

Tanimodori 的 \LaTeX 备忘录

v 0.1.0

Tanimodori

2020 年 7 月 10 日

0 约定

- \TeX 表示内容属于由 Plain \TeX
- \LaTeX 表示内容属于 \LaTeX
- PKG 表示内容属于宏包
- MACRO 表示内容属于本文中的宏

1 基本

1.1 L^AT_EX 文档基本格式

```
% !TEX TS-program = xelatex
% !TEX encoding = UTF-8
\XeTeXlinebreaklocale "zh"
\XeTeXlinebreakskip = 0pt plus 1pt      % 编译引擎选项、编码
\documentclass[12pt]{article}           % 全局选项、文档属性
\usepackage{xCJK,fontspec}              % 宏包及其选项
\usepackage{graphicx}
\usepackage{amsmath,amssymb,amsthm}
\title{\LaTeX\ Test}                   % 标题
\author{\LaTeX\ User}                  % 作者
\date{yyyy.mm.dd}                      % 日期
\begin{document}                        % 导言区结束，进入正文
\maketitle
\section{Foo}                           % 节
\subsection{Bar}                        % 小节
The quick fox jumps over the lazy dog. % 正文
\end{document}                          % 正文结束
```

1.2 空格

一次 L^AT_EX 源文件编译过程中的主要步骤包括：

1. 前端编辑器（例如 TeXWorks）根据源文件和用户设置交给对应的引擎（例如 X_YL^AT_EX）
2. 引擎将源文件令牌化
3. 引擎根据令牌序列输出结果。

在令牌化过程中，单个空格、多个空格或者单个换行符会被看作一个空格令牌，多个换行符会被看做换行令牌。但是控制字¹末尾的空格字符会被忽略，不会作为空格令牌。还需加上 “\ ” 来插入一个控制空格。控制换行符可以通过 “\\ ” 插入

L^AT_EXwithout space
L^AT_EX with space!

\LaTeX\ without space\\
\LaTeX\ with space!

我们用尺度（*dimension*）一词来描述空格的长度。尺度由“实数 + 单位”构成，例如“5pt”、“-1.1 em”、“+,6cm”都是合法的尺度。在 T_EX 中，可以通过 \hskip <dimen> 、\vskip <dimen> 插入水平和垂直空格。L^AT_EX 的对应命令为 \hspace{<dimen>} 和 \vspace{<dimen>}。一般来说，由于历史遗留问题，我们应尽量使用 L^AT_EX 包装好的命令，避免使用 T_EX 命令。

能够产生水平空格的常见指令见下表。

¹Control Word，由反斜杠和字母序列组成，例如 “\LaTeX”，是控制序列的一种。”

空格宽度	文本模式		数学模式	
	代码	示例	代码	示例
.16667em 或 3mu	a\,b	a b	$\$a\,,b\$$	$a\,b$
	a\thinspace b	a b	$\$a\,\thinspace b\$$	$a\,b$
-3mu			$\$a\,!b\$$	ab
			$\$a\mkern-\thinmuskip b\$$	ab
4.0mu plus 2.0mu minus 4.0mu			$\$a\>b\$$	$a\,b$
			$\$a\!:b\$$	$a\,b$
			$\$a\mkern\medmuskip b\$$	$a\,b$
5.0mu plus 5.0mu			$\$a\;;b\$$	$a\,b$
			$\$a\mkern\thickmuskip b\$$	$a\,b$
.5em	a\enspace b	a b	$\$a\,\enspace b\$$	$a\,b$
1em	a\quad b	a b	$\$a\,\quad b\$$	$a\,b$
2em	a\qquad b	a b	$\$a\,\qquad b\$$	$a\,b$
<len>	a\hskip 1em b	a b	$\$a\,\hskip 1em b\$$	$a\,b$
	a\kern 1pc b	a b	$\$a\,\kern 1pc b\$$	$a\,b$
	a\hspace{25pt} b	a b	$\$a\,\hspace{25pt} b\$$	$a\,b$
<stuff>	axyzb	axyzb	$\$axyzb\$$	$axyzb$
	a\hphantom{xyz}b	a b	$\$a\,\hphantom{xyz}b\$$	$a\,b$
inter-word	a{ }b	a b	$\$a\{ \}b\$$	ab
	a\space b	a b	$\$a\,\space b\$$	ab
control	a\ b	a b	$\$a\,\,b\$$	$a\,b$
unbreakable	a~b	a b	$\$a\sim b\$$	$a\,b$
rubber	a\hfill b	a b	$\$a\,\hfill b\$$	$a\,b$
	a\hspace{\fill} b	a b	$\$a\,\hspace{\fill} b\$$	$a\,b$

其中mu是数学模式下的单位长度。<dimen> plus <dimen> minus <dimen> 除了定义空格的长度以外,还定义了空格的伸缩量。\\hphantom会提供与参数水平尺寸相同的空格。“~”、\\hspace*和\\vspace*产生的空格是不换行空格,也就是说L^AT_EX不会在此处换行。\\hfill、\\vfill (T_EX)、\\hspace{\\fill}、\\vspace{\\fill} (L^AT_EX) 将产生填满剩余空间的空格,若有多个,则按照比例分配,例如:

左对齐	右对齐	左对齐 <code>\\hspace*{\\fill}\\</code>
		<code>\\hspace*{\\fill}</code> 右对齐 <code>\\</code>
居中		<code>\\hspace*{\\fill}</code> 居中 <code>\\hspace*{\\fill}</code>

1.3 字体

1.3.1 指定字体

我们首先需要导入xeCJK和fontspec宏包，然后才能指定中英文字体。

```
% 导言区
\usepackage{xeCJK,fontspec}
% 设置英文宋体、黑体、等宽字体
\setmainfont{Source Han Serif SC}
\setsansfont{Source Han Sans SC}
\setmonofont{Sarasa Mono SC}
% 设置中文宋体、黑体、等宽字体
\setCJKmainfont{Source Han Serif SC}
\setCJKsansfont{Source Han Sans SC}
\setCJKmonofont{Sarasa Mono SC}
```

我们还可以给字体指定伪斜体与伪粗体

```
\setmainfont[AutoFakeSlant=0.2, AutoFakeBold=1.5]{Source Han Serif SC}
```

1.3.2 指定字形

命令	等价命令	效果
<code>\textnormal{...}</code>	<code>{\normalfont ...}</code>	sample text 示例文本
<code>\emph{...}</code>	<code>{\em ...}</code>	<i>sample text</i> 示例文本
<code>\textrm{...}</code>	<code>{\rmfamily ...}</code>	sample text 示例文本
<code>\textsf{...}</code>	<code>{\sffamily ...}</code>	sample text 示例文本
<code>\texttt{...}</code>	<code>{\ttfamily ...}</code>	sample text 示例文本
<code>\textup{...}</code>	<code>{\upshape ...}</code>	sample text 示例文本
<code>\textit{...}</code>	<code>{\itshape ...}</code>	<i>sample text</i> 示例文本
<code>\textsl{...}</code>	<code>{\slshape ...}</code>	<i>sample text</i> 示例文本
<code>\textsc{...}</code>	<code>{\scshape ...}</code>	不支持
<code>\textbf{...}</code>	<code>{\bfseries ...}</code>	sample text 示例文本
<code>\textmd{...}</code>	<code>{\mdseries ...}</code>	sample text 示例文本
<code>\textlf{...}</code>	<code>{\lfseries ...}</code>	不支持

1.3.3 指定字体大小

命令	效果
<code>\tiny</code>	sample text 示例文本
<code>\scriptsize</code>	sample text 示例文本
<code>\footnotesize</code>	sample text 示例文本
<code>\small</code>	sample text 示例文本
<code>\normalsize</code>	sample text 示例文本
<code>\large</code>	sample text 示例文本
<code>\Large</code>	sample text 示例文本
<code>\LARGE</code>	sample text 示例文本
<code>\huge</code>	sample text 示例文本
<code>\Huge</code>	sample text 示例文本

1.3.4 指定字体颜色 PKG

red text
blue text

```
\usepackage{xcolor} % 导言区
{\color{red} red text}\\
\textcolor{blue}{blue text}
```

2 版式与结构 PKG

2.1 页边距

```
% 引言区
\usepackage{geometry}
\geometry{left=1.5cm,right=1.5cm,top=1.5cm,bottom=2cm,a4paper}
```

