color can be picked up on a conveyor, so it is easy to make a two-way 1 pass 8(Figure 5.27).

Figure **5.27:** The grid on the left of the conveyor can be connected to four different conveyors with four-color wires, and the right-hand grid can be connected to four different conveyors with four-color wires.

5.14.2 Bidirectional 1 Pass 20

The transfer area on a conveyor is 3 x 3in size, so the transfer area of multiple conveyors may overlap and can be delivered by multiple conveyors when the player is standing in the overlapping area, increasing the number of delivery targets (Figure 5.28).

5.14.3 One-way multi-pass 1

When a line connects multiple conveyor grids, only two of them are transmitted between them. As for which two grids are transferred, it depends on the order of settlement on the wire, i.e. on the bit where the wire is activated

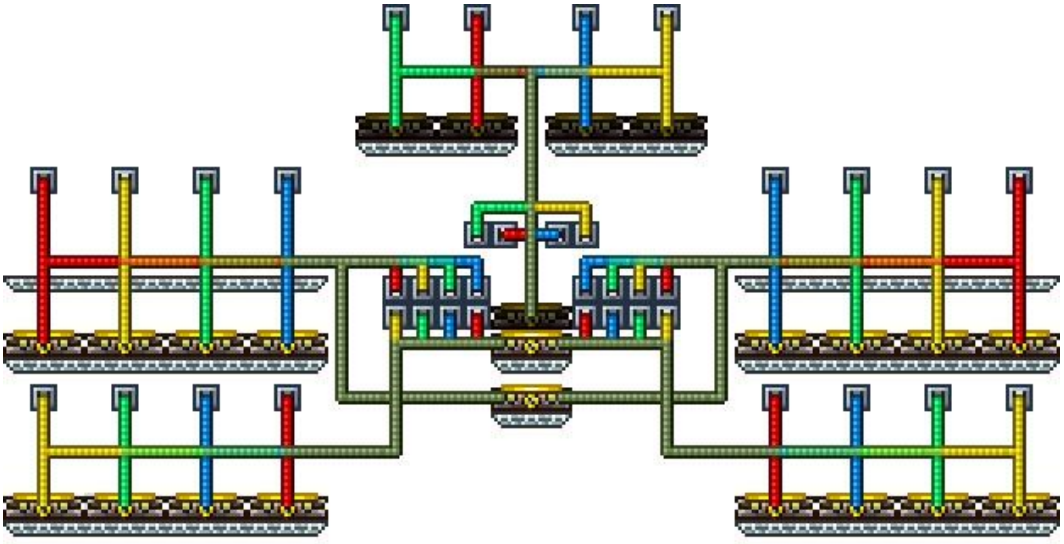


Figure **5.28:** The delivery area of the three conveyors overlaps in three grids and can be transferred to a total of 20 other conveyors when the player is standing in the overlapping area.

(see Section 5.14for details of the specific rules). This feature allows you to activate different locations on a line for the purpose of transmitting between specified conveyors (Figure 5.29).

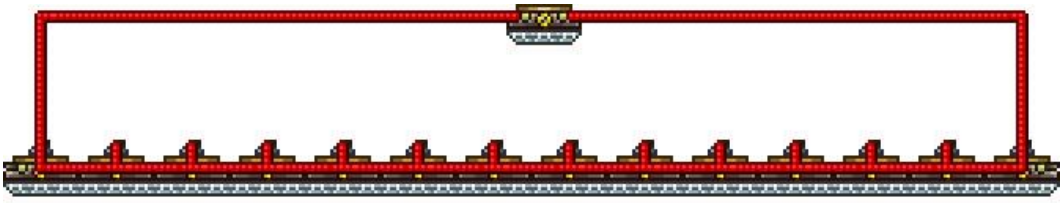
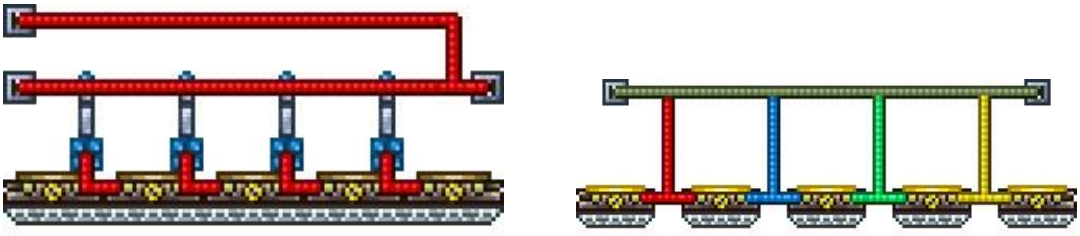


Figure **5.29:** One-way multi-pass one. Regardless of which conveyor switch is activated below, the first settled conveyor tug is the one directly below the switch, and the last settled conveyor is the left or right grid of the upper conveyor. The switch is therefore transmitted between the upper conveyor and the corresponding conveyor below when the switch is activated.

5.14.4 Transit

In Logical Settlement, we describe the order in which circuits are settled in various cases. Here we will use these settlement orders to construct more complex and powerful delivery arrays. Figure 5.30 shows two examples of using settlement order as a transit. Multi-level transit enables very complex delivery capabilities.



(a) (b)

Figure **5.30:** (a) The character stands on the left conveyor, right-clicks the left-down switch, the four logic gates are activated from left to right, the characters are transferred to the far right in turn;

5.14.5 Transfer chain

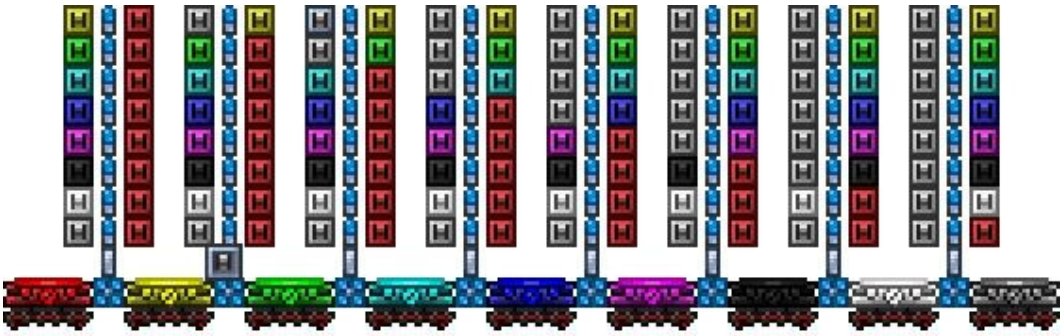


Figure **5.31:** 9- Transfer chain. Standing on each conveyor, operating the above two switches, you can transfer to the conveyor of the same color as the switch.

The conveyor chain is the result of applying Figure 5.30 (a), as shown in Figure 5.31. The transfer chain enables any number of conveyors to transmit to each other. In practice, if you don't need to transfer between two conveyors, you don't need to put so many horizontal wires.

5.14.6 Near - **Far** Relay Technology

operate	Up, bottom
initial	A B C
Transfers down	A C B
Transfer up	C A B
Transfers down	C B A

(a) Circuit. The transfer exchanges the principles of the player **(b)** near-far relay technology on the upper and lower two conveyors.

and NPC,unchanged on the intermediate conveyor.

Figure **5.32:** Near - Far Relay Technology

Figure 5.32 (a) shows near-far relay technology. Wherein the upper and middle two conveyors are near-end, the middle and lower two conveyors are the far end, the actual application of the far-end may be very far. It works as shown in Figure 5.32(b). Near-far relay technology allows the control circuit to be concentrated at the near end, while the far-end circuit is very simple (connected by only one wire). When implementing near-far relay technology, it is important to note that the color priority of near-end conveyed lines is higher than that of far-end transmissions, for example, in Figure 5.32(a), red lines are used at the near end and blue lines are used at the far end.

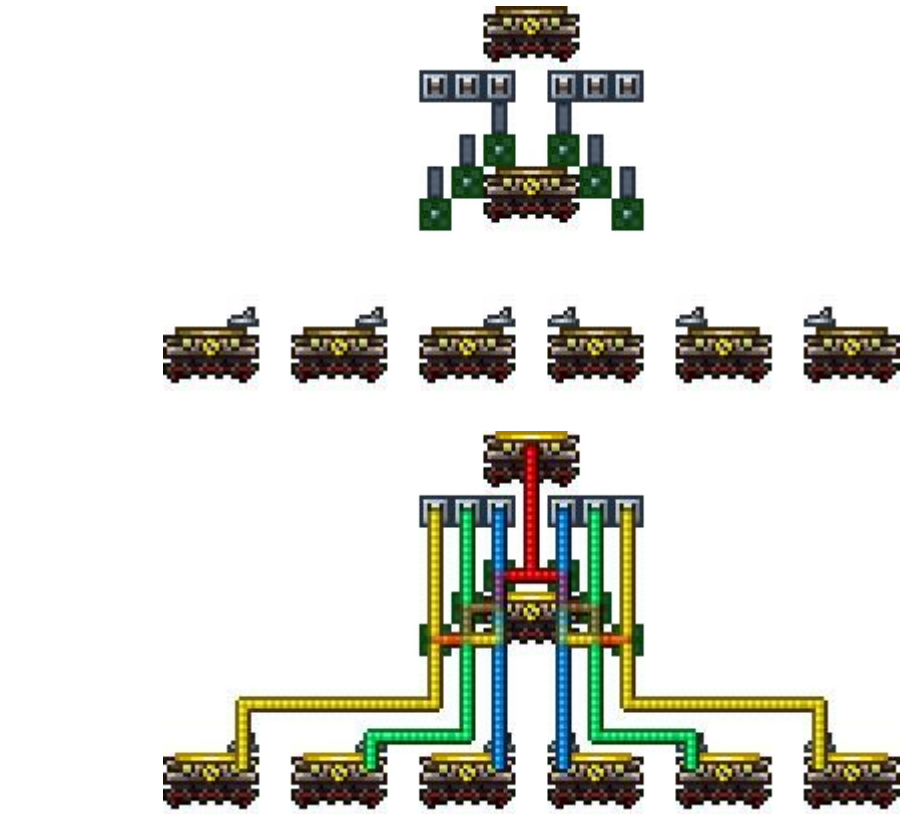


Figure **5.33:** Two-way 1 to 6 technology.

By combining the near ends of the 6 near-far trunks, you get a two-way 1-to-6 technique, as shown in Figure 5.33. Six remote conveyors can be very far away, and only one wire is connected, requiring no additional control.

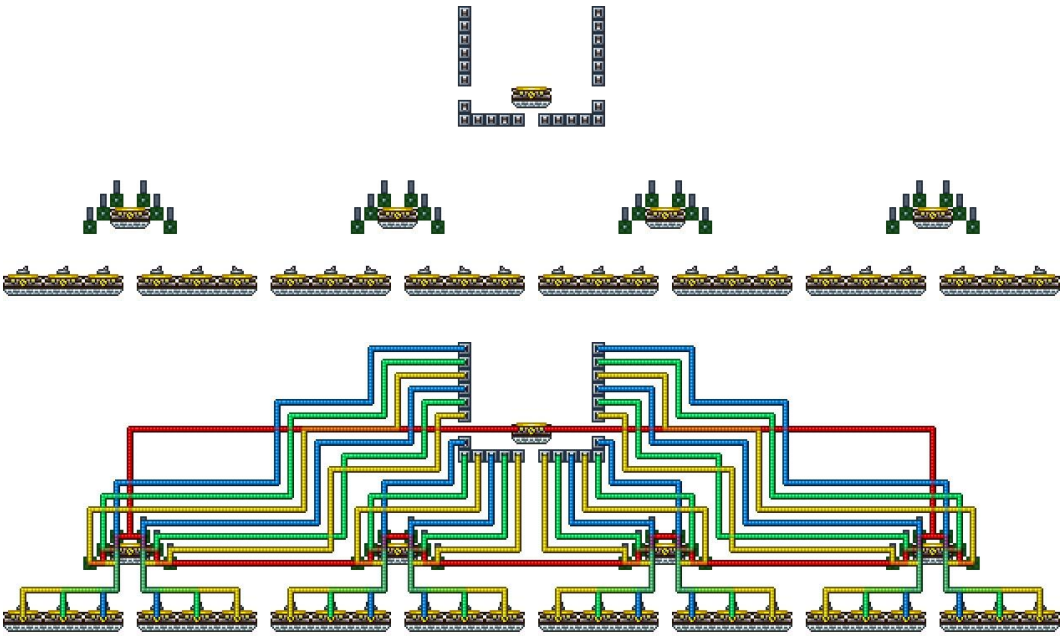


Figure **5.34:** Two-way 1 pass 24 delivery array, using 4 bidirectional 1-turn 6 and one-way multi-pass 1.

