Tanimodori 的 四EX 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
- MEX 表示内容属于 MEX
- 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 图EX 文档基本格式 图EX

```
% !TEX TS-program = xelatex
% !TEX encoding = UTF-8
\XeTeXlinebreaklocale "zh"
\XeTeXlinebreakskip = Opt plus 1pt // 编译引擎选项、编码
\documentclass[12pt]{article}
                                   % 全局选项、文档属性
\usepackage{xeCJK,fontspec}
                                    % 宏包及其选项
\usepackage{graphicx}
\usepackage{amsmath,amssymb,amsthm}
\title{\LaTeX\ Test}
                                    1. 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 TEX

- **№**TEX
- 一次 IATEX 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 $X_{\overline{1}}T_{\overline{1}}X$)
- 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

LATEX without 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。 LAT_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 LAT_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	
工作见及 	代码	示例	代码	示例
.16667em	a∖,b	ab	\$ab\$	a b
或 3mu	a\thinspace b	ab	\$a\thinspace b\$	a b
-3mu			\$a\!b\$	db
-Siliu			\$a\mkern-\thinmuskip b\$	db
4.0mu			\$a\>b\$	a b
plus 2.0mu minus 4.0mu			\$a\:b\$	a b
iiiiius 4.0iiiu			\$a\mkern\medmuskip b\$	a b
5.0mu			\$a\;b\$	a b
plus 5.0mu			\$a\mkern\thickmuskip b\$	a b
.5em	a\enspace b	a b	\$a\enspace b\$	a b
1em	a b	a b	\$a b\$	a b
2em	a\qquad b	a b	\$a\qquad b\$	a b
	a\hskip 1em b	a b	\$a\hskip 1em b\$	a b
<len></len>	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></stuff>	axyzb	axyzb	\$axyzb\$	axyzb
\Stull>	a\hphantom{xyz}b	a b	<pre>\$a\hphantom{xyz}b\$</pre>	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~b\$	a b
rubber	a\hfill b	a b	\$a\hfill b\$	a b
Tunnei	a\hspace{\fill} b	a b	<pre>\$a\hspace{\fill} b\$</pre>	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 \LaTeX 不会在此处换行。\hfill、\vfill(\TeX)、\hspace{\fill}、\vspace{\fill}、\phantom会是有一个,则按照比例分配,例如:

左对齐	分文	左对齐 \hspace*{\fill}\\ \hspace*{\fill} 右对齐\\
居中		\hspace*{\fill} 居中 \hspace*{\fill}

1.3 字体 [ATEX]

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
	{\normalfont}	sample text 示例文本
	{\em}	sample text 示例文本
	{\rmfamily}	sample text 示例文本
	{\sffamily}	sample text 示例文本
	{\ttfamily}	sample text 示例文本
	{\upshape}	sample text 示例文本
	{\itshape}	sample text 示例文本
\texts1{}	{\slshape}	sample text 示例文本
	{\scshape}	不支持
	{\bfseries}	sample text 示例文本
	{\mdseries}	sample text 示例文本
	{\lfseries}	不支持

1.3.3 指定字体大小

命令	效果	
\tiny	sample text 示例文本	
\scriptsize	sample text 示例文本	
│ footnotesize sample text 示例文本		
\small	sample text 示例文本	
\normalsize	sample text 示例文本	
\large	sample text 示例文本	
\Large	sample text 示例文本	
\LARGE	sample text 示例文本	
\huge	sample text 示例文本	
\Huge	sample text 示例文本	

1.3.4 指定字体颜色 PKG



red text		
blue text		

{\color{red} red text}\\ \textcolor{blue}{blue text}

版式与结构 2

2.1 参考文献

```
\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}
```

数学模式TE 3 IAT_EX

下花括号

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

分类

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

矩阵

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

\[A=\begin{bmatrix}

182\\
384
\end{bmatrix}\]

等式

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

连等式

$$y = (x+1)(x-1)$$
$$= x^2 - 1$$

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

4 表格 [ATEX

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
      & 0000 \\\hline
      & 0001 \\\hline
1
      & 0010 \\\hline
3
      & 0100 \\\hline
4
      & 1000 \\\hline
5
      & 1001 \\\hline
      & 1010 \\\hline
7
      & 1100 \\\hline
              \\\hline
8
      & 1101
      & 1110
              \\\hline
\end{tabular}
\end{center}
```