2.2 日期

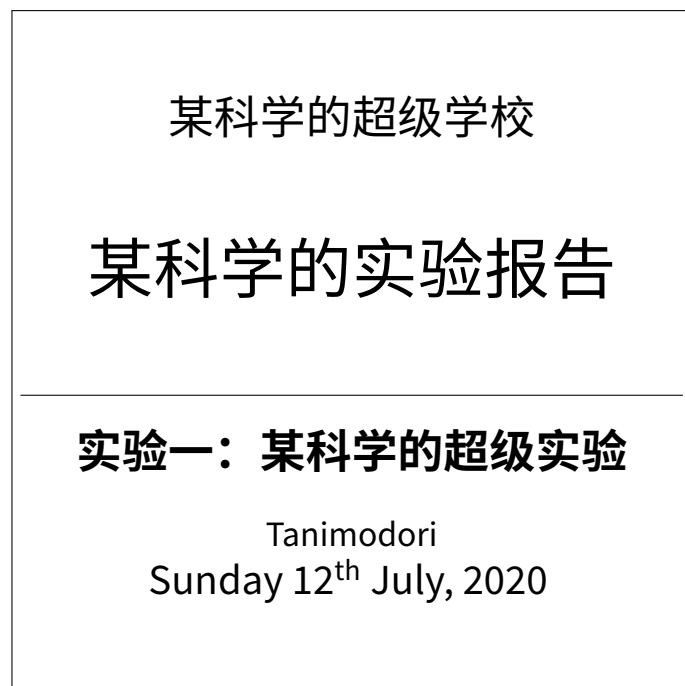
2020.07.12 2020 年 7 月 12 日

```
\usepackage{datetime} % 导言区
\renewcommand{\dateseparator}{.}
\yyyymmdddate
\today % yyyy.mm.dd
\newdateformat{cndate}{\THEYEAR 年\THEMONTH 月
↪ \THEDAY 日}
\cndate\today % yyyy年mm月dd日
```

2.3 分割线

```
\noindent\rule{\textwidth}{0.4pt}
```

2.4 自定义封面



```
\begin{document}
  \begin{titlepage}
    \sffamily
    \centering
    \vspace{1cm}
    {\Large 某科学的超级学校\par}
    \vspace{0.8cm}
    {\Huge 某科学的实验报告\par}
    \vspace{0.4cm}
    \noindent\rule{\textwidth}{0.4pt}
    {\bfseries\Large
      实验一：某科学的超级实验\par}
    \vspace{0.4cm}
    {Tanimodori}
    \vfill
    {\large \today}
    \vspace{1.2cm}
  \end{titlepage}
\end{document}
```

2.5 目录

通常结构

```
\renewcommand{\contentsname}{\centering 目录}
\begin{document}
\tableofcontents
\thispagestyle{empty} % 禁用页码
\newpage
\setcounter{page}{1}
\end{document}
```

带超链接的目录

```
\usepackage[CJKbookmarks]{hyperref}
\hypersetup{
  colorlinks=true,
  linktoc=page,
  linkcolor=blue
}
\begin{document}
\tableofcontents
\end{document}
```

2.6 参考文献

通常结构

```

\usepackage{hyperref}
\usepackage[hyperref=true,backend=biber,sorting=none,backref=false,doi=false,isbn=false,url=false]{
  ↪ biblatex}
\addbibresource{example.bib}
% 引言区
\begin{document}
Foo\parencite{Foo_Bar_2020}
\printbibliography[heading=none]
\end{document}

```

上标方括号参考

```

% 引言区
\DeclareCiteCommand{\supercite}[\mkbibsuperscript]
  {\iffieldundef{prenote}
    {}
    {\BibliographyWarning{Ignoring prenote argument}}}%
  {\iffieldundef{postnote}
    {}
    {\BibliographyWarning{Ignoring postnote argument}}}%
  {\usebibmacro{citeindex}}%
  {\bibopenbracket\usebibmacro{cite}\bibclosebracket}
  {\supercitedelim}
  {}

```

3 数学模式

下花括号

$$1 \underbrace{0 \cdots 0}_{10 \text{ zeros}}$$

```
\[1\underbrace{0\cdots0}_{\text{10 zeros}}\]
```

分类

$$x = \begin{cases} 1, & y = 1 \\ 2, & y = 2 \\ 0, & \text{otherwise.} \end{cases}$$

```

\[x=\begin{cases}
  1,&y=1\\
  2,&y=2\\
  0,&\text{otherwise.}
\end{cases}\]

```

矩阵

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

```

\[A=\begin{bmatrix}
  1&2\\
  3&4
\end{bmatrix}\]

```

等式

$$T_0(k) = \frac{1}{2}T_0(k-1) \quad (1)$$

```
\begin{equation}
T_{\{0\}}(k)=\frac{1}{2} T_{\{0\}}(k-1)
\end{equation}
```

连等式

$$\begin{aligned} y &= (x+1)(x-1) \\ &= x^2 - 1 \end{aligned}$$

```
\[\begin{aligned}
y&=(x+1)(x-1)\\
&=x^2-1
\end{aligned}\]
```

4 图片

4.1 位图、矢量图、PDF

```
\usepackage{graphicx}
\includegraphics[width=1\linewidth]{example.png}
\includegraphics[width=1\linewidth]{example.svg}
\includegraphics[width=1\linewidth]{example.pdf}
```

5 表格

列表

- item 1
- item 2

```
\begin{itemize} % 无序列表
\item item 1
\item item 2
\end{itemize}
```

1. item 1
 - (a) Nested item 1
 - (b) Nested item 2
2. item 2

```
\begin{enumerate} % 有序列表
\item item 1
\begin{enumerate} % 嵌套列表
\item Nested item 1
\item Nested item 2
\end{enumerate}
\item item 2
\end{enumerate}
```

Apple item 1

Butter item 2

Charlie item 3

```
\begin{description} % 描述列表
\item [Apple] item 1
\item [Butter] item 2
\item [Charlie] item 3
\end{description}
```


tabular

十进制数	4-3-2-1 编码
0	0000
1	0001
2	0010
3	0100
4	1000
5	1001
6	1010
7	1100
8	1101
9	1110

```
\begin{center}
\begin{tabular}{|1|1|1|1|}
\hline
十进制数& 4-3-2-1编码 \\\hline
0      & 0000 \\\hline
1      & 0001 \\\hline
2      & 0010 \\\hline
3      & 0100 \\\hline
4      & 1000 \\\hline
5      & 1001 \\\hline
6      & 1010 \\\hline
7      & 1100 \\\hline
8      & 1101 \\\hline
9      & 1110 \\\hline
\end{tabular}
\end{center}
```

多行与多列 PKG

现态	次态		当前输出
	$X = 0$	$X = 1$	
a	c	c	1
c	d	f	0
d	f	a	1
f	c	d	0

```
\usepackage{multirow} %导言区
\begin{center}
\begin{tabular}{c|cc|c}\hline
\multirow{2}{*}{现态}&\multicolumn{2}{|c|}{次态}
↪ &\multirow{2}{*}{当前输出}\\\cline{2-3}
~&X=0&X=1&~\\\hline
a&c&c&1\\
c&d&f&0\\
d&f&a&1\\
f&c&d&0\\\hline
\end{tabular}
\end{center}
```

6 代码 PKG

Matlab

```
function xbar=mymean(x)
    t=0;
    n=length(x);
    for i=1:n
        t=t+x(i);
    end
    xbar=t/n;
end
```

```
\usepackage{listings}
\usepackage{matlab-prettifier} % 导言区
\begin{lstlisting}[style=Matlab-editor]
function xbar=mymean(x)
    t=0;
    n=length(x);
    for i=1:n
        t=t+x(i);
    end
    xbar=t/n;
end
\end{lstlisting}
```

x86 汇编

```
data segment
s db 'Hello World!$'
data ends

code segment
assume cs:code, ds:data
start: mov ax, data
mov ds, ax

lea dx, s
mov ah, 09H
int 21H

mov ax, 4C00H
int 21H

code ends
end start
```

```
\usepackage{listings} % 导言区
\lstdefinestyle{customasm}{
language=[x86masm]Assembler,
basicstyle=\linespread{1.0}\small\ttfamily,
commentstyle=\color{Brown},
keywordstyle=\color{blue}\bfseries,
stringstyle=\color{Green}\bfseries,
keepspaces=true,
columns=fixed,
basewidth=0.5em,
}
\lstinputlisting[style=customasm]{x86asm_example.asm}
```