Near-end transmission can also be combined with the previously mentioned transmission array technology. For example, if you combine one-way multi-pass 1,you can make Figure 5.34.

5.14.7 One-way transmission array changes to two-way transmission array

Conveyor transfer is the exchange of players and NPCon two conveyors,in other words, the transfer is bidirectional. So why is one-way multi-pass 1 one-way? When using the conveyor array, we need to stand on a conveyor and press the switch. In normal games, players can only touch switches near themselves, but the switches in One Direction Multi-Pass 1 are on the next row of conveyors that the player cannot reach on the conveyor above. So although activating the following switch can also transfer the player from above to below, but can not be achieved through the player's own operation, it is not a two-way transmission array.

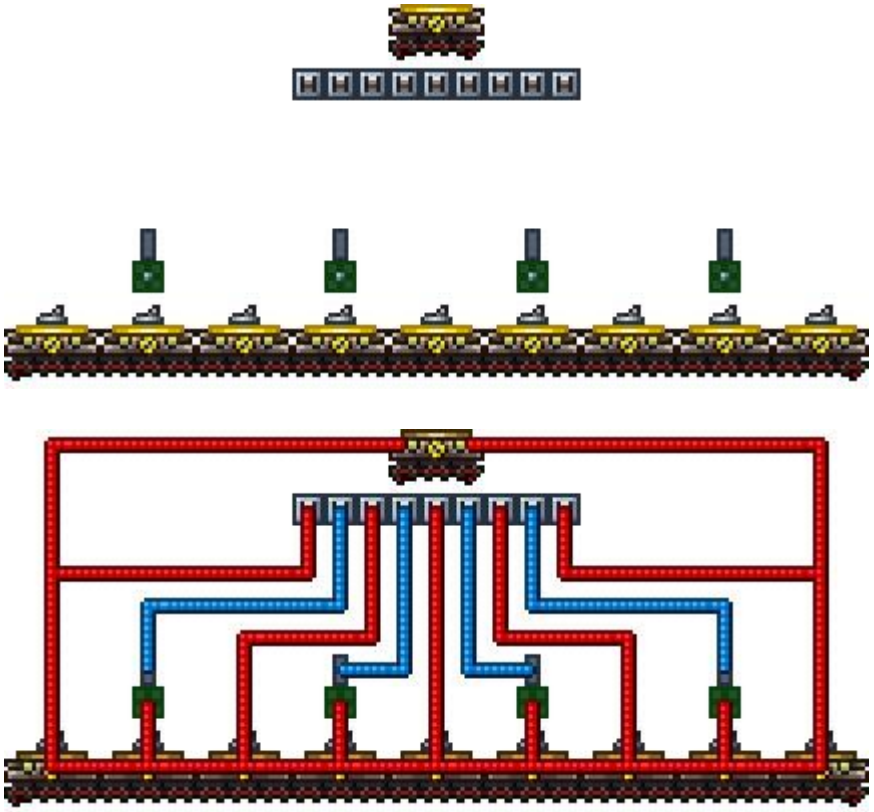


Figure **5.35:** Two-way 1 pass 9 transmission array. How many conveyors are below to connect up how many wires

One way to transform one-way transmission into two-way transmission is to allow the player to reach the switch at the far end. Figure 5.35 shows how easy it is to implement this idea. This implementation uses too many wires for large-scale long-range conveyor arrays because each remote conveyor requires a wire. Using network wire technology, the use of wires can be greatly reduced at the expense of higher computations (Figure 5.36).

The wire signal used in Figure 5.36 is different from the signal in section 3.3. In section 3.3, some logical frames of the network line are activated in a fixed eight logical frames. In

Figure 5.36, the network cable is connected to each of the doors of a horizontal logic delayer, which is activated continuously from the first logical frame: the left switch activates the network line in thefirst logical frame; ... 。 There is also a difference in the corresponding decoder, and the reader can think for himself about the decoding principle in Figure 5.36.

5.14.8 More details

Each transfer is done in four steps:

1. Transfer players in the A transfer area to the B transfer area in order in the player array.

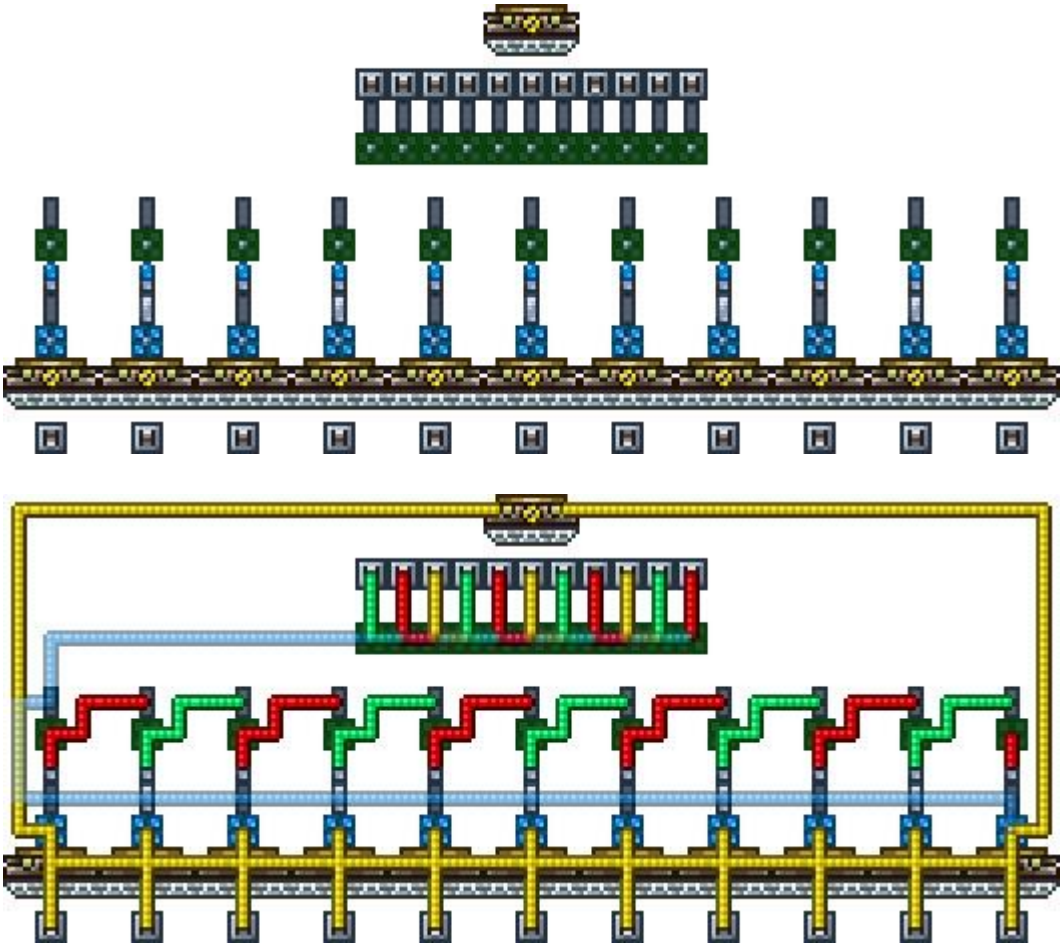


Figure **5.36:** Two-way 1 pass 11 transmission array. No matter how many conveyors there are below, only two wires are attached up.

2. Transfer the NPC in the A delivery area to the B delivery area in order in the ORDER in the NPC array.
3. Transfer players in the B transfer area to the A transfer area in order in the order in the player array.
4. Transfer the NPC in the B-delivery area to the A-delivery area in order in the ORDER in the NPC array.

When each player or NPC transfers, a teleporting property is set to true; At the end of all 4 steps, the teleporting properties of all players and NPC are restored to false,preventing players who are transferred in the first two steps and NPC from being passed back in the last two steps.

The pressure plate is checked and activated immediately after each player transfer. If the pressure plate is activated, it will

Set blockPlayerTeleport for OneIteration to true. blockPlayerTeleporta tionForOneIteration is restored to false at the end of the logical settlement. When blockPlayerTeleporta 1 and 3 are skipped directly when ForOneIteration is true.

Example **5.1** circuit as shown in Figure 5.37, the player stands in the middle of the left conveyor. If you right-click the switch on the left conveyor, two transfers will be made on the left-center conveyor, and the result will be transmitted back to the left conveyor, and if you right-click the switch on the middle conveyor, a transfer will take place on the left-center conveyor, and the result will be transmitted to the middle conveyor.

Right-click the switch on the left conveyor, left conveyor away close, so A is the left conveyor,B is the middle conveyor. First, the player is passed from A to B,activating the pressure plate. It is important to note that thetransfer step of the**AB** is still in step **1** when the pressure plate is activated! The pressure-reinforced plate will block PlayerTeleport for OneOneIteration

Figure **5.37**

Set to true and triggers a transfer from the center-right conveyor that does not transfer the player, so the player remains on the intermediate conveyor. At the end of the center-right conveyor transfer, restore the player's teleporting property to false. At the end of the logical settlement of the Blue Line, restore blockPlayerTeleportation ForOneIteration to false. At this point, continue with the delivery step of ab. In step 3, the player istransferred back to Abecause the player's teleporting property is restored to falseafter the transfer of the center-right conveyor.

Example **5.2** the circuit is still shown in Figure 5.37, with the player standing in the middle of the left conveyor. If you right-click the switch on the intermediate conveyor, a transfer is made on the left-center conveyor, and the result is passed to the intermediate conveyor.

This time, the intermediate conveyor leaves close, so A is the middle conveyor andB is the left conveyor. Step 3 of the player's AB transfer is passed from B to A,activating the pressure plate and performing a series of previous steps. Although the player's teleporting property is still restored to false,it does not continue to be transferred and the player remains on A.

Example **5.3** In Figure 5.37, cut the blue line so that it passes only through the intermediate conveyor. No matter which switch you right-click, the player is transferred to the middle conveyor.

This time, because the Blue Line does not trigger the transfer process, the player's teleporting property remains true,skipping step 3 of the AB transfer andleaving the player in the middle conveyor.

5.15 pixel box

Figure **5.38:** Pixel Box

Pixel boxes can be placed at will without support blocks or background walls. The split effect of a split box with a cross state in the pixel box.

When a power supply is activated, all wires on that power supply are activated, and if there is a horizontal wire active on the pixel box and no vertical wire is activated, the pixel box goes out, if there is both horizontal and portrait wire activation on the pixel box, then the pixel box lights up, and if there is no horizontal wire activation, the pixel box does not respond.

It is important to note that the pixel box's response is power sensitive, i.e. the signal from each power supply is settled separately, unlike the conveyor's sensitivity to wires. At the same

time, unlike the usual light source switching between light and off, the pixel box response is always adjusted to the corresponding state.