多行与多列 PKG

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

```
\usepackage{multirow} %导言区
\usepackage{multirow} %导言区
\usepackage{multirow} {c|cc|c}\usepackage{multirow} {c|cc|c}\usepackage{multirow} {2}*{现态} {8\usepackage} {ultirow} {2}*{现态} {2}*{ultirow} {2}
```

5 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
\usepackage{matlab-prettifier} % \usepackage{matlab-editor}
\usepackage{matlab-prettifier} % \under \unde
```

x86 汇编

```
data segment
s db 'Hello World!$'
data ends
                           \usepackage{listings} % 导言区
                           \lstdefinestyle{customasm}{
code segment
                           language=[x86masm]Assembler,
assume cs:code, ds:data
                           basicstyle=\linespread{1.0}\small\ttfamily,
start: mov ax, data
                           commentstyle=\color{Brown},
mov ds, ax
                           keywordstyle=\color{blue}\bfseries,
lea dx, s
                           stringstyle=\color{Green}\bfseries,
mov ah, 09H
                           keepspaces=true,
int 21H
                           columns=fixed,
                           basewidth=0.5em,
mov ax, 4C00H
int 21H
                           \lstinputlisting[style=customasm]{x86asm_example.asm}
code ends
end start
```

 \mathbf{C}

```
int main(int argv, char* argc[]) {
    for (int j = 0; j < 200; ++j) {
         int t, p, found;
         fscanf(input, "(%d,%d) ", &t, &
                                                \usepackage{listings} % 导言区
\hookrightarrow p);
                                                \lstdefinestyle{customc}{
         found = mem_find(p);
                                                language=C, %
         if (found != -1) {
                                                keepspaces=true, %
             ++hit:
                                                columns=fixed, %
             fprintf(output, "%c", mem[
                                                basewidth=0.5em, %
                                                basicstyle=\ttfamily,%
→ found].data[t]);
                                                breaklines=true, %
         } else {
                                                commentstyle=\color{Green},%
              found = mem_find_empty();
                                                keywordstyle=\color{blue}\bfseries, %
             mem_load(&mem[found], disk,
                                                stringstyle=\color{Brown}\bfseries,%
                                                numberstyle=\ttfamily\color{Gray},%
                                                showstringspaces=false%
             fprintf(output, "%c", mem[
\hookrightarrow found].data[t]);
                                                \lstinputlisting[style=customc]{c_example.c}
    return 0;
}
```

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;
       for (i = 0; i < N; i++)
13 -
14 -
            pthread_create(&tid[i], NULL,
       \hookrightarrow thread, &i);
       int i, args[N];
12 +
       for (i = 0; i < N; i++) {
13 +
14 +
            args[i] = i;
            pthread_create(&tid[i], NULL,
15 +
       \hookrightarrow 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}]{00},
    morecomment=[f][\color{diffincl}]{+\ },
    morecomment=[f][\color{diffrem}]{-\ },
    numbers=left,
    stepnumber=1
}
% LISTINGS linenumber hack
\makeatletter
\let\orig@lstnumber=\thelstnumber
\newcommand\lstsetnumber[1]{\gdef\thelstnumber
    → {#1}}
\newcommand\lstresetnumber{\global\let

→ \thelstnumber=\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)

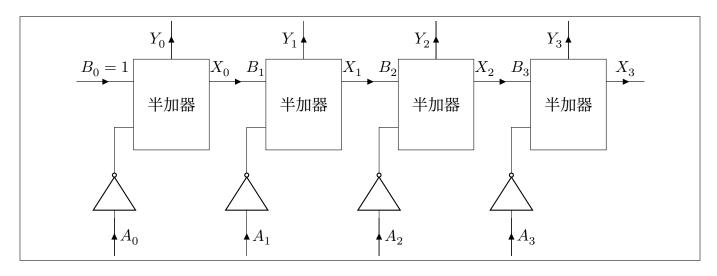
→ ;$\lstresetnumber\setcounter{lstnumber}

    \hookrightarrow }{11}$
    int i, args[N];
    for (i = 0; i < N; i++) {
        args[i] = i;
        pthread_create(&tid[i], NULL, thread,
    \hookrightarrow args + i);
\end{lstlisting}
```

