1. 确定电梯调度算法

根据学习和平时的观察积累，我们知道电梯的调度算法有许多种，可以粗略的分为传统电梯调度算法和实时电梯调度算法两类，各自分类下均有多种细化方案，多种方案各有利弊。

传统电梯调度算法可分为先来先服务算法（FCFS）、最短寻找楼层时间优先算法（SSTF）、扫描算法（SCAN）、LOOK 算法、SAFT 算法等几种。

FCFS算法的核心在于随即服务，其逻辑简单，实现容易。但是在上下楼用户较多（即电梯繁忙）的情况下，电梯的效率会受到严重影响，该种算法只能满足电梯的使用频率不太高时的情况。

SSTF算法注重优化缩短电梯寻找楼层的时间，因此优先响应附近楼层。在电梯不太繁忙的情况下，其效率较高，但是在电梯繁忙的时候，部分楼层可能会出现“久等不应”的情况，不利于部分用户的体验。

SCAN算法的核心在于扫描，电梯循环往返于最高层与最底层，运行过程中响应最近楼层的同向请求。该种算法由于电梯一直在运动，有效的减少了SSTF算法中潜在的“久等不应”问题，但是在电梯不太繁忙时，连续的运动有耗电嫌疑。

LOOK算法在SCAN算法的基础上增加了一个控制，即在电梯运行方向上无用户请求时，电梯直接反向。这一项控制一定程度上提高了效率。

SAFT算法以SSTF算法为基础，进行了拓展和完善，考虑到了乘客进出电梯的时间对整体运行的影响。

实时电梯调度算法可分为最早截止期优先调度算法（EDF）、SCAN-EDF 算法、PI 算法、FD-SCAN 算法等。

实时电梯调度算法以传统电梯调度算法为基础，在相关算法的基础上更强调实时性以及优先级问题。除此之外，随着控制手段以及软硬件的不断发展，也出现了基于专家系统、模糊逻辑、遗产算法以及模糊神经网络等的电梯群控方法。

在获得资料和自身经验的基础上，我们讨论出了如下电梯调度方案：

1. 电梯静止，位于x楼

响应所有上、下行请求，记请求楼层数为y

x>y 下行

x<y 上行

x=y 静止在原地，电梯开门

1. 电梯运行，目前位于x楼，目标楼层为y（x≠y），此时z楼发出请求

①指令来自电梯内

1. x<y 电梯上行
2. x≤z≤y 响应z楼
3. z<x 或y<z 暂不响应
4. x>y 电梯下行
5. y≤z≤x 响应z楼
6. z>x 或y>z 暂不响应

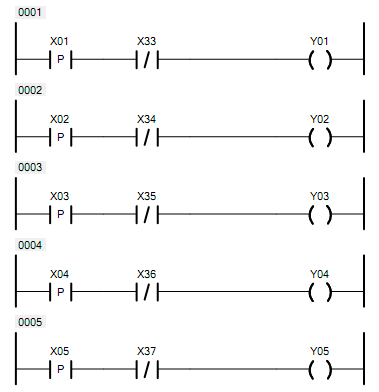
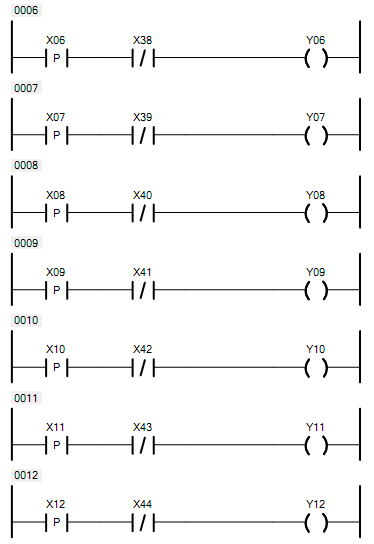
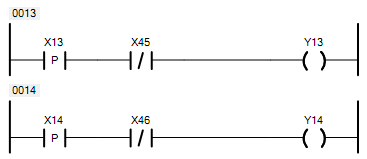
②指令来自电梯外