5.15.1 pixel box display

A dense matrix display up to 24 wide can be achieved using a pixel box. Because of its complex circuit, the circuit diagram is not given here, only the principle is given.

Let's first look at how to update the screen status. According to the pixel box characteristics, obviously each pixel box needs to be controlled by a line in both directions, where horizontal line activation causes the pixel box to respond, only vertical activation will not respond, so that each line of the screen is independent and can be updated line by line.

5.16 Minecart track intersection

Then let's see how to update a line. When the horizontal wires in a row are activated, the vertical wires on the pixel box that need to be lit must be activated by the same power supply, which means that each horizontal wire can only control up to 3 pixel boxes (Figure 5.39).

Figure **5.39:** Eight switches change the pixel box state to 000,001,010,011,100,101,110,111.

Fortunately, the pixel box itself has the function of a split box, so that eight horizontal wires can be arranged for each row, which is updated from the middle to both sides, each of which is a set of updates (Figure 5.40).

Figure **5.40:** When you update a row, the 12 pixel boxes on the left are updated from right to left in a group of 3 and the 12 pixel boxes on the right are updated from left to right in 3 sets.

5.16 Minecart track intersection

There must be smooth tracks in both directions on the intersection, then the two smooth tracks must be one overwriting the other, and activating the intersection changes the coverage relationship between the two smooth tracks. It is important to note that activating intersections can produce far fewer changes than using a hammer.

5.17 Other circuit items

The remaining circuit items have very little information outside the wiki and are therefore not specifically covered. They are: gem lock, statue, fireworks fountain, fireworks box,

bubble machine, budding balloon machine, party center, fountain, octave box, chimney, tower pillar, broadcast box, brake , conveyor belt, colored light bulb, growth track, golf hole, grille.

Chapter 6 more circuit topics

We've learned all the circuit principles before. In this chapter, we introduce some commonly used circuit modules in the category of circuit functions.

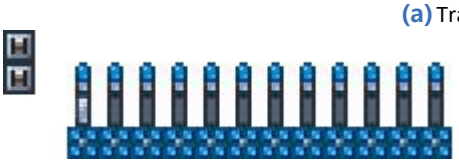
6.1 Pass circuit

In section 2.3.3, we've covered the principles of traditional pass-through circuits, and here we look at a variety of pass-through circuits.

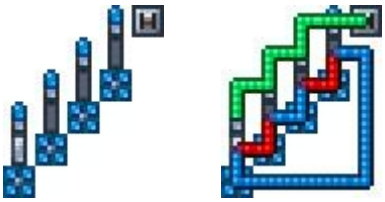
6.1.1 Traditional relay circuit

A "traditional" pass circuit is a circuit that is simply looped through multiple fault logic gates. We can see other ideas for the pass-through circuit in the later promotional and multi-level recurrings.

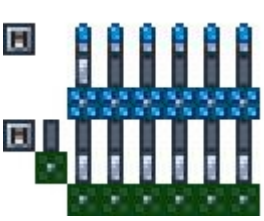
(a) Traditional delivery

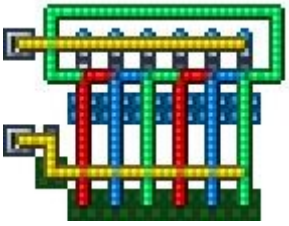


(b) Traditional pass with reset



(c) Oblique traditional recurring **(d)** dense traditional recurring





(e) The dense strip resets the traditional recidive

Figure **6.1:** Commonly used traditional relay circuits

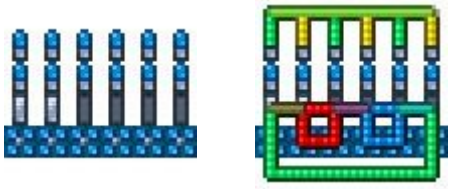
Figures 6.1 (a) and 6.1 (b) are our section 2 3.3 Basic circuits that have been learned. Figure 6.1 (b) is the module we used in Section 3.7. Figure 6.1 (d) exhausts the wire color, so you need to choose carefully when applying it.

The reset section of Figure 6.1 (e) uses an embedded assignment circuit.

6.1.2 Traditional two-way relay circuit

A two-way pass circuit is a pass circuit that can be cycled in both directions. Both positive and negative are controlled by two wires.

(a) Traditional two-way delivery



(b) The traditional two-way delivery of dense rows

(c) Traditional two-way delivery with reset

Figure **6.2:** Traditional two-way relay circuits

6.1.3 Promote the delivery circuit

The relay circuit can form a loop, which is to change the logic light state by relying on the fault logic gate, which in turn determines whether the fault logic gate is activated. In other words, the state of the circuit in each step determines the state of the next step. In this way, a chain of states is formed. We have reason to believe that in certain wiring modes, the state chain forms a loop.

Example **6.1** analyzes the 4-traditional relay circuit as shown in Figure 6.3.

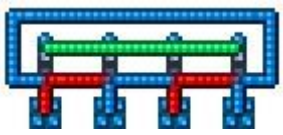


Figure **6.3**

Solution 4- Traditional recurrings have a total of 16 states, and their successors are shown in Table 6.1.

The current state	The next state
0000	0000
0001	1000
0010	0001
0011	1001
0100	0010
0101	1010
0110	0011
0111	1011
1000	0100
1001	1100
1010	0101
1011	1101
1100	0110
1101	1110
1110	0111
1111	1111

Table **6.1**

Based on Table 6.1,we get all the loops of 4- traditional recurrings:

0000 → 0000

1000 → 0100 → 0010 → 0001 → 1000

0110 → 0011 → 1001 → 1100 → 0110

0101 → 1010 → 0101

0111 → 1011 → 1101 → 1110 → 0111

1111 → 1111

4- Traditional recurrings have a total of 6 cycles, of which 2 are 1 loop,1 is 2 loops, and3 are 4 loops. Example **6.2** Analyzes the promotional delivery circuit shown in Figure 6.4.

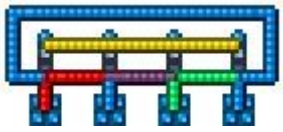


Figure **6.4**

Solution Based on Table 6.2,we arrive at all the loops for this promotion:

0000 →0000

0001 →1000 → 0110 → 0011 → 1001 → 1110 → 0101 → 1010 →

0111 → 1011 → 1111 → 1101 → 1100 → 0100 → 0010 → 0001

There are 2 loops for this promotion, 1 1 loop, 115 loops.

It's not hard to see that no matter how the promotion is connected, the 1 loop of 0000 → 0000 won't change, let's take this

The current state	The next state
0000	0000
0001	1000
0010	0001
0011	1001
0100	0010
0101	1010
0110	0011
0111	1011
1000	0110
1001	1110
1010	0111
1011	1111
1100	0100
1101	1100
1110	0101
1111	1101

Table **6.2**

Status number	Two-light status	The status of the blue line	Green line status
0	10	0	0
1	01	1	0
2	11	0	1

Table **6.3**

A loop is called an ordinary loop and is ignored in subsequent discussions. If all the non-zero states of a promotional pass together make up a loop, then we call this loop a full loop. Promoting the diversification of delivery connections, the diversification of states, the diversification of cycles bring more possibilities.

Example **6.3** Analyzes the down-frequency circuit in Figure 6.5.

Figure **6.5**

The circuit is divided into two parts: the two doors on the left are 2- promotional delivery circuit, the right door is 2- down-frequency circuit,and the green line of the promotion pass is received by 2-down-frequency circuit input.

When analyzing the output of the promotion pass, we should not only analyze the status of each lamp, but also analyze the status of each line. In this example, the status of the two lights and two wires for the promotion pass is shown in Table 6.3, where the initial state of the two lines is set to 0.

This promotional pass cycle is 3,and the green line is activated twice in each cycle, and after passing the 2-down frequency circuit, the torch is activated once in each cycle. So the down-frequency circuit in Figure 6.5 is the 3-down-frequency circuit.

If you use a 3- traditional pass-through circuit for frequency reduction, the width or height of the circuit should be increased by at least one grid due to the pendulum. Promotional delivery has the advantage of volume in this simple function.

If you remove the 2-down circuit on the far right, the 2-promote delivery on the left becomes an uneven down-frequency circuit, or ,(1,2)-down circuit, i.e. the cycle switches between 1 and 2. Changing the number of doors and wiring of the promotional pass can get a variety of uneven frequency reduction circuits. Series uneven frequency reduction circuits can produce more complex output uneven downfrequency circuits, such as two (1,2)-down-frequency circuits in series, which can obtain (1,3,2,3)-down-frequency circuits, the derivation process of which is left to the reader.

Not all promotional deliveries can produce non-trivial loops.

Example **6.4** The status chain of promotional deliveries shown in Figure 6.6 is shown in Figure 6.7. This state chain is a tree that does not contain any of the non-ordinary loops.

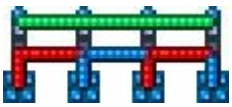


Figure **6.6**

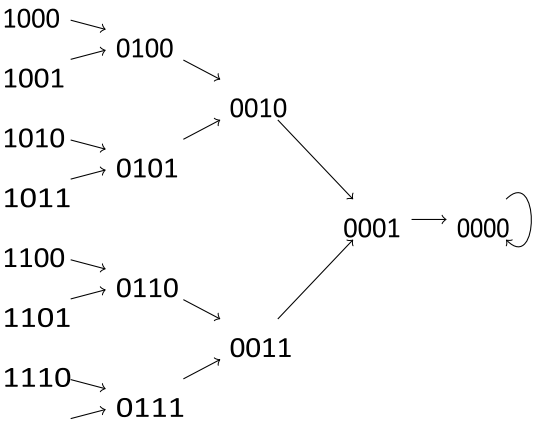


fig **6.7**

Example **6.5** The status chain of promotional deliveries shown in Figure 6.8 is shown in Figure 6.9. This state chain consists of two trees.

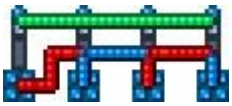


Figure **6.8**

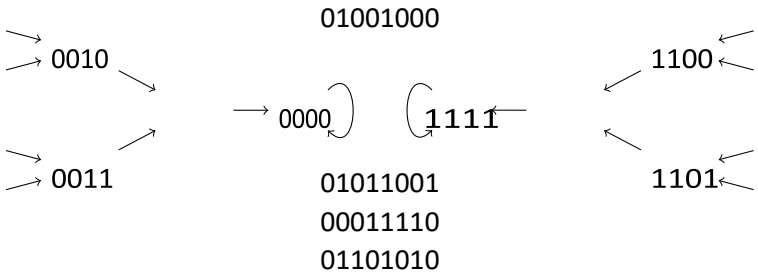


Figure **6.9**

Example **6.6** The status chain of promotional deliveries shown in Figure 6.10 is shown in Figure 6.11,which has both ring and treestructures.

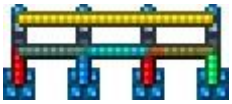


Figure **6.10**

At present, in addition to violence, it is not possible to predict the promotion of the behavior of the delivery, design promotion by trying and experience.

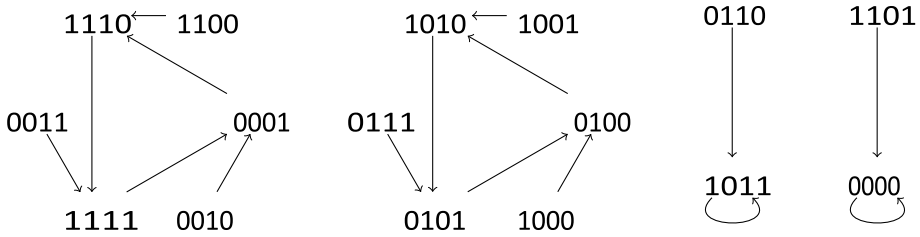


Figure **6.11**

6.2 Drive

6.2.1 Custom half-brick drive

A 60Hz drive can be obtained using a dummy drive. In addition to using down-frequency technology, you can modify the half-brick unit (Figure 6.12)if you want to do a frequencyreduction.