C

```

int main(int argv, char* argc[]) {
    for (int j = 0; j < 200; ++j) {
        int t, p, found;
        fscanf(input, "(%d,%d) ", &t, &
↪ p);
        found = mem_find(p);
        if (found != -1) {
            ++hit;
            fprintf(output, "%c", mem[
↪ found].data[t]);
        } else {
            found = mem_find_empty();
            mem_load(&mem[found], disk,
↪ p);
            fprintf(output, "%c", mem[
↪ found].data[t]);
        }
    }
    return 0;
}

```

```

\usepackage{listings} % 导言区
\lstdefinestyle{customc}{
    language=C,%
    keepspaces=true,%
    columns=fixed,%
    basewidth=0.5em,%
    basicstyle=\ttfamily,%
    breaklines=true,%
    commentstyle=\color{Green},%
    keywordstyle=\color{blue}\bfseries,%
    stringstyle=\color{Brown}\bfseries,%
    numberstyle=\ttfamily\color{Gray},%
    showstringspaces=false%
}
\lstinputlisting[style=customc]{c_example.c}

```

Python

```

#!/usr/local/bin/python
# -*- coding: utf-8 -*-
"""
This is an example module
"""
class Example():
    """
    This is an example class
    """

    def __init__(self, arg):
        # This is an example class
        self.data = arg
        if arg == 'foo':
            print('foo')
        else:
            print('bar')

```

```

\usepackage{listings} % 导言区
\lstdefinestyle{custompython}{
    language=Python,
    keepspaces=true,
    columns=fixed,
    basewidth=0.5em,
    basicstyle=\ttfamily,
    breaklines=true,
    commentstyle=\color{Green},
    keywordstyle=\color{blue}\bfseries,
    stringstyle=\color{Brown}\bfseries,
    showstringspaces=false
}
\lstinputlisting[style=custompython]{python_
↪ example.py}

```

diff

```
12 -   int i;
13 -   for (i = 0; i < N; i++)
14 -       pthread_create(&tid[i], NULL,
    ↪ thread, &i);
12 +   int i, args[N];
13 +   for (i = 0; i < N; i++) {
14 +       args[i] = i;
15 +       pthread_create(&tid[i], NULL,
    ↪ thread, args + i);
16 +   }
```

```
\usepackage{xcolor}
\definecolor{diffstart}{named}{gray}
\definecolor{diffincl}{named}{Green}
\definecolor{diffrem}{named}{Red}
\usepackage{listings}
\lstdefinestyle{customdiff}{
    language=C,
    keepspaces=true,
    columns=fixed,
    basewidth=0.5em,
    basicstyle=\ttfamily,
    breaklines=true,
    commentstyle=\color{Green},
    keywordstyle=\color{blue}\bfseries,
    stringstyle=\color{Brown}\bfseries,
    numberstyle=\ttfamily\color{Gray},
    showstringspaces=false,
    morecomment=[f][\color{diffstart}]{@@},
    morecomment=[f][\color{diffincl}]{+\ },
    morecomment=[f][\color{diffrem}]{-\ },
    numbers=left,
    stepnumber=1
}
% LISTINGS linenumber hack
\makeatletter
\let\orig@lstnumber=\the\lstnumber
\newcommand\lstsetnumber[1]{\gdef\the\lstnumber
    ↪ {#1}}
\newcommand\lstresetnumber{\global\let
    ↪ \the\lstnumber=\orig@lstnumber}
\makeatother
% 导言区
\begin{lstlisting}[style=customdiff,firstnumber
    ↪ =12, mathescape=true]
-   int i;
-   for (i = 0; i < N; i++)
-       pthread_create(&tid[i], NULL, thread, &i)
    ↪ ;$\lstresetnumber\setcounter{lstnumber
    ↪ }{11}$
+   int i, args[N];
+   for (i = 0; i < N; i++) {
+       args[i] = i;
+       pthread_create(&tid[i], NULL, thread,
    ↪ args + i);
+   }
\end{lstlisting}
```

终端

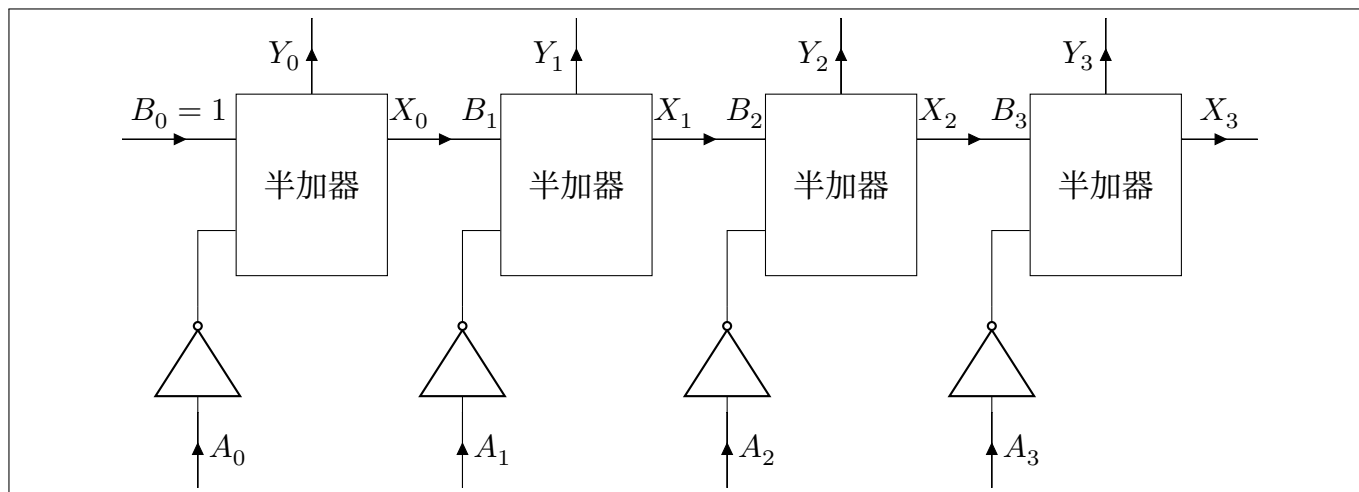
A small rectangular window with a black background and a white border. Inside, the text "C:\>" is written in white, representing a command prompt.

```
\usepackage{listings}
\lstdefinestyle{terminal}{
  backgroundcolor=\color{black},
  basicstyle=\ttfamily\color{white},
  numbers=none,
  columns=fixed,
  basewidth=0.5em
} % 导言区
\begin{lstlisting}[style=terminal]
C:\>
\end{lstlisting}
```

7 Tikz PKG

7.1 命令示例

\foreach



```

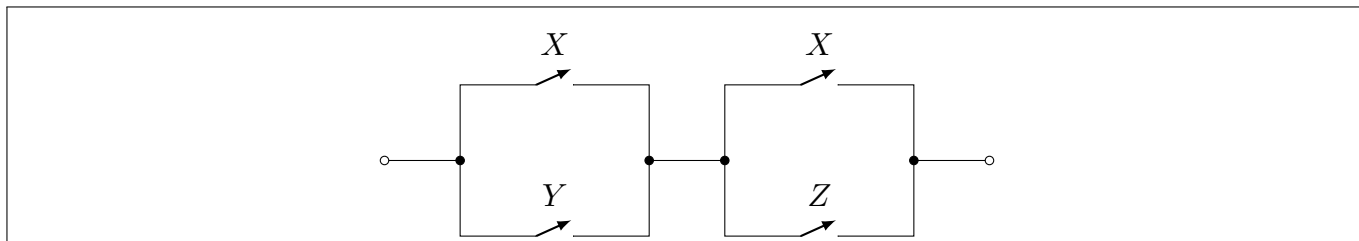
\begin{center}
\begin{tikzpicture}[>=latex]
\foreach \x in {1, 2, 3, 4} {
  \pgfmathtruncatemacro{\xminus}{\x-1}
  \draw
    (\xminus*3.5,0) node[draw,minimum width=2cm,minimum height=2.4cm] (ha\x) {半加器}
    ($ (ha\x.west)!0.5!(ha\x.south west)$) coordinate (ha\x-a)
    ($ (ha\x.west)!0.5!(ha\x.north west)$) coordinate (ha\x-b)
    ($ (ha\x.east)!0.5!(ha\x.north east)$) coordinate (ha\x-x)
    ($ (ha\x.north)$) coordinate (ha\x-y)
    (ha\x-a) -| ++ (-0.5,-1) node[not port, anchor=out, rotate=90] (ha\x-an) {}
    (ha\x-an.in) to[short,i^<=${A_{\xminus}}$] ++ (0,-1)
    (ha\x-y) to[short,i^>=${Y_{\xminus}}$] ++ (0,1);
}
\foreach \x in {1, 2, 3} {
  \pgfmathtruncatemacro{\xminus}{\x-1}
  \pgfmathtruncatemacro{\xplus}{\x+1}
  \draw
    (ha\x-x) to[short,i^>=${X_{\xminus}} \quad B_{\x}$] (ha\xplus-b);
}
\draw
  (ha1-b) to[short,i^<=${B_0=1}$] ++ (-1.5,0)
  (ha4-x) to[short,i^>=${X_3}$] ++ (1,0);
\end{tikzpicture}
\end{center}