6 Tikz PKG

6.1 命令示例

\foreach

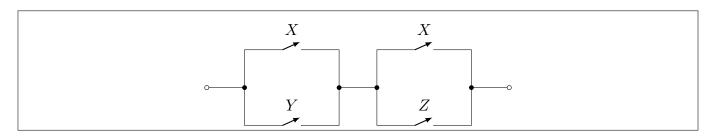


```
\begin{center}
 \begin{tikzpicture}[>=latex]
\foreach \x in \{1, 2, 3, 4\} {
                                           \protect\operatorname{\protect} \operatorname{\protect} \operatorname{\prote
                                           \draw
                                           (\xminus*3.5,0) node[draw,minimum width=2cm,minimum height=2.4cm] (ha\x) {半加器}
                                           (\frac{(\frac{x\cdot y}{1})}{0.5!}(\frac{x\cdot y}{1}) coordinate (\frac{x\cdot y}{1})
                                           (\frac{(ha\x.west)!0.5!(ha\x.north west)}) coordinate (ha\x-b)
                                           ($(ha\x.north)$) 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^<=${A_\times minus}$] ++ (0,-1)
                                           (ha\x-y) to[short,i^>=${Y_xminus}$] ++ (0,1);
}
 \foreach \x in \{1, 2, 3\} {
                                           \protect\operatorname{\protect} \operatorname{\protect} \operatorname{\prote
                                           \pgfmathtruncatemacro{\xplus}{\x+1}
                                           \draw
                                           (ha\x-x) to[short,i^>=${X_\times \alpha B_x}$] (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}
```

6.2 内容示例

6.2.1 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,->,1=$X$] ++ (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, ->, 1=\$Y\$] ++ (0.5,0) --++ (1,0)
-- (left_out)
(right_start) --++ (0,1)
--++ (1,0) to[nos,->,1=$X$] ++ (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, ->, 1=$Z$] ++ (0.5,0) --++ (1,0)
-- (right_end);
\end{circuitikz}
\end{center}
```

门电路 MACRO

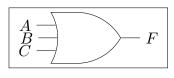
首先导入 xparse 宏包

```
\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}
```

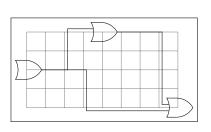
该命令的定义如下:

```
\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。



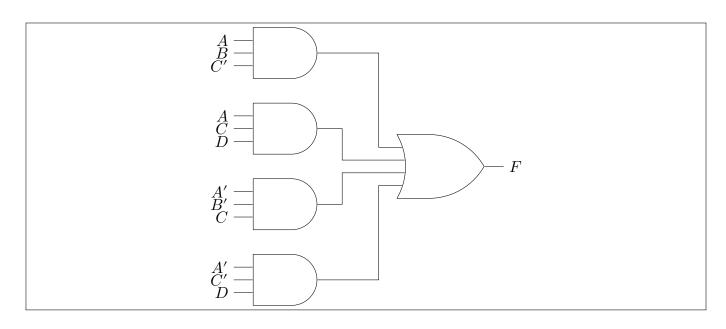
```
\begin{tikzpicture}[every node/.style={or gate US, draw}]
\draw[help lines, step=0.5cm] (0,0) grid (4,2);
\node (or1) at (0,1) {};
\node (or2) at (2,2) {};
\node (or3) at (4,0) {};
\gatelink{or1.output}{or2.input 1}
\gatelink[0.5]{or2.output}{or3.input 1}
\gatelink[-0.5]{or1.output}{or3.input 2}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

该命令的定义如下:

```
\newcommand*\gatelink[3][0]{
\draw let \p1=(\p1=(\p12),\p2=(\p13),\n1=\{\x1*0.5+\x2*0.5\}

in (\x1,\y1) -- (\s(\n1,\y1)+(\p1,0)\s) |- (\x2,\y2);
}
```

使用以上两个命令的基本门电路用例如下:



```
\usetikzlibrary{arrows, shapes, shapes.gates.logic.US, shapes.gates.logic.IEC}

% 导言区
\usetikzlibrary{arrows, shapes, shapes.gates.logic.US, shapes.gates.logic.IEC}