(a) (b)

Figure **6.12:** (a) Use only one pressure plate, the frequency is reduced to 30Hz,and (a) the lengthening of the brick is reduced to 20Hz.

6.2.2 Other driver summary

Since timers and dummy drivers are already free to control time, other drivers fade out. But in order to keep thinking, still introduce these drivers, perhaps when there is a strange effect.

- Bio-driven: a drive that uses biowalking speed to fix the structure. Often use statues to brush monsters. Distance from the player is not

  Can be too far away, otherwise the creature will disappear. Statue Monster Speed Test https://www.bilibili.com/video/av22739934. Before the introduction of puppets in version 1.3.0.1, half-brick drives typically used skeletons generated by skeleton statues.
- Column drive: A driver constructed using a stable interval of the lower stacking platform. LINKHTTPs://www.biliibili

  .com/video/av23028215。

- Conveyor drive: Described. Although still full-frequency driven, player freedom needs to
- be restricted.

Organ drive: The drive constructed by using a variety of circuit items and turquoise pressure pads that can fire projectiles. Because the projectile speed varies and is also affected by the fluid deceleration, control is not as simple as dummy drive. However, because the footprint can be small, this type of drive still has some value.

- Conveyor drive: Uses the conveyor to transfer objects with the same characteristics as the position of the transfer area. See Figure 6.13

6.3 Sensor

(a) (b)

Figure 6.13: (a) Trigger the conveyor when stepping on the red pressure plate, which hangs one grid after transmission and continues to transmit when it falls onto the pressure plate, so that the cycle occurs;

6.3 Sensor

Terraria already comes with a number of sensors, such as pressure plates, sensors, etc. In practice, we sometimes need to detect things that the game can't detect with its own sensors, so we need to design additional sensors.

6.3.1 Open sensor

The so-called service sensor is the power that is activated when the map is turned on. As long as the map is not turned off, the power supply will not be activated again.

The easiest thing to think about is putting the player sensor on the rebirth point and transferring the player away. However, this activates again when the player returns. Use bed transfer technology to avoid this, but it cannot be reset automatically when exiting the map.

From the moment the map is opened, the puppet shadow is generated on the puppet the first time the player approaches the puppet. Puppets are furniture and puppet shadows are enemies, and they are settled separately: furniture is fixed, and shadows can move, be hurt, and be debuffed.

Although shadows do not move actively, they do, such as falls, conveyor transfers, and half-brick transfers. The furniture plays an animation when the shadow is injured. Shadows and

furniture are located in blocks that for their own reasons cannot be placed in prospect [12]blocks.

As long as the puppet shadow is near the rebirth point, the puppet shadow is regenerated on the puppet when the map is opened.

The simplest opening sensor is Figure 6.14. The puppet is suspended and there is a pressure plate underneath. Each time the map is opened, the puppet shadow is generated and dropped on the pressure plate to activate the pressure plate. This open sensor has a slight delay, which is the time when the puppet shadow falls onto the pressure plate. Generally speaking, such a short delay is not a problem.

Figure **6.14**

If you want a shorter delay, consider using a conveyor to transfer the puppet shadow away at the moment of opening, and do some processing so that the shadow is not transmitted back when the conveyor is reactivated (Figure 6.15).

In the unit above, after 1 frame of the conveyor is activated, the puppet shadow is pushed away from the conveyor by half a brick and the red pressure plate is triggered, so that it is no longer transmitted back. However, the player sensor and the pressure booster do not trigger the conveyor, and the player is born at the rebirth point and does not trigger the normal pressure plate. Only trigger the organ with the player sensor or the accent pressure plate, and then use the turquoise pressure pad

6.3 Sensor

Figure **6.15**

The board activates the conveyor, which results in a short delay between the trigger organ and the activation of the turquoise pressure pad.

Liquids allow for a delay-free opening sensor. One of the waiting interfaces for opening the map is "Liquids in place." This step is to transfer all the unstable liquids in the map to the lowest point. In Figure 2.36, the water brusher continuously generates water, which enters the slender passage to the left and is absorbed by the lava track below. Water is always

[12] Foreground blocks cannot be placed on furniture panels or on biological collision boxes。

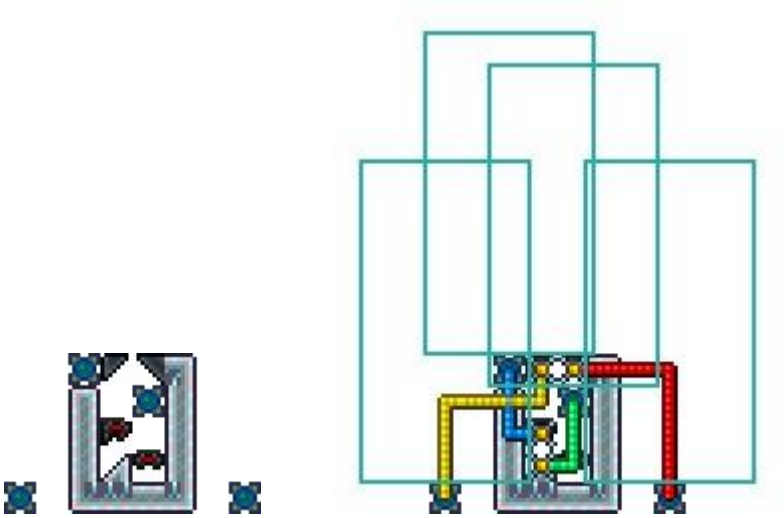
present in the channel when the channel is long enough and the water is brushed fast enough. When you exit the map and reload it, the water in the channel is automatically placed on the lowest liquid sensor, which activates the liquid sensor. The function of the rest is to drain water from the liquid sensor and to turn on the 1-second timer that brushes the water. There is no delay in triggering this device, but the water brush is always running in the game, which may affect the performance of the computer.

6.3.2 Direction sensor

Direction sensors are widely used in circuit games and can detect player movements up and down left and right.

A scheme for direction sensors is shown in Figure 6.16. Another scenario refers to the https://www.bilibili.com

/video/av23633364/?p=7。



(a) (b)

Figure **6.16:** When a player tries to move left or right, the player sensor is triggered and half a brick is solidified to push the player back;

6.3.3 Brush Monster Sensor

Brush Monster sensors are used in brush monster fields and Boss battlefields to detect various enemy generations. Most enemy monsters can trigger the pressure plate directly, so in general the pressure plate can be used as a brush monster sensor. Wall-piercing monsters do not trigger

Pressure plates can only be handled using special methods.

A grudging scheme is to place pressure plates around the player, and when the wall monster attacks the player, the player is repulsed, triggering the pressure plate, so that the enemy monster is detected. The disadvantages of this scheme are obvious, but there is no better way for the time being.

6.4 Monitor

The display is divided into segmented display, dense matrix display, sparse matrix display and pixel box display.

6.4.1 Segmented display

A segmented display, which is a display that is controlled separately by several parts of the display interface, depending on the display needs. When some light source states in the display are always synchronized, treat these light sources as a single light source, i.e. as a segment. In decimal number display, we divide the display into seven segments, in binary number display into two segments, and in the thinking question of Chapter 2, we leave the segmentation task of the moon phase display to the reader.

The control circuit can be designed after segmentation. Figures 2.17 and 2.39 show two different control circuit styles for decimal numbers. The former logic doors are tightly arranged and require more line changers, while the latter logic doors have intervals and require fewer wire changers but take up more space. Because the display part itself is small in size, the use of a large footprint control circuit can easily lead to unsightly wiring (Figure 2.32).

In addition, Figures 2.17 and 2.39 show two different segmentation methods for decimal numbers. A segmented display can have many ways of segmenting, with different segmentation methods corresponding to different control circuits and wiring. A natural question is, can a segmented approach be devised to make the control circuit simpler? The answer is that there is no fixed circuit that makes the control circuit simple. Nevertheless, we can often make some simplifications. Here's an example of how to optimize segmentation.

We want to modify the segmentation of the decimal number to reduce the size of the control circuit. The initial segment is shown in Figure 2.15. Under this segment, the correspondence between numbers and segments is shown in Table 6.4.

	a	b	c	D	and	f	g
0	1	1	1	0	1	1	1
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4	0	1	1	1	0	1	0

5	1	1	0	1	0	1	1
6	1	1	0	1	1	1	1
7	1	0	1	0	0	1	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Table **6.4:** Decimal numbers show the display matrix of traditional wiring

The matrix above ranks [13]7 3 under F2,so the display must be divided into at least 7 segments, and there is no way to reduce the number of segments. So is it impossible to reduce the number of wire changers in the control circuit? some. In some cases, you can connect two lines of the same color to the same line changer and make them do not overlap**(?).**）。 In this way,a seven-segment line can be combined into up to three pairs so that it can be joined on a single line.

We make a primary column transformation of the corresponding matrix and try to separate 1 of the two columns. The columns for each column are a subset of the collection , sa, b, c, d, e, f, g, and the contents of the columns marked X are called column X. Primary column transformations are divided into two operations:

1. Swap columns X and Y,and X and Yat the sametime;
2. Add column X to column Y (F_2 addition) and change X to Symmetry Difference 4 for X and Y.

The primary column transformation process is shown in Figure 6.17 and Figure 6.18. Figure 6.17 makes the upper part of the right column [14] 7, and Figure 6.18 makes the bottom of the left column 0. After the transformation, column 1 can be merged with column 5, column 2 can be merged with column 6, and column 2 [1516]can be merged with column 7. The combined wiring is like **?** Note That because the two columns are merged with duplicate segments, the same color wire cannot be used on the display and an additional wire changer must be used. Continuing to use the primary column transformation reduces the use of line breakers (**?**）。

In the example above, the display goes out when the binary input is 1010 to 1111, which means that the display has eleven display states. If we don't need to turn off the state, reduce one state and see if we can further reduce the use of wire changers.

Using another technique here, we first do a pre-processing to reduce the number of 1s in the matrix, which facilitates separation. Define the reverse operation of the matrix column:

---

[13] 域 F_2 contain 0,1 Two elements that you can 0 Think of it as an even number,1 think of it as an odd number. Addition rules:0+0=1+1=0，0+1=1+0=1Multiplication rules:

[14] *0=0*1=1*0=0，1*1=1。

[15] Later, the primary column transformation operation of the matrix is introduced and the column rank of the matrix is proved 7。

[16] gather A with the collection B The symmetry difference is defined as $A\triangle B = (A \cup B) - (A \cap B)$

swap the case of the column label and swap the column 01. When a segment marked in capitals is wired, the 01 status of the torch is the opposite of normal. Obviously this only makes sense before the primary column transformation.