```

7.2 内容示例

7.2.1 数字逻辑设计

开关电路



```
\usepackage{tikz}
\usetikzlibrary{arrows,shapes}
\usepackage[american,RPvoltages]{circuitikz}
% 导言区
\begin{center}
\begin{circuitikz}[>=latex]
\draw
(0,0) to [short,o-*] (1,0) coordinate (left_in) ---+ (0,1)
---+ (1,0) to[nos,->,l=$X$] ++ (0.5,0) ---+ (1,0)
---+ (0,-1) coordinate (left_out) to [short,*-*] ++ (1,0) coordinate (right_start)

(left_in) ---+ (0,-1)
---+ (1,0) to[nos,->,l=$Y$] ++ (0.5,0) ---+ (1,0)
-- (left_out)

(right_start) ---+ (0,1)
---+ (1,0) to[nos,->,l=$X$] ++ (0.5,0) ---+ (1,0)
---+ (0,-1) coordinate (right_end) to [short,*-o] ++ (1,0) coordinate(end)

(right_start) ---+ (0,-1)
---+ (1,0) to[nos,->,l=$Z$] ++ (0.5,0) ---+ (1,0)
-- (right_end);
\end{circuitikz}
\end{center}
```

门电路 MACRO

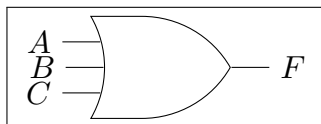
首先导入 xparse 宏包

```
\usepackage{xparse} % 导言区
```

\gatelabel 命令可以为门电路标记输入输出，其用法如下：

```
\gatelabel{gate-name}{{comma-seperated-list}}
\gatelabel*{gate-name}{output-label}
```

注意：该命令只能处理输入在左侧，输出在右侧的门。使用示例如下。



```
\begin{tikzpicture}
\node (or) [or gate US, draw, logic gate inputs=nnn,scale=2] {};
\gatelabel{or}{{$A$, $B$, $C$}}
\gatelabel*{or}{$F$}
\end{tikzpicture}
```

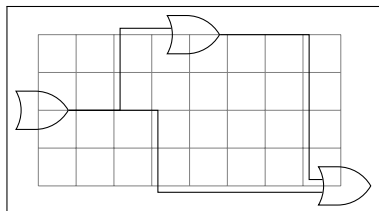
该命令的定义如下:

```
\makeatletter
\newcommand*{\gatelabel@unstarred}[2]{
\foreach \label [count=\li] in #2 {
\draw (#1.input \li) ++ (-0.5,0) node (#1_i_\li) [left] {\label} -- (#1.input \li);
}
}
\newcommand*{\gatelabel@starred}[2]{
\draw (#1.output) -- ++(0.5,0) node (#1_o) [right] {#2};
}
\newcommand*{\gatelabel}{\@ifstar{\gatelabel@starred}{\gatelabel@unstarred}}
\makeatother
```

\gatelink 命令可以为连接两个门之间的输入输出, 其用法如下

```
\gatelink[offset]{anchor1}{anchor2}
```

注意: 该命令只能处理输入在左侧, 输出在右侧的门。使用示例如下。

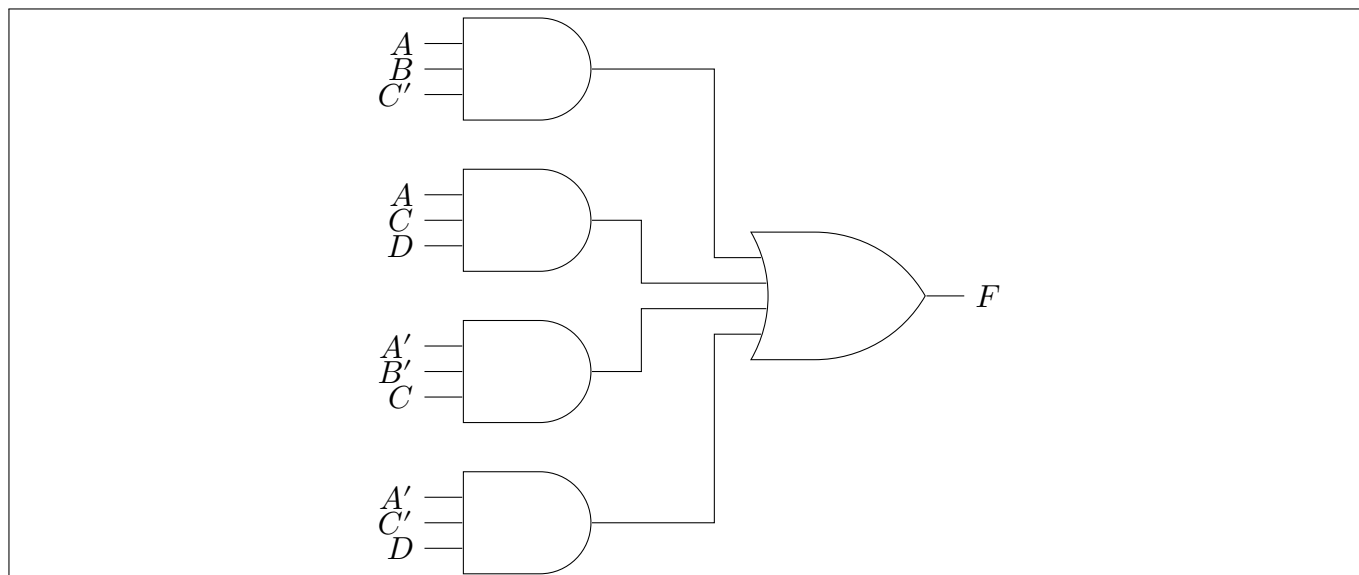


```
\begin{tikzpicture}[every node/.style={or gate US, draw}]
\draw[help lines, step=0.5cm] (0,0) grid (4,2);
\node (or1) at (0,1) {};
\node (or2) at (2,2) {};
\node (or3) at (4,0) {};
\gatelink{or1.output}{or2.input 1}
\gatelink[0.5]{or2.output}{or3.input 1}
\gatelink[-0.5]{or1.output}{or3.input 2}
\end{tikzpicture}
```

该命令的定义如下:

```
\newcommand*\gatelink[3][0]{
\draw let \p1=(#2), \p2=(#3), \n1={\x1*0.5+\x2*0.5}
in (\x1, \y1) -- ($(\n1, \y1)+(\#1,0)$) |- (\x2, \y2);
}
```

使用以上两个命令的基本门电路用例如下:



```
\usepackage{tikz}
\usetikzlibrary{arrows,shapes,shapes.gates.logic.US,shapes.gates.logic.IEC}
% 导言区
\begin{center}
\begin{tikzpicture}[>=latex]
\node (or) [or gate US, draw, logic gate inputs=nnnn,scale=2] at (4,-5) {};
\gatelabel*{or}{$F$}
\foreach \i/\list/\offset in
{1/{A$,B$,C'$/0.5,2/{A$,C$,D$/-0.5,3/{A'$,B'$,C$/-0.5,4/{A'$,C'$,D$/0.5}
{
\node (and\i) [and gate US, draw, logic gate inputs=nnn, scale=2] at (0,-\i*2) {};
\gatelabel{and\i}{\list}
\gatelink[\offset]{and\i.output}{or.input \i}
}
\end{tikzpicture}
\end{center}
```