1. x<y 电梯上行
2. x≤z≤y且请求上行 响应z楼
3. x≤z≤y且请求下行 暂不响应
4. z<x 或y<z 暂不响应
5. x>y 电梯下行
6. y≤z≤x且请求下行 响应z楼
7. y≤z≤x且请求上行 暂不响应
8. z>x 或y>z 暂不响应
9. 其余响应均以时间顺序为准
10. 简单梯形图设计

首先，根据已有面板定义硬件接口，如下表：

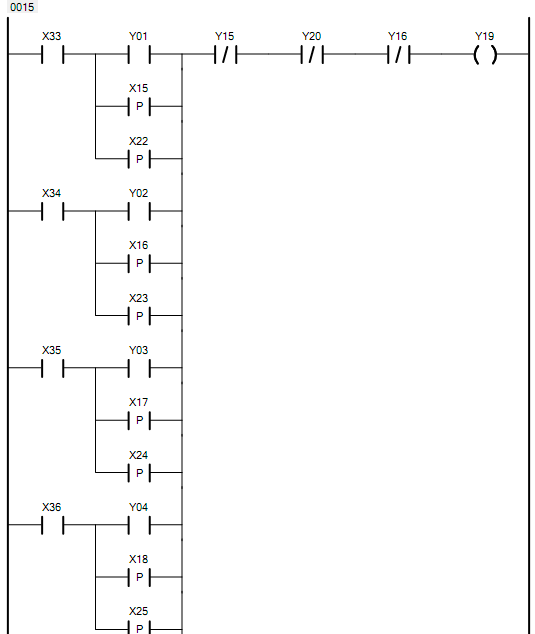
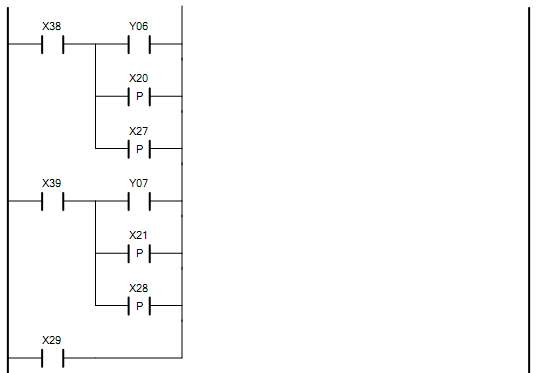
|  |  |  |  |
| --- | --- | --- | --- |
| 电梯与PLC接口对应关系 | | | |
| 电梯输入接口 | PLC输入 | 电梯输出接口 | PLC输出 |
| 左边电梯1楼按键 | X0001 | 左边电梯1楼按键LED | Y0001 |
| 左边电梯2楼按键 | X0002 | 左边电梯2楼按键LED | Y0002 |
| 左边电梯3楼按键 | X0003 | 左边电梯3楼按键LED | Y0003 |
| 左边电梯4楼按键 | X0004 | 左边电梯4楼按键LED | Y0004 |
| 左边电梯5楼按键 | X0005 | 左边电梯5楼按键LED | Y0005 |
| 左边电梯6楼按键 | X0006 | 左边电梯6楼按键LED | Y0006 |
| 左边电梯7楼按键 | X0007 | 左边电梯7楼按键LED | Y0007 |
| 右边电梯1楼按键 | X0008 | 右边电梯1楼按键LED | Y0008 |
| 右边电梯2楼按键 | X0009 | 右边电梯2楼按键LED | Y0009 |
| 右边电梯3楼按键 | X0010 | 右边电梯3楼按键LED | Y0010 |
| 右边电梯4楼按键 | X0011 | 右边电梯4楼按键LED | Y0011 |
| 右边电梯5楼按键 | X0012 | 右边电梯5楼按键LED | Y0012 |
| 右边电梯6楼按键 | X0013 | 右边电梯6楼按键LED | Y0013 |
| 右边电梯7楼按键 | X0014 | 右边电梯7楼按键LED | Y0014 |
| 1楼上按键 | X0015 | 左边电梯电机正转 | Y0015 |
| 2楼上按键 | X0016 | 左边电梯电机反转 | Y0016 |
| 3楼上按键 | X0017 | 右边电梯电机正转 | Y0017 |
| 4楼上按键 | X0018 | 右边电梯电机反转 | Y0018 |
| 5楼上按键 | X0019 | 左边电梯门电机正转 | Y0019 |
| 6楼上按键 | X0020 | 左边电梯门电机反转 | Y0020 |
| 7楼上按键 | X0021 | 右边电梯门电机正转 | Y0021 |
| 1楼下按键 | X0022 | 右边电梯门电机反转 | Y0022 |
| 2楼下按键 | X0023 |  |  |
| 3楼下按键 | X0024 |  |  |
| 4楼下按键 | X0025 |  |  |
| 5楼下按键 | X0026 |  |  |
| 6楼下按键 | X0027 |  |  |
| 7楼下按键 | X0028 |  |  |
| 左边电梯开门 | X0029 |  |  |
| 左边电梯关门 | X0030 |  |  |
| 右边电梯开门 | X0031 |  |  |
| 右边电梯关门 | X0032 |  |  |
| 左边电梯1楼行程开关 | X0033 |  |  |
| 左边电梯2楼行程开关 | X0034 |  |  |
| 左边电梯3楼行程开关 | X0035 |  |  |
| 左边电梯4楼行程开关 | X0036 |  |  |
| 左边电梯5楼行程开关 | X0037 |  |  |
| 左边电梯6楼行程开关 | X0038 |  |  |
| 左边电梯7楼行程开关 | X0039 |  |  |
| 右边电梯1楼行程开关 | X0040 |  |  |
| 右边电梯2楼行程开关 | X0041 |  |  |
| 右边电梯3楼行程开关 | X0042 |  |  |
| 右边电梯4楼行程开关 | X0043 |  |  |
| 右边电梯5楼行程开关 | X0044 |  |  |
| 右边电梯6楼行程开关 | X0045 |  |  |
| 右边电梯7楼行程开关 | X0046 |  |  |
| 左边电梯门开行程开关 | X0047 |  |  |
| 左边电梯门关行程开关 | X0048 |  |  |
| 右边电梯门开行程开关 | X0049 |  |  |
| 右边电梯门关行程开关 | X0050 |  |  |