In the initial display matrix, all columns except e have less than 1 to 0, so reversing all other columns reduces the number of 1s in the matrix (Table 6.5). To simplify this matrix, get **?** , and do some more technical work (**?** .), you can use the wire changer **(?).** ）。

	a	b	c	D	and	f	g
0	0	0	0	1	1	0	0
1	1	1	0	1	0	0	1
2	0	1	0	0	1	1	0
3	0	1	0	0	0	0	0
4	1	0	0	0	0	0	1
5	0	0	1	0	0	0	0
6	0	0	1	0	1	0	0
7	0	1	0	1	0	0	1
8	0	0	0	0	1	0	0
9	0	0	0	0	0	0	0

Table **6.5**

In addition, there are other ways to simplify it:

	a	b	c	D	a
0	1	1	1	0	1
1	0	0	1	0	0
2	1	0	1	1	1
3	1	0	1	1	0
4	0	1	1	1	0
5	1	1	0	1	0
6	1	1	0	1	1
7	1	0	1	0	0
8	1	1	1	1	1
9	1	1	1	1	0

(a) The original matrix

	abcefg	c	b	D	and	f	g
0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0
2	1	0	1	1	0	1	0
3	1	0	1	1	1	0	0
4	0	1	1	1	0	1	0
5	1	1	0	1	1	0	0
6	1	1	0	1	0	0	0
7	1	0	1	0	1	0	1
8	1	0	0	1	0	0	0
9	1	0	0	1	1	0	0

(b) adds column 1 to column 2, 3, 5, 6, 7 and swaps columns 2,3

	abcefg	cf	b	D
0	1	0	0	0
1	0	1	0	0
2	1	0	1	1
3	1	0	1	1
4	0	1	1	1
5	1	1	0	1
6	1	1	0	1
7	1	0	1	0

	abcefg	cf	Bdf	and	D	f	g
0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0
2	1	0	1	0	0	0	0
3	1	0	1	1	0	1	0
4	0	1	1	0	0	1	0
5	1	1	0	1	1	1	0
6	1	1	0	0	1	1	0
7	1	0	1	1	1	1	1

8	1	0	0	1	0	8	0	0	1	0	0	0	1	0	0
9	1	0	0	1	1	9	0	0	1	0	0	1	1	0	0

(c) Add column 2 to column 6 **(d)** Add column 3 to column 4, 6, in exchange for column 4,5

	abcefg	cf	Bdf	if	f	D	g
0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0
2	1	0	1	0	0	0	0
3	1	0	1	1	0	0	0
4	0	1	1	0	1	0	0
5	1	1	0	1	0	1	0
6	1	1	0	0	1	1	0
7	1	0	1	1	0	1	1
8	1	0	0	0	0	1	0
9	1	0	0	1	1	1	0

(e) Add column 4 to column 6 in exchange for column 5, column **6.17**

	abcefg	cf	Bdf	if			abcefg	cf	Bdf	if	and	abcdfg	g
0	1	0	0	0	(	0	1	0	0	0	0	0	0
1	0	1	0	0	(	1	0	1	0	0	0	0	0
2	1	0	1	0	(	2	1	0	1	0	0	0	0
3	1	0	1	1	(	3	1	0	1	1	0	0	0
4	0	1	1	0	:	4	0	1	1	1	1	0	0
5	0	1	0	0	:	5	0	1	0	1	1	1	0
6	0	1	0	1	(	6	0	1	0	1	0	1	0
7	0	0	1	0	:	7	0	0	1	1	1	1	1
8	0	0	0	1	:	8	0	0	0	0	1	1	0
9	0	0	0	0	(	9	0	0	0	0	0	1	0

(a) Add column 6 to column 1, column 4,5 **(b)** column 5 to column 4

	abcefg	cf	Bdf	if	and	abcdfg	acf
0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0
2	1	0	1	0	0	0	0
3	1	0	1	1	0	0	0
4	0	1	1	1	1	0	0
5	0	1	0	1	1	1	0
6	0	1	0	1	0	1	0
7	0	0	0	0	0	0	1
8	0	0	0	0	1	1	0
9	0	0	0	0	0	1	0

(c) Add column 7 to column 3, 4, 5, 6

	abcefg	cf	Bdf	that	and	abcdfg	acf
0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0
2	1	0	1	0	0	0	0
3	1	1	1	1	0	0	0
4	0	0	1	1	1	0	0
5	0	0	0	1	1	1	0
6	0	0	0	1	0	1	0
7	0	0	0	0	0	0	1
8	0	0	0	0	1	1	0
9	0	0	0	0	0	1	0

(d) add column 4 to column 2

	abcefg	abeg	Bdf	that	and	abcdfg	acf
0	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0
2	1	0	1	0	0	0	0
3	0	1	1	1	0	0	0
4	0	0	1	1	1	0	0
5	0	0	0	1	1	1	0
6	0	0	0	1	0	1	0
7	0	0	0	0	0	0	1
8	0	0	0	0	1	1	0
9	0	0	0	0	0	1	0

(e) Add column 2 to column 1 Figure **6.18**

- The row of the fabric. In the previous operation, the ten logical doors that control 0 9 were in numerical order.

  With less readability, you can change their order, which may separate the columns of the matrix. Because the order changes, the number of appearances controlled by the relay circuit cannot be used in this way.

- Control a segment with two wires. A 7-segment line merge 3 pair can be placed on one line, and an8-segment line merge 4 pair can be placed on a single line. This way, you can show that a segment with more 1 in the matrix is divided into two segments with less than 1, which may facilitate separation.

- Merge more than two segments. This method is more limited because it affects the wiring of multi-layered control circuits. This approach is available if it can eventually be only tier 1.

6.4.2 Dense matrix display

A dense display with a monochrome light source with a length of at least one side of up to 4 can be achieved (Figure 6.19).

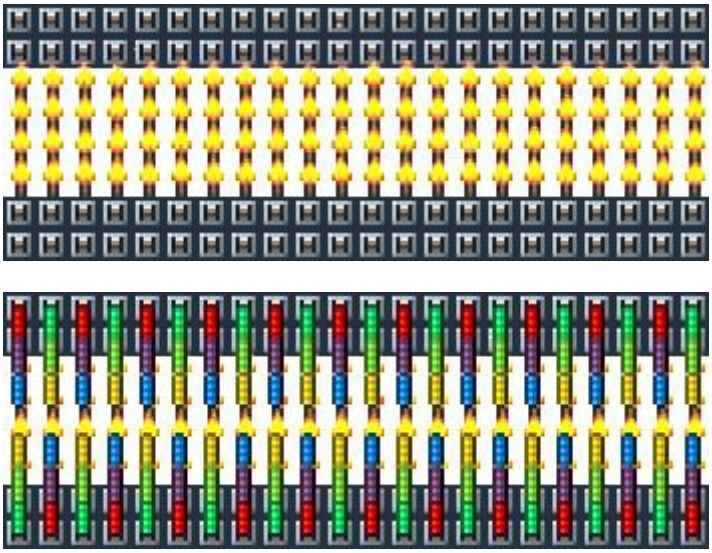


Figure **6.19:** The first line of switch controls the first line of torches, the second line of switches controls the second line of torches, the third line of switches controls the third line of torches, and the fourth line of switches controls the fourth line of torches.

6.4.3 Sparse matrix display

Sparse matrix displays have two main parameters: the light source area and the footprint area of each pixel. The size of the sparse matrix display is unlimited, and a typical sparse matrix display, as shown in Figure 6.20, responds to the torch and switches states when the red and blue lines are entered at the same logical frame. When the display is updated, the red lines of each row are activated one by one in each logical frame, and the blue lines of each column are selectively activated in that logical frame to control the display of the row. In this example, each light source area is 4,the footprint is 12,or 4 for 12,or 4/12. The most outstanding work of this display is currently in use

Yes, https://www.bilibili.com/video/av22343683.

Here we see that the module that controls each pixel is a 3-block fault logic gate. This is currently considered to be the smallest control [17]unit. The interval between the illuminated area is used to allow the controlled wires to pass through. If the torch can be placed on a controlled wire, the display can be made more compact (Figure 6.21).

[17] Expect someone to develop a smaller display

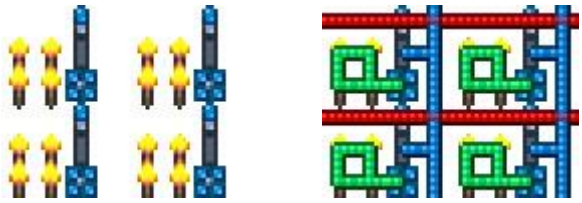
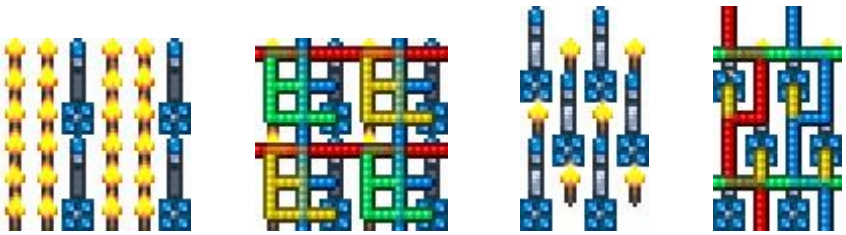


Figure **6.20:** Typical matrix display



(a) 6/9 **(b)** 1/4

Figure **6.21:** Compact matrix display

Here you may wonder, if the torch is placed on the controlled wire, will not be disturbed by the controlled wire? Yes, it will be disturbed, but we can offset this interference with additional activation. Specifically, we can record the number of times each control wire is activated. If a control wire is activated evenly several times, the torch it passes through is not disturbed. If oddly activated, then the torch it passes through is reversed, and then we activate the line one more time to restore the torch to an undisturbed state.

The next question to deal with is, will the extra activation interfere with a pixel? Obviously the wires that connect the valid logic lights will not, and the wires that connect the faulty logic lights may. In fact, if we activate the wire connecting the valid logic lamp first, all valid logic lights will be turned off because each line is activated evenly several times, and then activate the wire connecting the faulty logic light without interfering with any one pixel.

In addition, if you are careful enough, will you find that the wires connected to the faulty logic lights in Figure 6.21 (b) are also connected to the valid logic lights, which is not a bug? In this case, just set the default state of the valid logic light to on.

Another very clever way to reduce the area of the display unit is to use a small map. 2.5 pixels per grid in a small translucent map, alternating between 2 pixels and 3 pixels on the screen. There are many examples of using small maps:

- Terrariais a snake-eater -The RedstoneCrafter https://www.bilibili.com/video/av32265379
- Play Russian Blocks in Terraria!? https://www.bilibili.com/video/av38924330

6.5 Automation

K Thinking question k

1. Why do I need to label columns as symmetrical when transforming the matrix in the simplified parting of the segmented display?
2. Why doesn't a non-pixel box-intensive rectangular random display with more than 4 lengths on both sides exist?
3. Why is the pixel box display up to 24wide?

Appendix circuit-related theme furniture

topic	Candle	CandleStick	Lantern	Chandelier		Door Chest	Toilet	Trapped
not				not	not			
nebula								
Dayao								
Stardust								
Star spin								
bone	not 							

damage									
flesh and blood									
glass									
honey									
freeze									
Jungle lizards									
Life									
bamboo									
Blue Dungeon				not					

topic	Candle	CandleStick	Lantern	Chandelier		Door	Chest	Toilet	Trapped
Green dungeon				not					

Powder dungeon				no				
Conifer wood								
cactus								
crystal								
Dynasty Tree								
Shadow wood								
granite								
marbled								
Mars								
Meteorite								
mushrooms								
obsidian								
Palm wood								
Pearl wood								
pumpkin								
annatto								

topic	Candle	CandleStick	Lantern	Chandelier	Door	Chest	Toilet	Trapped
sandstone								
Shadow wood								
spider								
Ghastly								
Sky								
Slim								
Steampunk								
gold								

Appendix cooling time mechanism

注意本机制属于 1.3.5.3 版本。

The name of the tugger	Cooldown time / frame
fort	30
Snowball launcher	10
Fireworks box	30
Fireworks Fountain	30
Minecart track	5
Darts organ	200
Super Darts Agency	200
Flame organs	200
spear organ	90
Sharp ball organ	300
Fountains (organs).	200
Brush the monster statue	30
Brush the statue of the item	600
Transfer the NPC statue	300

Table **B.1:** The diagram that uses the cooldown mechanism and its cooling time.