MUX MACRO

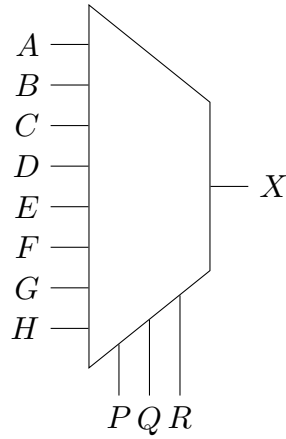
mux 样式的用法如下

```
\node [mux=in-ports-numbers] {};
```

\gateselectlabel 命令为门电路标记选择端口，其用法如下：

```
\gateselectlabel{gate-name}{{comma-seperated-list}}
```

示例代码如下



```

\begin{center}
\begin{tikzpicture}
\node (mux) [mux=8] {};
\gatelabel{mux}{{A$,B$,C$,D$,E$,F$,G$,H$}}
\gatelabel*{mux}{{X$}}
\gateselectlabel{mux}{{P$,Q$,R$}}
\end{tikzpicture}
\end{center}

```

mux 样式的定义如下

```

% see https://stackoverflow.com/questions/61729168/
\pgfkeys{
  /tikz/mux ports/.initial=1
}

\makeatletter
\pgfdeclareshape{muxshape}{
\inheritsavedanchors[from=trapezium]
\inheritanchorborder[from=trapezium]
\inheritbackgroundpath[from=trapezium]
\foreach \anchor in {bottom left corner, top right corner, top left corner, bottom right corner,
  ↪ bottom side, left side, right side, top side, center, text, mid, base, mid west, base west, mid
  ↪ east, base east, west, east, north, south, north west, north east, south west, south east}{
  ↪ \inheritanchor[from=trapezium]{\anchor} }
\savedmacro\numports{%
  \edef\numports{\pgfkeysvalueof{/tikz/mux ports}}%
}
\savedmacro\numselports{%
  \pgfmathtruncatemacro{\selports}{ceil(log2(\pgfkeysvalueof{/tikz/mux ports}))}
  \edef\numselports{\selports}%
}
\anchor{output}{\csname pgf@anchor@muxshape@top side\endcsname}
% input ports and select ports
\pgfutil@g@addto@macro\pgf@sh@s@muxshape{%
  % input ports
  \c@pgf@counta\numports\relax%
  \pgfmathloop%\
  \ifnum\pgfmathcounter>\c@pgf@counta%

```

```

\else%
\expandafter\xdef\csname pgf@anchor@muxshape@input\space\pgfmathcounter\endcsname{%
  \noexpand\pgf@sh@muxshapeanchor{\pgfmathcounter}{\numports}{\csname
↪ pgf@anchor@muxshape@bottom left corner\endcsname}{\csname pgf@anchor@muxshape@bottom right
↪ corner\endcsname}%
}
\repeatpgfmathloop%
% select ports
\c@pgf@counta\numselectports\relax
\pgfmathloop%
\ifnum\pgfmathcounter>\c@pgf@counta%
\else%
\expandafter\xdef\csname pgf@anchor@muxshape@select\space\pgfmathcounter\endcsname{%
  \noexpand\pgf@sh@muxshapeanchor{\pgfmathcounter}{\numselectports}{\csname
↪ pgf@anchor@muxshape@bottom right corner\endcsname}{\csname pgf@anchor@muxshape@top right corner
↪ \endcsname}%
}
\repeatpgfmathloop%
}
}

\def\pgf@sh@muxshapeanchor#1#2#3#4{%
#3
\pgf@xa=\pgf@x \pgf@ya=\pgf@y
#4
\pgf@xb=\pgf@x \pgf@yb=\pgf@y
\pgfmathsetlength{\pgf@x}{\pgf@xa+(\pgf@xb-\pgf@xa)*#1/(#2+1)}%
\pgfmathsetlength{\pgf@y}{\pgf@ya+(\pgf@yb-\pgf@ya)*#1/(#2+1)}%
}
\makeatother

\tikzset{
mux/.code={
  \pgfmathtruncatemacro{\si}{ceil(log2(#1))}%
  \pgfkeys{/tikz/mux ports=#1}
  \pgfkeys{
    /tikz/shape=muxshape,
    /tikz/draw,
    /tikz/trapezium stretches,
    /tikz/shape border rotate = 270,
    /tikz/minimum height=(\si+1)*0.4cm,
    /tikz/minimum width=\si*1.6cm
  }
}
}

```

`\gateselectlabel` 命令的定义如下

```

\newcommand*\gateselectlabel[2]{%
\foreach \label [count=\ni] in #2 {%
  \draw let \p1=(#1.bottom right corner),\p2=(#1.select \ni) in (\p2) -- (\x2,\y1-10) node[below] {

```

```

    → \label};%
};
}

```

线网图

MACRO

paramlines 环境可以用于绘制线网图，其定义和示例如下：

```

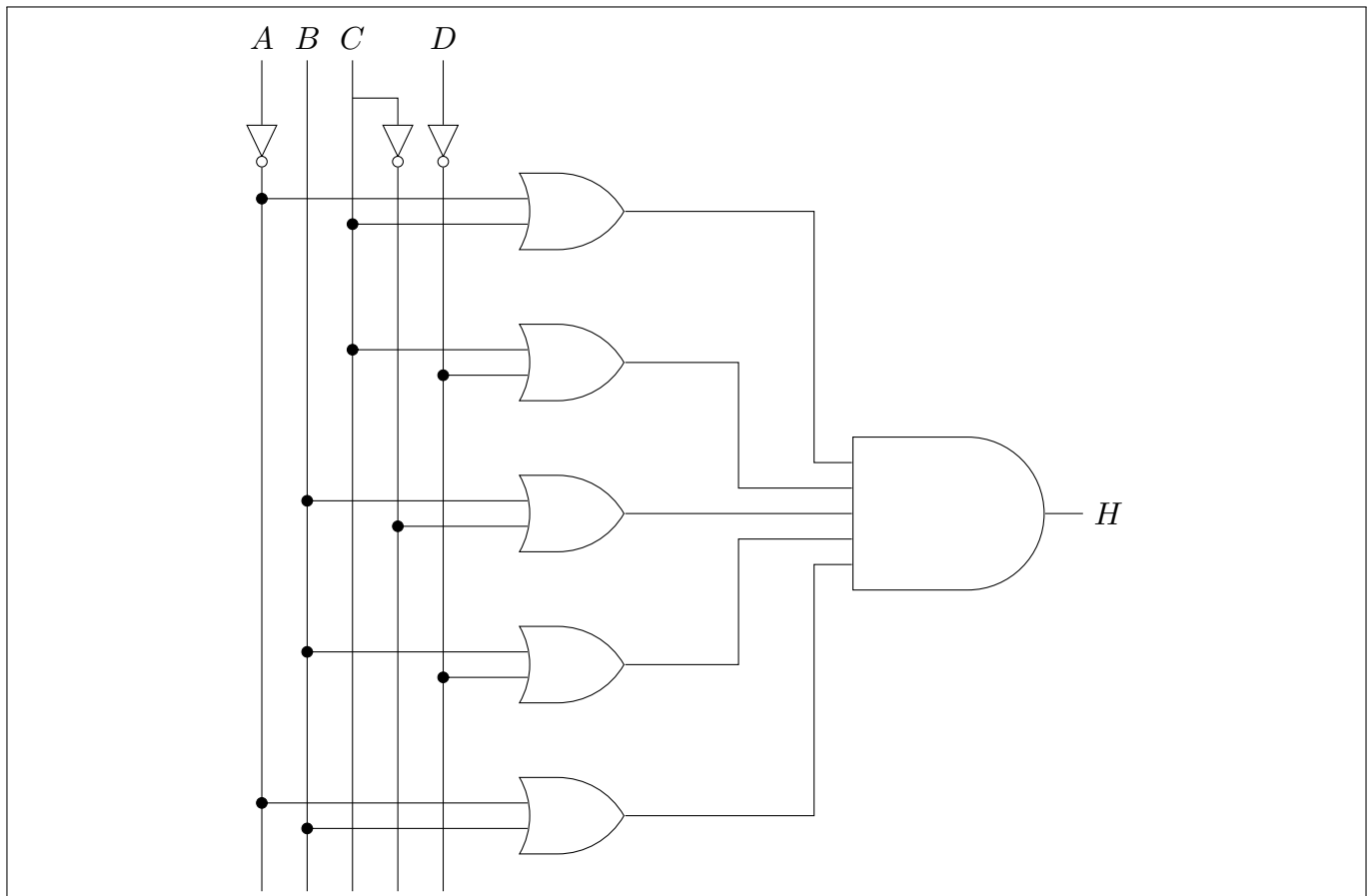
\newcounter{pl@netcount}
\newcounter{pl@linecount}[pl@netcount]
\NewDocumentEnvironment{paramlines}{0{5} 0{1}}{%
  \stepcounter{pl@netcount}%
  \newcommand\plparamdef[1]{%%
    \expandafter\edef\csname pl@##1\endcsname{\arabic{pl@linecount}}%%
    \stepcounter{pl@linecount}%%
  }
  \newcommand\plsetbasex[1]{%%
    \pgfmathsetmacro{\basex}{\csname pl@##1\endcsname*#2}
  }
  \newcommand\plparambase[1]{%%
    \plparamdef{##1}%%
    \plsetbasex{##1}%%
  }
  \newcommand\pldrawbaselabel[1]{%%
    \node at (\basex,0.3) {##1};%%
  }
  \newcommand\plparamlabel[2]{%%
    \plparambase{##1}%%
    \pldrawbaselabel{##2}%%
    \draw (\basex,0) --+ (0,{-(#1)});%%
  }
  \newcommand\plparam[1]{%%
    \plparamlabel{##1}{$##1$}%%
  }
  \newcommand\plparaminvlabel[2]{%%
    \plparambase{##1'}%%
    \pldrawbaselabel{##2}%%
    \node [not gate US,draw,rotate=270] at (\basex,-1) (not##1'){};%%
    \draw (\basex,0) -- (not##1'.input);%%
    \draw (not##1'.output) -- (\basex,{-(#1)});%%
  }
  \newcommand\plparaminv[1]{%%
    \plparaminvlabel{##1}{$##1$}%%
  }
  \newcommand\plparambothlabel[2]{%%