首先在梯形图中定义按钮上LED显示状态：

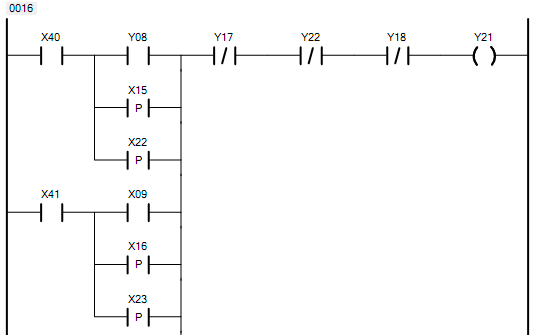
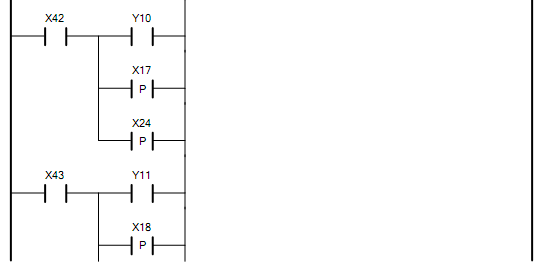
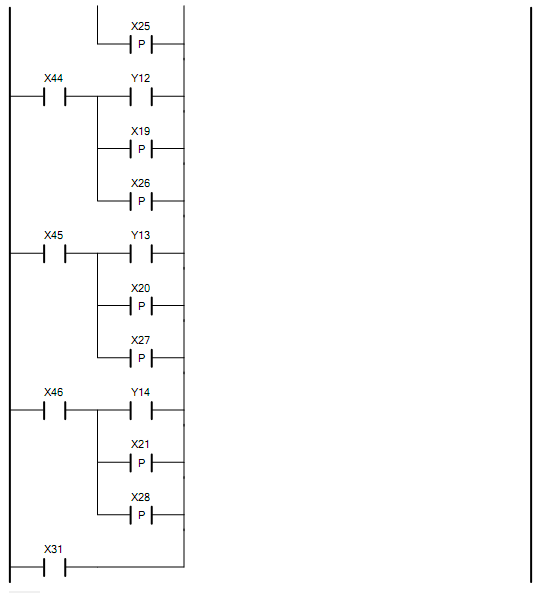
其基本逻辑是：按下电梯内按键，如果此时电梯不处于按键对应楼层，则LED灯亮，一旦电梯运行到对应楼层，LED灯灭。