Many circuit items have cooling times (Table B.1). The game uses a list of 1000 sizes (hereafter referred to as a cooling list) to store the Tug in the cooling and its remaining cooling time. In each frame, the game refreshes the entire cooling list, subtracts the remaining cooling time for all of these grids by 1,and removes the grid from the cooling list if it isreduced to 0.

In addition, some tuggers do not have cooling time, but also borrow cooling lists for countdowns, including timers and detonators.

When you add a tug to the cooling list, add it to the first empty bit from front to back, and if the grid already exists in the cooling list, or if the cooling list is full, join fails. All other cooling tugs, except the detonator, do not respond to activation if they fail to successfully join the cooling list. Therefore,

not only can the Tug not be activated while it is cooling, but it cannot be activated when the entire cooling list is full.

The refresh order of the cooling list is forward from the back.

When you remove a grid from the cooling list, all the diagrams that follow the cooling list are moved one position forward in turn, so that the cooling list remains continuously filled from the time it goes.

Appendix expertise

C.1 original, anticode, complement code

注意本节系转载。 Reprinted after not changing the original format modification, and deleted the last part of the original text "original code, anti-code, complement code and then go deeper."

Author: Zhang Ziqiu

Origin:https://www.cnblogs.com/zhangziqiu/archive/2011/03/30/computercoof.html

Copyright in this section is shared by the author and the blog park and is welcome to be reproduced, but this statement must be retained without the author's consent and the original connection given in the obvious location on the article page, otherwise the right to be held legally responsible is reserved.

Before you learn the original code, transcode, and complement, you need to understand the concept of machine numbers and true values.

Machine Number A binary representation of a number in a computer, called the number of machines. The number of machines is signed, and the symbol is stored in the computer with the highest bit of a number, positive 0 andnegative number 1. For example, the number of decimals is 3,the computer word is 8 bits long, and the conversion to binary is 00000011. If it is -3,it is 10000011. So, here's the number of machines in 00000011 and 10000011.

True Value Because the first bit is a sign bit, the formal value of the number of machines is not equal to the true value. For example, the signed number 1000011above, whose highest bit 1 represents negative, and whose true value is 3instead of the formal value of 131(10000011 converted to decimal equals 131). Therefore, for the sake of distinction, the true value corresponding to the number of machines with signed bits is called the true value of the number of machines.

Example **C.1** The true value of 000000001 is the true value of .0000001 ,thetrue value of 10000001 is .0000001 .

Before we explore why the machine uses supplements, let's look at the concepts of original code, reverse code, and complement code. For a number, the computer is stored in a certain encoding manner. Original code, anti-code, complement code is the machine stores a specific number of coding methods.

The original code is the absolute value of the symbol bit plus the true value, that is, the symbol is represented by the first bit and the value by the remaining bits. For example,
If it is an 8-bit binary:

[+1] Originally: 00000001

[−1] Originally: 10000001

The first bit is the symbol bit. Because the first bit is the sign bit, the range of values for the 8-bit binary number is: .. .. The original code is the easiest way for the human brain to understand and calculate the representation.

Anti-code Anti-code is represented by: the positive anticode is itself.

C.1 original, anticode, complement code

- The negative number of the anticode is on the basis of its original code, the symbol bits remain unchanged, the rest of the bits are reversed.

The original is the anti of the original

The original s

It can be seen that if an anticode represents a negative number, the human brain can not intuitively see its value. It is usually converted to the original code before it is calculated.

The complement is represented by a positive complement itself.

- The negative number of complements is on the basis of its original code, the symbol bit unchanged, the rest of you take the opposite, the last . (i.e. on the basis of the anti-code

The original is the original s.0000001

The original s

For negative numbers, the complement representation is also the human brain can not visually see its value. It is often also necessary to convert to the original code in calculating its value.

Why use original code, transcode, and complement Code Before I begin my in-depth study, my learning advice is to "rotaly" the original code above, the representation of the countercode and the complement code, and the calculation method. Now we know that computers can represent a number in three encoding ways. For positive numbers because the results are the same for all three encoding methods:

The original s. But for negative numbers:

The original s

Visible original code, transcode and complement code are completely different. Since the original code is directly identified by the human brain and used to calculate the representation, why is there a transcoding and supplement?

First, because the human brain can know that the first bit is a sign bit, in the calculation, we will be based on the symbol bit, choose the real value area addition and subtraction (the concept of true value). But for computers, the addition and subtraction multiplier is already the most basic operation, to design as simple as possible. Computer identification "symbolic bits" will obviously make the computer's basic circuit design very complex! So people came up with a way to involve the symbolic bits in the operation. We know that subtracting a positive number from the algorithm is equal to adding a negative number, i.e.:1 , 1 , 1 , 1 , 0,so the machine can only add without subtraction, so that the computer operation is designed more simply.

So people began to explore ways to involve symbol bits in operations and to preserve only addition. Let's start with the original code: the expression that evaluates the decimal:1 - 1 - 0

1 s 1 s 1 s (-1) s0000001 s original s 1 s0000001 s s.100001 s

If the original code is used to involve the symbol bits in the calculation, it is clear that the result is incorrect for subtraction. This is why the inside of the computer does not use the original code to represent a number. In order to solve the problem of subtraction of the original code, the anti-code appears:

C.2 Floats

The expression of the decimal is evaluated:1 - 1 - 0

1 s 1 s 1 s (-1) s0000001 s original s 1000001 s s 0000001 s anti-111110

The reverse of the s---11111111111-to-10000000000-

It was found that subtraction was calculated using anticode, and the true value of the result was correct. The only problem is actually the special valueof "0". Although it is understood that the upper 0 and the upper0 are the same, the 0 symbol does not make any sense.

And there will be two codes, theoriginal and the original, which represent 0.

The emergence of the complement solves the problem of the symbol of 0 and two encodings:

1 - 1 s 1 s (-1) s0000001 s original s 1000001 snr s 0000001 s.111111

s00000000

"Thus 0 is represented by a s0000000" and the previous problem of s0 does not exist. And it can also be expressed in the name of10000000:

(−1)+(−127) = [10000001] The result of the original s.11111111111111111111111111111111111111111111001. Note, however, that because the previous complement of 0 is actually used to represent thepreviousone, it does not have the original and anticode representations. (The complement tothe s128 indicates that the original code of the s10000000 is the original code of the s.0000000, which is incorrect) uses a complement code that not only fixes the symbol of 0 and the problem of two encodings, but also represents a minimum number. That's why the 8-bit binary, represented by the original or anticode, has a range of .127, .

Because machines use complements, the range can be represented by : for the 32-bit int type commonly used in programming

$[-2^{31}, 2^{31} - 1]$ Because the first bit represents a sign bit, you can save an additional minimum value when using a complement representation.

C.2 Floats

注意本节系摘编。

Floating points mean that floating points are a formulaic representation that approximates the representation of real numbers and can be traded between the range of expressions and the precision of the representation (hence the name float).

Floating points are usually expressed as :

$$N = M \times R^E$$

For example:12. 345 = 1. 2345 × 10^1。

Where*M* (Mantissa)is called the end of the float, *R*(Radix)is called the cardinality of the order code, and*E*(Exponent)is called the order code of the order. R is generally specified in a computer as 2,8, or 16and is a definite constant that does not need to bespecified in floats.

C.2 Floats

Therefore, under known criteria, to represent floating points,

One is to give the value of the mantissa M, usually in the form of fixed-point decimals, which determines the representation accuracy of floating points, i.e. the number of digits of valid numbers that can be given.

The second is to give the order code, usually in the form of fixed-point integers, which indicates the position of decimal points in the data, determines the range of floating points.

Therefore, in a computer, floats are usually represented as Figure C 1 in the format shown. (Assuming a 32-bit float with a base of 2,where thehighest bit is the sign bit).

Figure **C.1**

The specification of floating points Represents a floating point that has a different representation according to the exponential representation method above:

0. 3 × 10^0; 0. 03 × 10^1; 0. 003 × 10^2; 0. 0003 × 10^3

In order to improve the accuracy of data representation while ensuring the uniqueness of data representation, it is necessary to do the specification of floating points.

Within a computer, for floating points that are not **0** values, the absolute value of the mantissa is required to be greater than the cardinality inverse, that is, | M| ≥ $\overline{R}1$。

That is, the highest effective bit of the required end field should be 1,so that the floating points that meet this representation requirement is standardized to indicate: the tail number that does not meet this representation requirement, into the operation process of the tail to meet this requirement, called the specification process of the floating point number, through the tail shift and modification of the order code to achieve.

For example, the normalized number of binary codes is expressed :(0 positive 1 negative).

- Positive 0.1xxxxxx
- negative 1.1xxxxxx

Note that the highest bit of the mantissa is always **1,**so we can omit that bit completely.

At this point, we introduce the IEEE754 standard, which constrains most of the usage settings for floating points: the tails are in the original code and the highest number of tails is hidden.

The highest number of the tail value of the original code non-0 value floating point must be 1,so the bit can be ignored, so that with the same number of digits can save an extra binary number, which is conducive to improving the accuracy of data representation, said this processing scheme uses hidden bit technology. Of course, when you retrieve such floats to the operator to perform an operation, you must first restore the hidden bit.

- The order code uses "shift code" and the **base is** fixed to **2**

As in Figure C.2, the number of 32bit floats and 64bit floats, from the highest bits to the sign bits, order codes, and tails, the true value of a normalized 32-bit floating point x is :

$$x = (-1)^S \times (1.M) \times 2^{E-127}$$ The true value of

a normalized 64-bit floating point x is :

$$x = (-1)^S \times (1.M) \times 2^{E-1023}$$

C.3 BCD 码

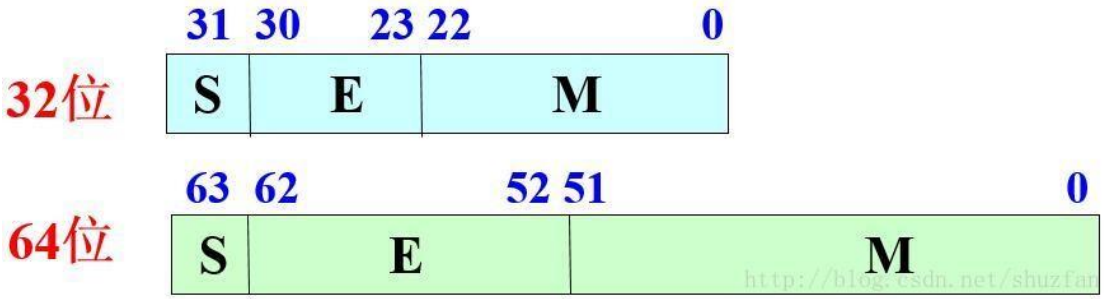


Figure C.2

Here's an example of a 32-bit single-precision floating point -3.75 that helps to understand: First convert to a 2-in representation

?3. 75 =?(2 + 1 + 1/2 + 1/4) =?1. 111× 2^1 Organize the symbol bits and specification them

?1. 111 × 2^1 = (? 1) $^{(1)}$ × (1 + 0. 11100000000000000000000000000

×00000000000000000000000000000000000000000000000000000000000000000000000000000000

00000000000000000

(? 1)$^{(1)}$×(1+0. 11100000000000000000000)×2^1 = (?1)$^{(1)}$×(1+0. 11100000000000000000000)×$2^{128?127}$

Thus, the symbol bit *S* is 1,the tail *M* is 11100000000000000000000000000, and the order code *E* is 128_{10} s 10000000_2,and thefinal 32-bit single-precision float is

11000000011100000000000000000000

C.3 BCD 码

注意本节系摘编。

Binary-Coded Decimal,or BCDforshort, is called a BCD code or binary decimal code, also known as a binary decimal.

is a binary form of digital encoding, with binary encoded decimal code. This encoding form uses four bits to store a decimal number, allowing the conversion between binary and decimal to be carried out quickly. This coding technique is most commonly used in the design of accounting systems,

which often require accurate calculations of long strings of numbers. Compared with the general floating-point counting method, the BCD code can save the accuracy of the value, but also avoid the time it takes for the computer to do floating-point operation. BCD coding is also common for other calculations that require high accuracy.