```

```

\plparambase{##1}%%
\pldrawbaselabel{##2}%%
\draw (\basex,0) --+ (0,{-(#1)});%%
\node [not gate US,draw,rotate=270] at ({\basex+#2},-1) (not##1){};%%
\draw (\basex,-0.5)-|(not##1.input);%%
\draw (not##1.output)--({\basex+#2},{-(#1)});%%
\plparamdef{##1'}%%
}
\newcommand\plparamboth[1]{%%
  \plparambothlabel{##1}{$##1$}%%
}
\newcommand\pllink[2]{%%
  \plsetbasex{##1}%%
  \filldraw let \p1=(##2) in (\p1)--(\basex,\y1) circle (2pt);%%
}
}
{}

```

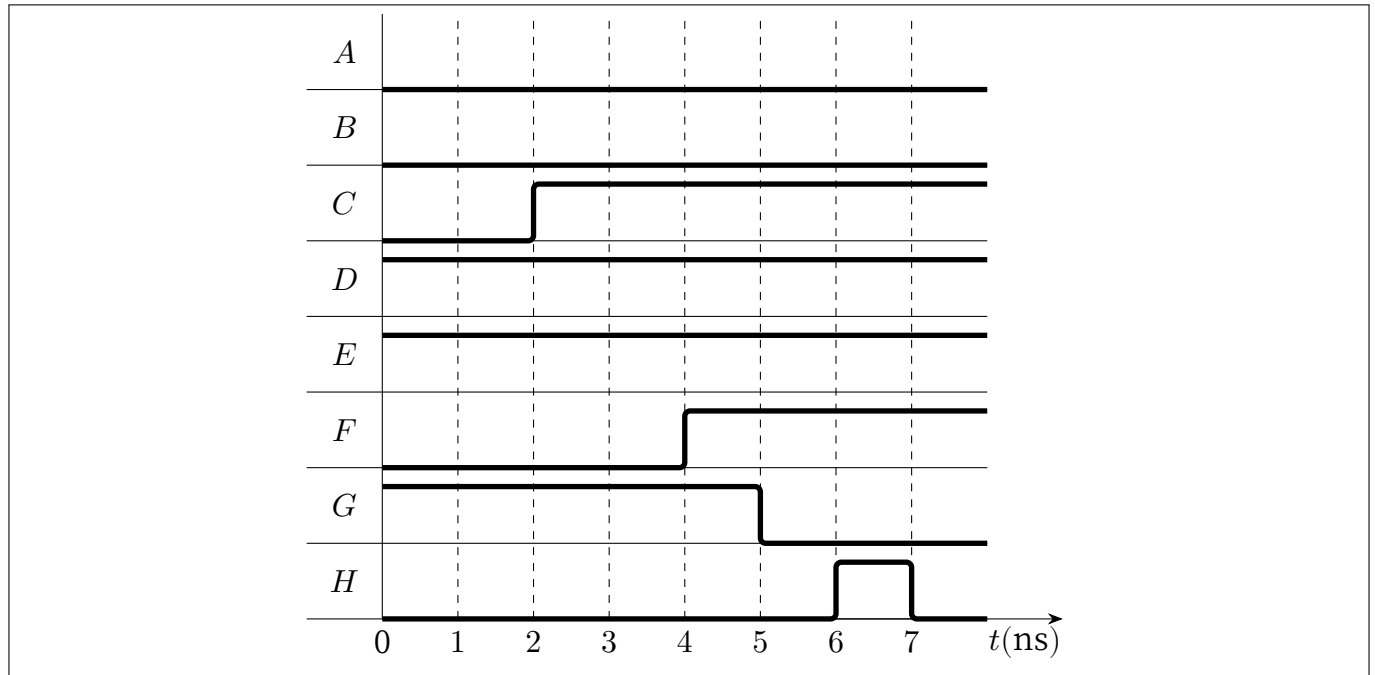


```

\begin{center}
\begin{tikzpicture}
\begin{paramlines}[11][0.6]
\plparaminv{A}\plparam{B}\plparamboth{C}\plparaminv{D}
\foreach \x in {1,...,5} {\node [or gate US,draw,scale=2] at (4,-\x*2) (or\x) {};}
\node (and) [and gate US, draw, anchor=input 3, logic gate inputs=nnnnn, scale=2] at ($(or3.output)
  \red{<-} +(3,0)$) {};
\pllink{A'}{or1.input 1} \pllink{C} {or1.input 2}
\pllink{C} {or2.input 1} \pllink{D'}{or2.input 2}
\pllink{B} {or3.input 1} \pllink{C'}{or3.input 2}
\pllink{B} {or4.input 1} \pllink{D'}{or4.input 2}
\pllink{A'}{or5.input 1} \pllink{B} {or5.input 2}
\gatelink[1] {or1.output}{and.input 1}
\gatelink {or2.output}{and.input 2}
\gatelink[-1]{or3.output}{and.input 3}
\gatelink {or4.output}{and.input 4}
\gatelink[1] {or5.output}{and.input 5}
\gatelabel*{and}{{H}}
\end{paramlines}
\end{tikzpicture}
\end{center}

```


时序图



```
\begin{center}
\begin{tikzpicture}[>={Stealth[length=2mm]}]
\def\lshift{-0.3}    % yshift of label on axes
\def\rlimit{8}       % max x can time line reach
\def\hlevel{0.75}    % yshift of high level signal
% variables
\foreach \label [count=\cnt] in {H,...,A} {%
\node at (-0.5,\cnt-0.5) {$\label$}; \draw (-1,\cnt-1) ---+ (\rlimit+1,0);
};
% axes
\draw (0,0)--(0,8); \draw[>] (0,0)--(\rlimit+1,0);
\node at (0,\lshift) {0}; \node at (\rlimit+0.5,\lshift) {$t(\mathrm{ns})$};
% verticle lines
\foreach \x in {1,...,7} {%
\pgfmathtruncatemacro{\label}{\x}
\node at (\x,\lshift) {$\label$};
\draw[dashed] (\x,0) -- ++ (0,\rlimit);
};
% time line
\begin{scope}[line width=2pt,rounded corners=2pt]
\draw (0,7)--(\rlimit,7); % A
\draw (0,6)--(\rlimit,6); % B
\draw (0,5)---+(2,0)|-(\rlimit,5+\hlevel); % C
\draw (0,4+\hlevel)--(\rlimit,4+\hlevel); % D
\draw (0,3+\hlevel)--(\rlimit,3+\hlevel); % E
\draw (0,2)---+(4,0)|-(\rlimit,2+\hlevel); % F
\draw (0,1+\hlevel)---+(5,0)|-(\rlimit,1); % G
\draw (0,0)---+(6,0)|-++(1,\hlevel)|-(\rlimit,0); % H
\end{scope}
\end{tikzpicture}
\end{center}
```