接下来进行电梯门的控制，以左边电梯门开启指令的触发为例：

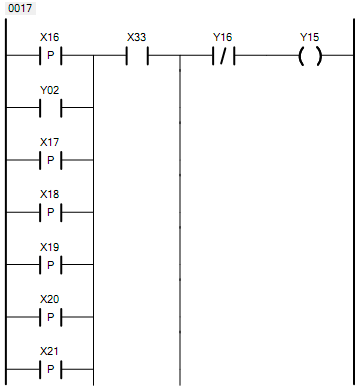
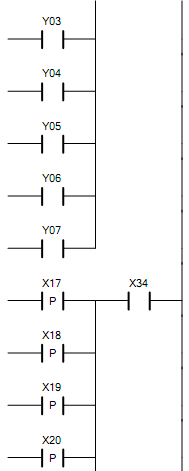
  

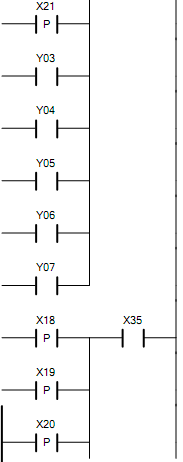
其基础逻辑为：当电梯运行到某一楼层，电梯电机停止，且电梯不处于正在关门状态下，若该楼层为电梯内按键对应目标楼层，或者该楼层有用户发出上行或下行请求时，电梯门控制开启，响应请求。若此时电梯电机停止且电梯门不处于正在关闭的状态，按下电梯内开门按键，也可以开门。

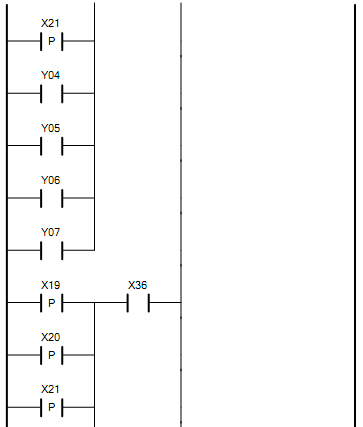
右边电梯开门的情况同理：

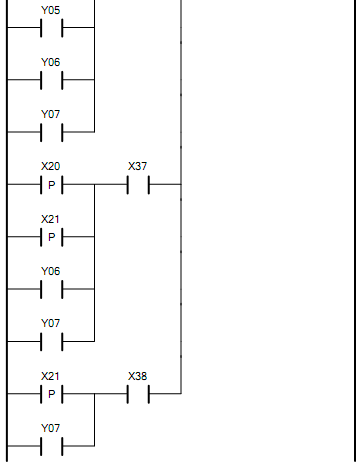
  

对于电梯电机运行情况，梯形图内容较多，仅以电梯停在某层楼，响应其他楼层请求或者电梯内用户按键请求为例：

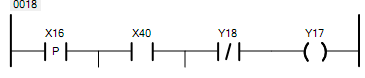
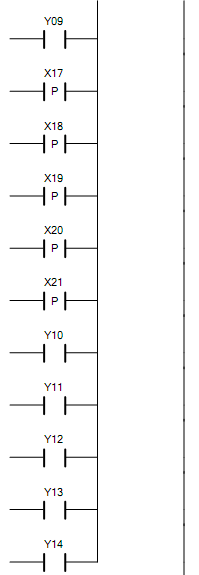


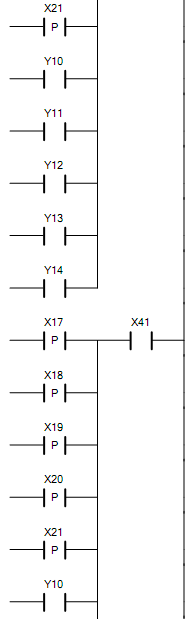


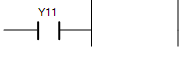


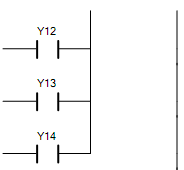
其基本思路是当电梯位于某一楼层，更高楼层有用户发出请求或者电梯内用户发出目标为高楼层的请求时，左边电梯电机正转。