Because the decimal number is 0,1,2,. . ,9 ten digits, so at least 4-bit binary codes are required to represent 1-bit decimal numbers. When using BCD coding, be sure to note that it has only ten valid codes, i.e.0000 1001. The remaining six encodings of the four-digit binary numbers 1010,1011,1100,1101,1110,1111 are not valid encodings. Common BCD codes are 8421BCD codes,2421BCD codes, and remaining 3 yards, corresponding to coding tables such as Table C.1. In this document, you can make

C.4 BCD code to binary

with **8421** yards.

Decimal number	8421 yards	2421 yards	The remaining 3 yards
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100

Table **C.1:** Common BCD encoding. Typically, development uses 8421,8421 for novices to understand at a glance.

C.4 BCD code to binary

The BCD code is the way we represent decimal numbers, andthe algorithm ofBCD to binary is the algorithm of decimal to binary. Here we introduce the short division method, which is the content of the high school curriculum.

The short division divides a number repeatedly by 2 until the quotient is 0,and the remainder is in reverse order as a binary result.

Example **C.2** turns 19 into binary .

$$19 \div 2 = 9 \cdots 1$$

$$9 \div 2 = 4 \cdots 1$$

$$4 \div 2 = 2 \cdots 0$$

2 ÷ 2 = 1 · · · 0

1 ÷ 2 = 0 · · · The inverted

10011 of the remainder is a binary representation of 19.

For BCD, we can replace the operation of dividing by 2,taking and the remainder with some simpleoperations.

Division is a complex operation, so the benefits of replacing division with simple operations are high. To understand this approach, let's first review the long division that primary schools have learned.

C.4 BCD code to binary

Example **C.3** Calculates the quotient and remainder of 123456789 divided by 2 by the long division of BCD codes.

business	0000	0110 0100	0001	0111	0010	1000	0011	1001
dividend	0001	0010 1001	0011	0100	0101	0110	0111	1000
	0000							
	1	0010						
	1	0010						
			0	0011				
				0010				
			1	0100				
				1	0100			
						0	0101	
					0100			
						1	0110	
						1	0110	
							0	0111
						0	0110	
						1	1000	
							1	1000

	0 1001 0 1000
remainder	1

From the long division of division with a division of 2 (not only the example above), we found the following pattern:

1. The last bit of a decimal bit of the BCD code is the remainder left after the decimal bit is removed.
2. The value of one decimal bit of the quotient depends on the first three digits of the decimal bit BCD code and the last digit of the previous decimal bit BCD code, as detailed in Table C.2.

The previous decimal bit is divided BCD code	The division corresponds Decimal bit BCD code	business	interpretation
XXX0	000X	0000	0, 1 ÷ 2 = 0
XXX0	001X	0001	2, 3 ÷ 2 = 1
XXX0	010X	0010	4, 5 ÷ 2 = 2
XXX0	011X	0011	6, 7 ÷ 2 = 3
XXX0	100X	0100	8, 9 ÷ 2 = 4
XXX1	000X	0101	10, 11 ÷ 2 = 5
XXX1	001X	0110	12, 13 ÷ 2 = 6
XXX1	010X	0111	14, 15 ÷ 2 = 7
XXX1	011X	1000	16, 17 ÷ 2 = 8
XXX1	100X	1001	18, 19 2 = 9

÷

Table **C.2:** X for 0 or 1

C.5 Binary to BCD code

In this division, the 4-bit BCD code for each decimal of the quotient is determined by the 4-bit dislocation in the divided BCD code, and its correspondence is already given by Table C.2. The remainder is the last digit of the divided BCD code. So far, we have got the method of getting the quotient and the remainder directly without dividing. By repeating this method, you can convert the BCD to binary.

Example **C.4** converts the decimal number 123 to binary.

1. Write down the BCD code of 123 for 1 0010 0011;
2. Take out the last bit 1 as the first remainder and divide the remaining 8 bits from right to left into a group of 4 bits: 100 ÷1 0001;
3. Take out the last bit 1 as the second remainder and divide the remaining numbers from right to left into groups of 4 digits: 0011 ÷0000;
4. Take out the last bit 0 as the third remainder and divide the remaining numbers from right to left into groups of 4 digits: 000 ÷1 1000;
5. Take out the last bit 1 as the fourth remainder and divide the remaining numbers from right to left into groups of 4 digits: 1010;

   To the merchant is 0111(15 ÷ 2 , 7 , 1;
6. Take out the last bit 1 as the fifth remainder and divide the remaining numbers from right to left into groups of 4 digits: 0011;

   To the merchant is 0011(7 ÷ 2 , 3 , 1;

7. Take out the last bit 1 as the sixth remainder and divide the remaining numbers from right to left into groups of 4 digits: 0001;

   To the merchant is 0001(3 ÷ 2 , 1 , 1;

8. Take out the last bit 1 as the seventh remainder and divide the remaining numbers from right to left into groups of 4 digits: 0000;

   To the merchant is 0000(1 ÷ 2 , 0 , 1; .

9. Out of the reverse sequence of all remaining numbers,1111011 is the binary representation of 123.

In each of the above steps, the implementation of the residual, grouping, check table three steps, we combine these three steps, with vertical simple expression:

Example **C.5** Converts the BCD code 100100011 to binary.

1	0	0	1	0	0	0	1	1
0	1	1	0	0	0	0	1	1
0	1	1	0	0	0	0	1	1
0	1	0	1	0	1	0	1	1
0	0	1	1	1	1	0	1	1
0	0	1	1	1	1	0	1	1
0	0	1	1	1	1	0	1	1

C.5 Binary to **BCD** code

Review the algorithm in example C.5. In each step, the last, group, and table checks are separated, and eventually pushed from the BCD code on the first line to the binary on the last line. Each step is uniquely determined by Table C.2, so you can

C.5 Binary to BCD code

As a reliable algorithm. Conversely, if you know the binary of the last line first, can you push back to the BCD code on the first line? Let's try.

Example **C.6** converts binary 1111011 to a BCD code. Try to reverse the steps in example C.4.

1. The binary number is 1111011.
2. Take 0 as the quotient, from right to left in group ÷s of 4 digits: 0000;
3. Take the dibutary 0001 as the quotient, check table C.2 to get the division is 0001 X,replace X with the second bit of the binary and regroup to get the division of 0011(3 ÷ 2 , 1 , 1, and
4. Take the previous division 0011 as a quotation, check table C.2 to get the division of 0011 X,replace X with the third digit ofthe binary and regroup to get the division of 0111(7 ÷ 2 , 3 , 1;

5. Take the divided number 0111 as the quotient, check table C.2 to get the division is 1010 X,replace X with the fourth digit ofthe binary and regroup to get the division of 0001 0101(15 ÷ 2 , 7, 1;

6. Take the previous step of the division 0001 0101 as a quotient, check table C.2 gets the division of 0001 1000 X,replace X with the fifth bit of thebinary and regroup to get the division of 0011 0000(30 ÷ 2 , 15 , 0;

7. Take the previous step of the division 0011 0000 as a quotient, check table C.2 to get the division of 0011 0000 X,replace X with the sixth bit ofthe binary number and regroup to get the division of 0110 0001(61 ÷ 2 , 30 , 1);

8. Take the previous step of the division 0110 0001 as a quotient, check table C.2 to get the division of 1001 0001 X,replace X with the seventh bit ofthe binary and regroup to get the division of 0001 0010 0011(123 ÷ 2 , 61 , 1;

9. The BCD code sought is 1 0010 0011.

The steps above are written as vertical

0	0	1	1	1	1	0	1	1
0	0	1	1	1	1	0	1	1
0	0	1	1	1	1	0	1	1
0	1	0	1	0	1	0	1	1
0	1	1	0	0	0	0	1	1
0	1	1	0	0	0	0	1	1
1	0	0	1	0	0	0	1	1

Common tools and web sites in the appendix

D.1 Wiki English | Chinese | Chinese mirror station |

Chinese mirror (fun) | English apk

D.2 Forum

Official Forum - The | Of The T-MEC Alliance Of Circuit EngineerS T-MEC Posts Archive | Chinese forum | CurseForge Map Download

D.3 Map Editor

Home | Download 1 | Download 2 | Recommended tutorials (1)(2)(3)(4).

D.4 module

D.4.1 tModLoader official website

| GitHub | Download the | Wiki

| The document

D.4.2 CheatSheet

Home | Download 1 | Next 2 | The publishing page

D.4.3 HERO's Mod

Home | Download 1 | Download 2 | The publishing page

D.4.4 MechScope

Home | Download | The publishing page | Chinese introduction

D.5 Source

1.3.5 and 1.4 versions | 1.3.2.1 | 1.3 | 1.2.4.1 | 1.2.0.3.1 | Xbox 360

The appendix recommends video

E.1 Teaching

(Danger)Tyraria Terraria circuit teaching is https://www.bilibili.com/video/av1

9065519

- Terrariacircuit advanced (1) circuit settlement and burst door https://www.bilibili.com/video/av5

6232462

- Terrariacircuit advanced (2) circuit settlement example https://www.bilibili.com/video/av566

12391

- Terrariacircuit advanced (3) equivalent logic
- https://www.bilibili.com/video/av56811001 Terrariacircuit advanced (4) pixel box

display https://www.bilibili.com/video/av578

42510 - Tyralia 1.3pe High Efficiency Pumpkin Theological Details Tutorial

https://www.bilibili.com/video/av7

0594057

- Terrariav1.0 Semi-Automatic and Fully Automatic Tree Farm Construction (with fully automatic night-light mushroom farm and fully automatic fairy

Palm field) https://www.bilibili.com/video/av27408545

- The Terraria circuit transmits a array teaching map to explain the https://www.bilibili.com/video/av249

05110

- [Terraria] Mappygaming's video fan question-and-answer session https://www.bilibili.com/video/av2

2788696

- Terraria Old Legion Event Tips - Tricks -MappyGaming https://www.bilibili.com/v

ideo/av21707397

- [Terraria] The stone giant of the ghost animal! -Zerogravitas https://www.bilibili.com/video/av220

88547

- [Terraria] Item half brick as well as some interesting apps!-Zerogravitas https://www.bilibili.com/v ideo/av22739847
- A brief analysis and utilization of the principle of hard upper limit

E.2 Visual effects

- Terraria Bad Apple!! https://www.bilibili.com/video/av46694445
- "Tyralia" Rainbow TV We take off https://www.bilibili.com/video/av16601896
- [Terraria] Officially recommended circuit works - Bad apple (music added later) https://www.bilibili .com/video/av22343683

- Terraria Circuit Pixel Box Fixed Animated Rabbit https://www.bilibili.com/video/av50343156
- [Terraria] Digi-Comp 2 unit display -Zerogravitas https://www.bilibili.com/video /av22690346
- Tyraliya interesting minecart https://www.bilibili.com/video/av5271577
- Terraria Circuits is a birthday present for your girlfriend https://www.bilibili.com/video/av21009075/
- Terraria in-game Pixel Art Animation https://www.bilibili.com/video/av5301176/?p=5
- Terraria Moore Striped Animated Rabbit https://www.bilibili.com/video/av49966963
- Terraria Electronic Clock https://www.bilibili.com/video/av55613747
- Terraria 异 World Quartet ED"Other World Girls Talk" (传动)https: www.bilibili.com/video/av54977071 Tyra copier

E.3 mini-game

- [Terraria] Barrier Challenge Map - Mappygaming https://www.bilibili.com/video/av22787315
- Terraria Terraria Super Difficult Creative Puzzle Map Clearance Introduction (above)https://www.bilibili.com/v ideo/av19793617