卡诺图

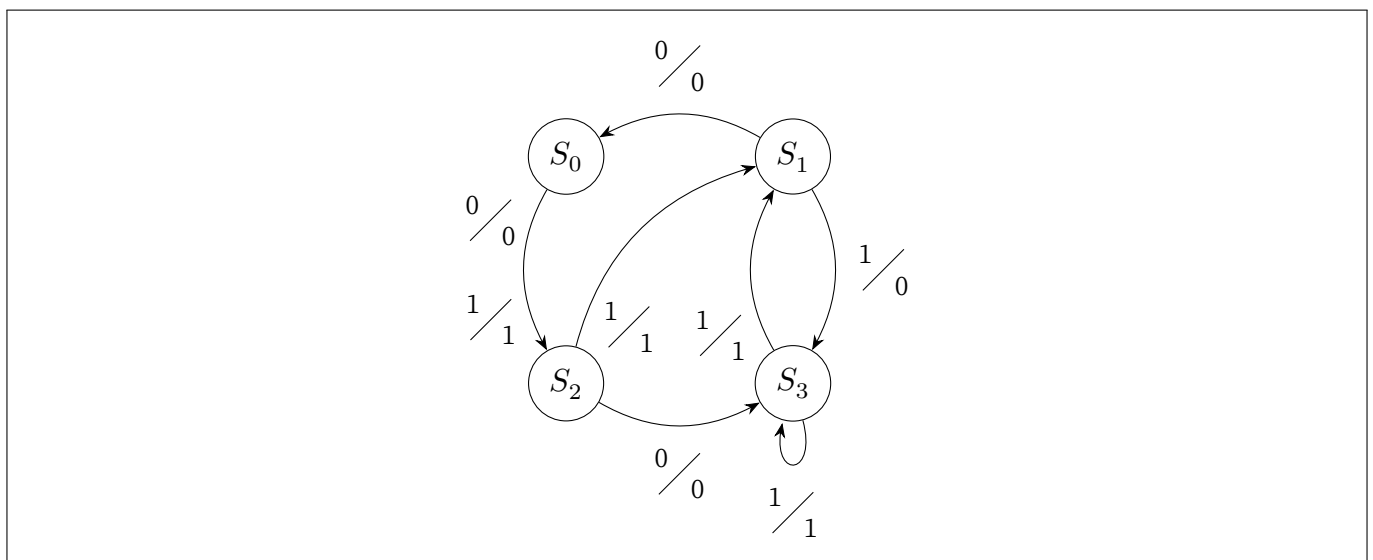
		AB			
		00	01	11	10
CD	00	1	X		
	01	1	1	1	
	11	1			1
	10	1		1	1

```

\usepackage{karnaugh-map} % 导言区
\begin{center}
\begin{karnaugh-map}[4][4][1][AB][CD]
  \minterms{0,4,5,7,8,10,11,12,14}
  \terms{1}{X}
  \implicant{0}{8}
  \implicant{5}{7}
  \implicant{11}{10}
  \implicantedge{12}{8}{14}{10}
\end{karnaugh-map}
\end{center}

```

自动机



```

\usepackage{tikz}
\usetikzlibrary{arrows.meta,calc,graphs,positioning,quotes,shapes} % 导言区
\begin{center}
\begin{tikzpicture}[>={Stealth[length=2mm]}]
\graph[math nodes, nodes={draw,circle}, edge quotes={auto,circle solidus,scale=0.8}, grow right=3cm,
  <- branch down=3cm] {
S_0 <- [bend left, "0 \nodepart{lower} 0"] S_1;
S_2 -> [bend right, "0 \nodepart{lower} 0" swap] S_3;
S_0 -> [bend right, "0 \nodepart{lower} 0" {swap,pos=0.4}, "1 \nodepart{lower} 1" {swap,pos=0.6}] S_2;
S_2 -> [bend left, "1 \nodepart{lower} 1" {swap,pos=0.2}] S_1;
S_3 -> [bend left, "1 \nodepart{lower} 1" pos=0.3] S_1;
S_1 -> [bend left, "1 \nodepart{lower} 0"] S_3;
S_3 -> [loop below, "1 \nodepart{lower} 1"] S_3;
};
\end{tikzpicture}
\end{center}

```

implication 环境可以用于绘制隐含表，其定义和示例如下：

```
% Implication Table environment
% #1 cell size
\makeatletter
\newcounter{impt@netcount}%
\newcounter{impt@linecount}[impt@netcount]%
\newcounter{impt@columncount}[impt@linecount]%
\newenvironment{implication}[1][1.5]{%
  \stepcounter{impt@netcount}%
  \pgfmathsetmacro\cellsize{#1}%
  \pgfmathsetlengthmacro\cellsize\cellsize cm}%
  % \drawcell{label}
  % draw a cell
  \newcommand*\impt@defcxy{%
    \pgfmathsetmacro\cx{\cellsize*\arabic{impt@columncount}}%
    \pgfmathsetmacro\cy{-\cellsize*\arabic{impt@linecount}}%
  }
  \newcommand*\impt@putnode[1]{%
    \node [draw,minimum size=\cellsize\cellsize,align=center] at (\cx,\cy) {##1};%
    \stepcounter{impt@columncount}%
    \ifnum\the\value{impt@columncount}>\the\value{impt@linecount}%
      \stepcounter{impt@linecount}%
    \fi%
  }
  \newcommand*\impt@putx{%
    \draw ({\cx-0.3*\cellsize},{\cy-0.3*\cellsize})---+(0.6*\cellsize,0.6*\cellsize);%
    \draw ({\cx-0.3*\cellsize},{\cy+0.3*\cellsize})---+(0.6*\cellsize,-0.6*\cellsize);
  }
  \newcommand*\drawcell@star[1]{%
    \impt@defcxy%
    \impt@putnode{##1}%
  }
  \newcommand*\drawcell@nostar[1]{%
    \impt@defcxy%
    \impt@putnode{##1}%
    \impt@putx%
  }
  \newcommand*\drawcell{%
    \ifstar{\drawcell@star}{\drawcell@nostar}%
  }
  % \drawlabelv{labels}
  % draw vertical labels
  \newcommand*\drawlabelv[1]{%
    \foreach \label [count=\n from 0] in {##1}{%
      \node [anchor=east] at ({-\cellsize*0.5},{-\cellsize*\n}) {\label};%
    }%
  }
  % \drawlabelh{labels}
```

```

% draw horizontal labels
\newcommand*\drawlabelh[1]{%
  \foreach \label [count=\n from 0] in {##1}{%
    \node [anchor=north] at ({\cellsize*\n},{-\cellsize*(\arabic{impt@@linecount}-0.5)}) {
      \label};%
    }%
  }
}
\makeatother

```

b	$b-g$				
c	$b-g$ $e-f$	$c-f$			
d	\times	\times	\times		
e	$b-d$ $e-f$	$d-g$ $e-f$	$c-e$ $d-g$	\times	
f	$a-f$	$a-f$ $b-g$	$a-e$ $b-g$	\times	$a-e$ $b-d$
g	\times	\times	\times	$a-f$	\times
	a	b	c	d	e