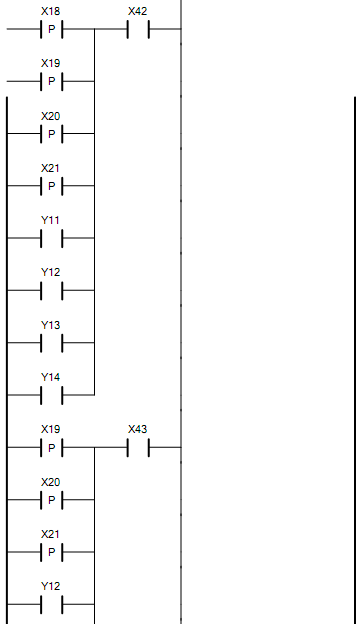
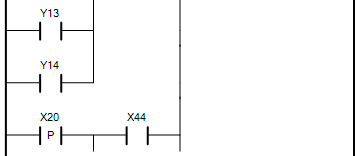
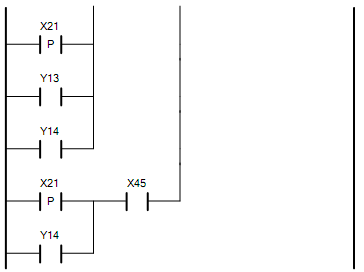
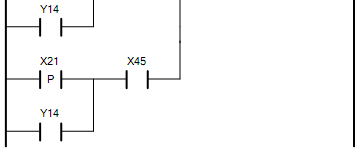
右边同理：

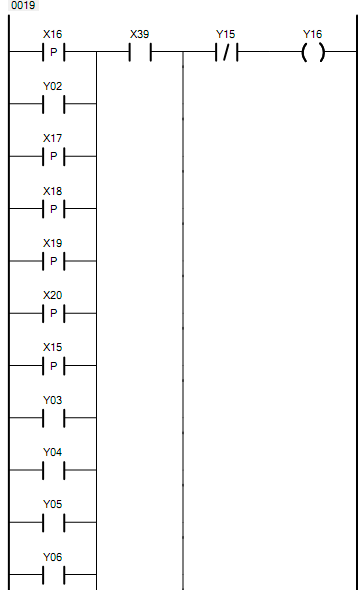
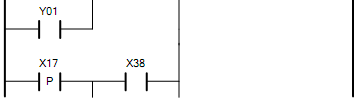
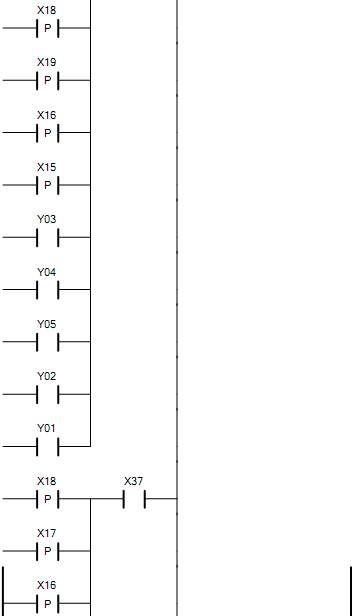
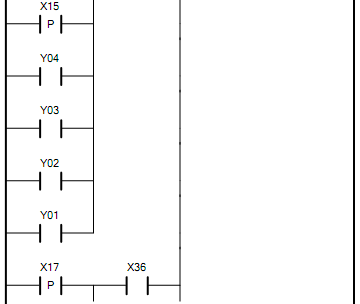
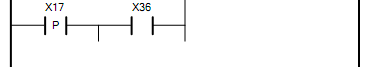
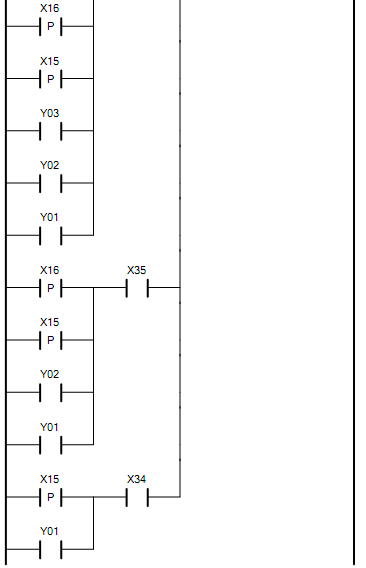