- Tyralria Terraria Super Difficult Creative Puzzle Map Clearance Introduction (middle)https://www.bilibili.com/v ideo/av20101902
- Terraria Terraria Super Difficult Creative Puzzle Map Clearance Introduction (below)https://www.bilibili.com/v ideo/av20524074
- Tyraliya plays the game inside the game https://www.bilibili.com/video/av6594668
- [Terraria] Use a conveyor gun to reach maximum speed! 3000 plus mph! https://www.bilibili.com/vid eo/av24580434
- Terraria,FuryForgedAdventure Map Harbinger Clearance Live https://www.bilibili. . c om/video/av37553671/
- Play Russian Blocks in Terraria!? https://www.bilibili.com/video/av38924330 solve
- the Rubik's Cube in Terraria? (Read Description) https://www.bilibili.com/video/av5 6760618
- Tyralaya's first five-piece chess https://www.bilibili.com/video/av93790872
- Terrarialife game https://www.bilibili.com/video/av46416137
- Terrariais a snake-eating https://www.bilibili.com/video/av32265379

E.4 Printing items

- [Terraria] Half-brick truffle worm farm(390 plus bars / hour) ! -DicemanX https://www.bilibili .com/video/av23029412
- Tyralia - Spoofing Corpses https://www.bilibili.com/video/av13411383 Terraria statue
- back blood naked expert month total https://www.bilibili.com/video/av6393957
- Terraria Circuits ,Joe Price,fully automatic brush monster field, brush BOSS collection expert mode https://www. . bilibili.com/video/av32865707/
- Terraria 1.3.1 Pumpkin Moon Final Wave at 806 pm, 255 platinum coins in a night https://www.bilibili.com/video/av5356226/?p=5
- Terraria 1.3 Frost Moon Final Wave at 856 pm, 75820 Total Points - World Record https://www.bilibili.com/video/av5356226/?p=6
- The world record for the fastest boss kill collection that can't be surpassed? https://www.bilibi li.com/video/av20725652
- Pumpkin Gods - Old Legion https://www.bilibili.com/video/av10885401/?p=3
- Pumpkin Gods - Frost Moon Nights https://www.bilibili.com/video/av10885401/?p=4
- Terraria Circuit: Improved efficient fully automatic tree farm https://www.bilibili.com/video/av388 82621/
- Terraria Brusher https://www.bilibili.com/video/av50203346
- Terraria Crystal Crumb Farm https://www.bilibili.com/video/av57287619

The appendix recommends literature

F.1 Mechanism Study

- What exactly do truffles need? Why isn't your
- house qualified? The various information-related
- mechanisms displayed by the personal digital
- assistant/mobile phone are explained in detail
- Dry goods / disaster / originalLifeSteal- blood-sucking cooling details
- Various application tips for broadcast boxes (application of TR string code).
- Terraria Mechanism Puppet Tutorial
- 绳索 glitch Someone found a new glitch? 虚化家具
- glitch Actuating tiles using sand 傀儡影子 glitch I
- solved the "Rouge Dummy" Glitch 烈焰机关 Flame
  Traps And Teal Pads 邪教徒 AI Prototype Lunatic Cultist
  Farm & AI Analysis

F.2 Half Brick

- 半砖驱动 [Showcase] Player Sensor/Weighted Pressure Plate
- Engine 半砖坐骑 Mount hoik tracks and teleportation 半砖史莱
- 姆 Possible to hoik slimes?
- Half-brick town NPC (Guide) Using Character NPCs in Hoiktronics builds half-brick 0
- frame changers Look Discovery: instant one way relay relay for conveyors of the half-brick line changer / split box (Guide) Bypassing certain colored wires with tele
  porters
- Half-brick delay (Guide) How to constructs with build-in delays
- semi-brick vertical reset Vertical reset mechanism half-brick
- random number Hoik Randomisers
- Half-brick conveyor array (Guide How) build a Hoiktronics- Based Teleporter Hub
- (updated version) half-brick conveyor array improvement Diceman Teleporter Hub
- Tweaked
- Half-brick password door Hoiktronics: Programmable Passcode-Protected Door and
- Video Tutorial half-brick multi-digit display smh.com.au Displaying Multiple Numbers Numbers Entered in Sequence Semi-Brick Stack Seven-Segment Line Decimal Display (Project) Sequential Input Digit MaChine (dummies - amaz-).
  ing signaling agents)

- Basic half-brick logic gate (Guide) Update Universal Logic Gates
- 定制半砖逻辑门 [Guide] How to Build Logic Gates of Any Size or Complexity

F.3 Small module

- 通用半砖逻辑门 [Project] "Universal" Logic Gate (can be adapted to a variety of functions depending on wiring)
- Single-frame half-brick logic gate (Guide) 1 Tick Logic Gates half-
- brick logic module (Mini-Builds) Small Mechanisms half-brick
- combination logic (Video Tutorial) Giant Hoiktronics Logic Gates
- half-brick super forward adder (Digital Electronics) 8-Bit Ripple-
- Carry Adder half-brick BCD to binary (Project) BCD-to-Binary
- Transformer
- Half-brick binary turn decimal (Video Guide) How to Build a Binary to Decimal Converter
- Half Brick II - Decimal Conversion (Project) Binary to Decimal Converter (Double Dabble Method) Half-brick Decimal Converter (Project) Inputor with Decimals and Output puzzlings with dummy ghosts
- Half Brick Calculator (Video Guide) How Terraria Computers function
- Artificial Intelligence Inside Terraria (pre-1.3.1).
- 21 点 [Guide] Hoiktronics Computer Programmed to Play Blackjack (with special rules)

F.3 Small module

- Simple logic device A reference guide for simple logic devices
- 飞镖机关枪 [Tutorial WIP] Cooldown Juggling Rapid Fire Dart Trap Engines (Animated Guide)

- Liquid color display smh.com.au Writing in Neon drop
- swindle a new type of invulnerability machine drop
- cheat fall Damage Prevention for Lunar Pillar Event
- Fireworks Box drive New Of Engine! Fireworks?
- Gold Coin Drop Device (showcase) Reliable coin stack breaker
- organ clock Small (?) 12 hour clock!
- Fear drives Fear Engine
- Avoiding total monthly damage Moon Lord Damage Tweak
- uses conveyor to do a half-brick loop A hoik loop out of of
- teleporters conveyor belt drives The Small Little Player
- Over Sensor Engine Gate Transfer (Showcase) Two-Way
- Warp Chamber uses the ROM PERFECT 3x5 (4xN pixel)
- display of the different or door!

Liquid Timing Liquid Trickle Timer (Liquid Sensors, Pumps, Actuator).

Liquid Reproduction (Project/Showcase) More efficient liquid duplication , flooding the world in survival platform tips Physics. Ghost town.

- Open Sensor (Idea/Device) World Load Detector
- 3*5 显示 [Project] unspaced, 3 by 5 matrix display

- The 14-segment and 16-segment characters display the "Project" 14 and 16 Segment Alphanumeric Displays

F.4 Large modules

- 心雕阵（不用拾心药水） [Showcase] The fastest possible heart generator (statue
- farm) 旋转锁 [Showcase] Reprogrammable rotary lock RAM Final RAM Design and
- Update on the Computer Project 滚动文字显示屏 scrolling text on a programmable
- text board 硬盘 Compact Hard Drive
- Pixel Box Display 24 Œ 24 Pixel Box Screen w/16 Configurable Animation
- Frames Programmable Password Lock (Project) Reprogrammable passcode
- lock rail logic Tracktronics:How to Build a ComputerUsing Minecart Tracks

server with a password door (Project) PvP Coded Door

F.5 Circuit Works

- Sea War Minigame with logic gates!
- Primary Cell Automaton (Showcase) elemenT - Elementary cellular
- 16-bit computer (Showcase) TerraByte,the 16 Bit Programmable Computer
- 2D printer (Showcase) 2D Printer
- Play Duck Duck Dart! (Duck Hunt in Terraria?)
- Nim mini-game Nim in Terraria
- [Showcase] Bad Apple!!
- 计算器 [Showcase] A calculator using only logic gates! 井字
- 棋 Play Tic Tac Toe in Terraria!
- Official golf map Journey's End Golf Official Map Starter Kit

8 位 CPU 8-bit CPU v2.0

F.6 Architectural works

Run cool mini-game Gotta Go Fast!

大型建筑 [Project] Castlevania SOTN. A To-Scale Replica of Dracula’s Castle.

- Large building Spaceship "HOPE" wiring
- flush toilet Down the Toilet official
- puzzle map Tales of the Terrarian
  - How It's Made: Tales of the Terrarian
  - Power Generator - Tales of the Terrarian
  - Combining Player Sensors - Tales of the Terrarian
  - Shooting Minigame - Tales of the Terrarian

F.7 Farm

  - Mechanical Boss - Tales of the Terrarian 戈德堡机
- 械 Rube Goldberg Machine

F.7 Farm

- 全自动开荒 Mission Impossible: Expert Mode Automated Playthrough (AFK from PreHardmode to Moon Lord) 采沙场 Automated Sand Factory - Useless Farm? 松露虫
- 农场 [Showcase] Truffle Worm Autofarm: 390 Worms/hr + 50 Plat/hr (Bonus: Best Way to Farm Dev Armor)
- Moon Event Speed Killing Site (Showcase) Lunar Event Speedrun Farm
- Improved Moon Event Kill Site (Showcase) Improved Lunar Event Speedkill
- Farm Ladder Gods (Showcase) 1.3 Fro/Pumpkin New Records Moon New
- Records Arena Coral Farm Super-Compact Semi-automatic coral farm
- (Buggy).

  Fully automatic Hell Brush Monster Field (including Flesh Wall)(Showcase) Recursive Wall of Flesh AFK Autofarm

  Underworld Mob Grinder Cobweb Farm
- Automated Harvest Cobweb Farm
- Fight with 14 boss AFKs at the same time (Showcase) The Ultimate Battle: 14 Bosses Fought Simultaneously AFK in 1.3 Expert Mode

- Fully automatic jungle brush monster fields (including flowers of the century and stone giants)(Showcase) Expert Mode Temple AFK

  Autofarm with Recursive Golem and Plantera Autofarming
- Fully automatic moon events (Project) Expert Mode Recursive Fully-Automated Lunar Event AutoFarm

F.8 Other works

- Transfer gun acceleration Fastest Terrarian ever! (portal
- gun speed boosts) Half-brick script Automating the
- Construction of Hoiks phishing script Automating Fishing
- 1.2 Previous circuit technology , Showcase, All Wiring Systems in Custom Maps

Appendix person's name code

This appendix contains the forum account number for the person's name mentioned in this document, as well as the authors referenced in the writing of this document and their contributions. If your name or links are missing here, welcome to the Issues report on the project's home page or submit a change directly through Pull request.

- putianyi888 : bilibili | Official forum | Baidu paste bar | Chinese forum
- | YouTube | GitHub usdanger : bilibili | Youku | Baidu paste bar |
- YouTube dcfhft : bilibili | Baidu paste bar | Chinese Forum
- Slaughterhouse:bilibili | Chinese forum
- Sleeping:bilibili
- White Frost Heart:bilibili | Youku | Baidu paste it
- TNoName : bilibili | Chinese Forum
- Machine Brother: Youku
- WIAADC : bilibili | Mailbox | QQ : 2607989335
- ZeroGravitas: Official Forum | YouTube
- Programmatic: Official Forum | YouTube
- TheRedStoneCrafter: Official Forum | YouTube ekinator:
- Official Forum
- DRKV: Official Forum
- DicemanX: Official Forum | YouTube
- MappyGaming:YouTube Kaiser Yoshi: Official

Forum