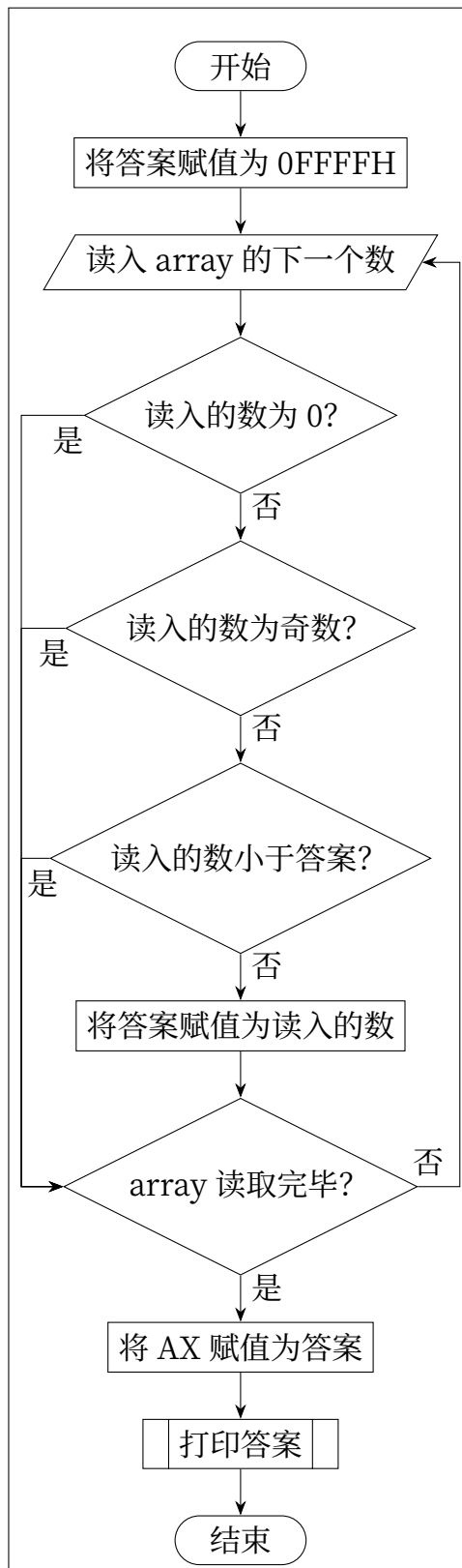
```

\usepackage{tikz} % 导言区
\begin{center}
\begin{tikzpicture}
\begin{implication}[1.35]
%S_1
\drawcell*{$b-g$}
%c
\drawcell{$b-g$\\$c-f$}
\drawcell{$c-f$}
%d
\drawcell{}
\drawcell{}
\drawcell{}
%e
\drawcell{$b-d$\\$e-f$}
\drawcell{$d-g$\\$e-f$}
\drawcell*{$c-e$\\$d-g$}
\drawcell{}
%f
\drawcell*{$a-f$}
\drawcell{$a-f$\\$b-g$}
\drawcell{$a-c$\\$b-g$}
\drawcell{}
\drawcell{$a-e$\\$b-d$}
%g
\drawcell{}
\drawcell{}
\drawcell{}
\drawcell*{$a-f$}
\drawcell{}
\drawcell{}
%labels
\drawlabelv{$b$,$c$,$d$,$e$,$f$,$g$}
\drawlabelh{$a$,$b$,$c$,$d$,$e$,$f$}
\end{implication}
\end{tikzpicture}
\end{center}

```

7.2.2 程序设计

流程图



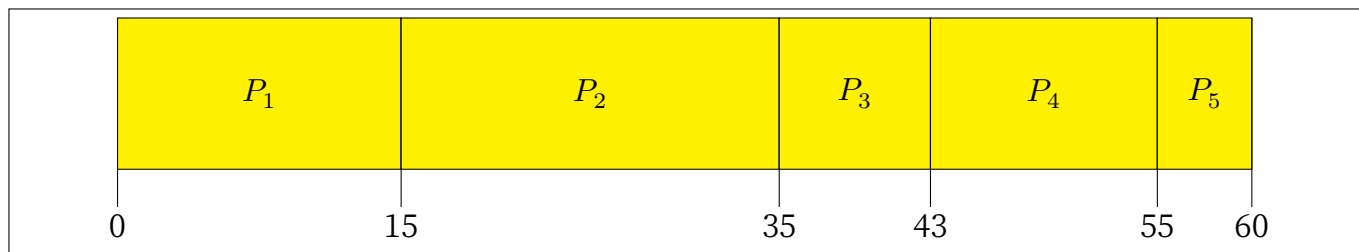
```
\usepackage{tikz}
\usetikzlibrary{arrows.meta,graphs,quotes}
\usepackage{flowchart}
% 导言区
\begin{center}
\begin{tikzpicture}[
>={Stealth[length=2mm]},
io/.style = {draw, trapezium, trapezium left angle=60, trapezium
    right angle=120},
decide/.style={draw, diamond, shape aspect=#1},
decide/.default={2},
term/.style = {draw, terminal},
proc/.style = {draw, process},
preproc/.style = {draw, predproc},
hv path/.style =
{to path={-| (\tikztotarget) \tikztonodes}},
vh path/.style =
{to path={|- (\tikztotarget) \tikztonodes}},
skip loop/.style =
{to path={-- ++(0,#1) \tikztonodes -| (\tikztotarget) }},
vskip loop/.style =
{to path={-- ++(#1,0) \tikztonodes |- (\tikztotarget) }},
every to quotes/.style={auto, near start},
every edge quotes/.style={auto, near start}
]
\graph[grow down sep=1.5em]
{
s [term,as=开始]->
a [proc,as=将答案赋值为0FFFFH]->
b [io,as=读入array的下一个数]->
c [decide,as=读入的数为0? ]->["否"]
d [decide,as=读入的数为奇数? ]->["否"]
e [decide,as=读入的数小于答案? ]->["否"]
f [proc,as=将答案赋值为读入的数]->
g [decide,as=array读取完毕? ]->["是"]
h [proc,as=将AX赋值为答案]->
i [preproc,as=打印答案]->
t [term,as=结束];
c ->["是", vskip loop=-3cm] g;
d ->["是", vskip loop=-3cm] g;
e ->["是", vskip loop=-3cm] g;
g ->["否", vskip loop=3cm] b;
};
\end{tikzpicture}
\end{center}
```

7.2.3 操作系统

调度结果图

dispatchgraph 环境可以用于绘制隐含表，其定义和示例如下：

```
\newcommand*\dgset[1]{%
\pgfkeys{%
/dispatch graph/.cd,#1%
}}
\dgset{
  xscale/.initial = 1,
  height/.initial = 1,
  fill/.initial = yellow!20
}
\newenvironment{dispatchgraph}
{%
  \newcommand*\dgcalc{%
    \edef\dgheight{\pgfkeysvalueof{/dispatch graph/height}}%
    \edef\dgxscale{\pgfkeysvalueof{/dispatch graph/xscale}}%
  }
  \newcommand*\drawdgmark[1][0]{%
    \dgcalc%
    \draw ({##1 * \dgxscale}, -0.5) -- ({##1 * \dgxscale}, \dgheight);%
    \node at ({##1 * \dgxscale}, -0.75) {##1};%
  }
  \newcommand*\drawslice[2]{%
    \dgcalc%
    \filldraw[fill=\pgfkeysvalueof{/dispatch graph/fill}] ({\dgxcur * \dgxscale}, 0) -| ({##1 *
↪ \dgxscale}, \dgheight) -| cycle;%
    \node at ({(\dgxcur + ##1) * \dgxscale * 0.5}, {\dgheight * 0.5}) {##2};%
    \drawdgmark[##1]
    \pgfmathsetmacro\dgxcur{##1}%
  }
  \begin{tikzpicture}%
    \pgfmathsetmacro\dgxcur{0}%
    \drawdgmark[0]%
  }%
  \end{tikzpicture}%
}
```



```

\usepackage{tikz}
\begin{tikzpicture}
  \draw[fill=yellow] (0,0) rectangle (15,2) node[midway] { $P_1$ };
  \draw[fill=yellow] (15,0) rectangle (35,2) node[midway] { $P_2$ };
  \draw[fill=yellow] (35,0) rectangle (43,2) node[midway] { $P_3$ };
  \draw[fill=yellow] (43,0) rectangle (55,2) node[midway] { $P_4$ };
  \draw[fill=yellow] (55,0) rectangle (60,2) node[midway] { $P_5$ };
  \draw (0,0) -- (60,0);
  \draw (0,2) -- (60,2);
  \draw (0,0) -- (0,2);
  \draw (15,0) -- (15,2);
  \draw (35,0) -- (35,2);
  \draw (43,0) -- (43,2);
  \draw (55,0) -- (55,2);
  \draw (60,0) -- (60,2);
  \draw (0,0) node[below] {0};
  \draw (15,0) node[below] {15};
  \draw (35,0) node[below] {35};
  \draw (43,0) node[below] {43};
  \draw (55,0) node[below] {55};
  \draw (60,0) node[below] {60};
\end{tikzpicture}

```