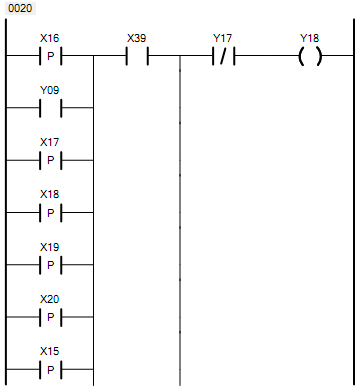
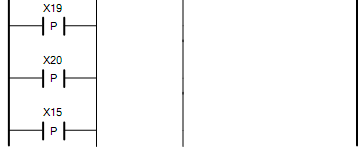
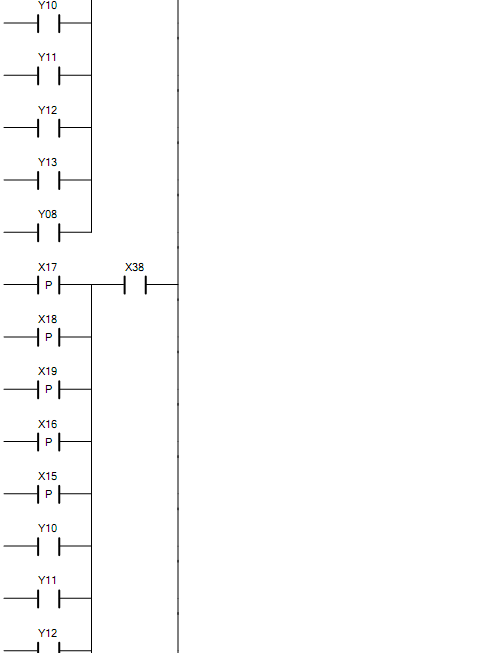
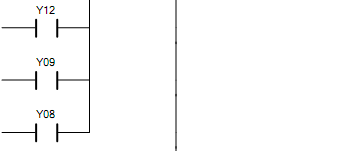
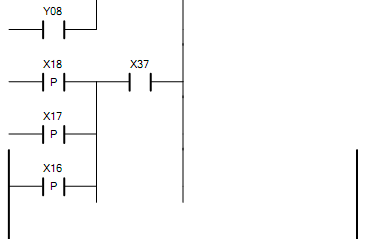
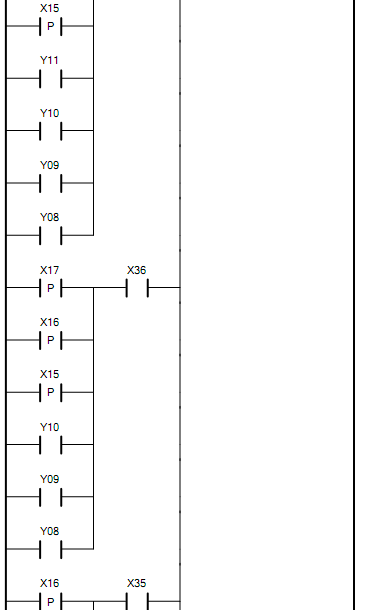
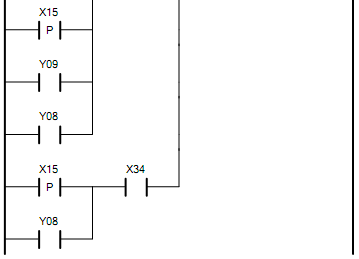


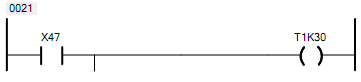
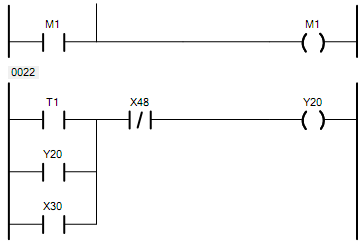
对于电机反转，即电梯下行，考虑的情况与上行同理，即电梯位于某一层楼，响应低楼层请求。

右边电梯同理：

最后考虑电梯门的自动关闭，当电梯门完全开启，行程开关被触发后，借助计时器进行一段延时，然后电梯门电机反转，电梯门自动关闭，电梯门关闭的行程开关被触发后，电梯门电机停止。

右边电梯同理：