% 导言区
\usetimes (center)
\usetimes (center)
\usetimes (center)
\usetimes (center) (center)
\usetimes (center) (center) (center)
\usetimes (center) (center) (center)
\usetimes (center) (center) (center) (center)
\usetimes (center) (center) (center) (center) (center) (center)
\usetimes (center) (
```

MUX MACRO

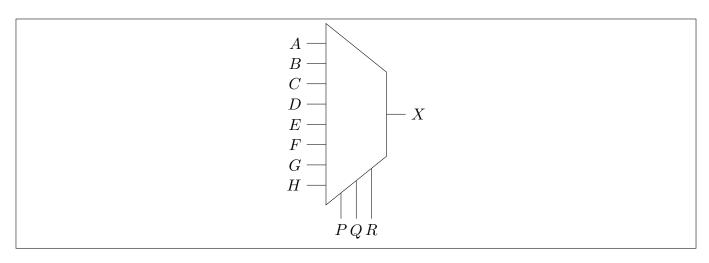
mux 样式的用法如下

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

\gateselectlabel 命令为门电路标记选择端口,其用法如下:

 $\verb|\gateselectlabel| \{ \textit{gate-name} \} \{ \{ \textit{comma-seperated-list} \} \}$

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
    \hookrightarrow 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
          }
           \repeatpgfmathloop%
          % select ports
          \c@pgf@counta\numselports\relax
           \pgfmathloop%\
           \ifnum\pgfmathcounter>\c@pgf@counta%
           \else%
           \expandafter\xdef\csname pgf@anchor@muxshape@select\space\pgfmathcounter\endcsname{%
                     \noexpand\pgf@sh@@muxshapeanchor{\pgfmathcounter}{\numselports}{\csname
          → pgf@anchor@muxshape@bottom right corner\endcsname}{\csname pgf@anchor@muxshape@top right corner
           \repeatpgfmathloop%
\def\pgf@sh@@muxshapeanchor#1#2#3#4{%
           \pgf@xa=\pgf@x \pgf@ya=\pgf@y
           \pgf@xb=\pgf@x \pgf@yb=\pgf@y
           \proonup \
           \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

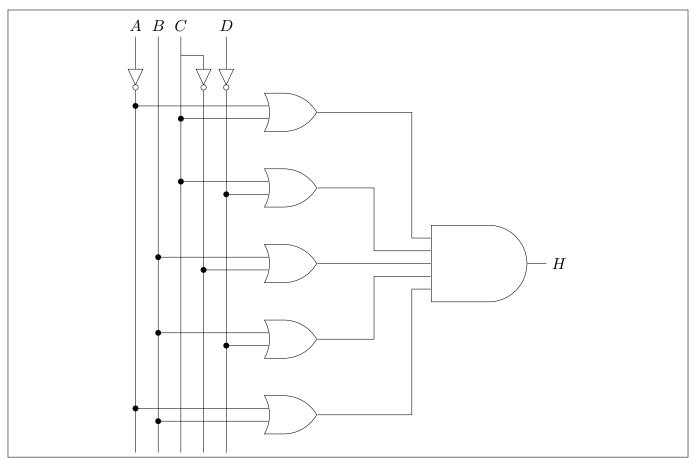
```
\newcommand*\gateselectlabel[2]{%
\foreach \label [count=\ni] in #2 {%
\draw let \p1=(#1.bottom right corner),\p2=(#1.select \ni) in (\p2) -- (\x2,\y1-10) node[below] {
```

线网图 MACRO

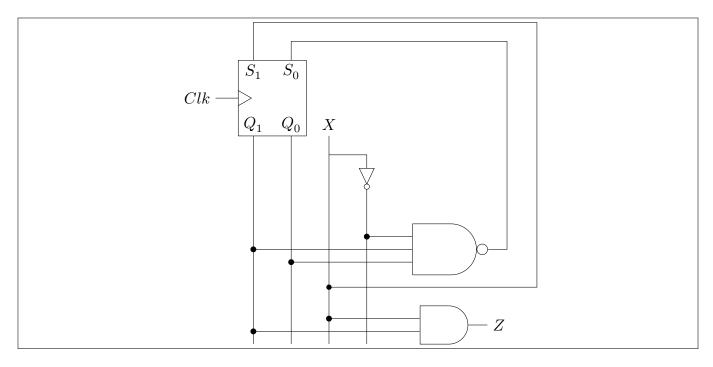
paramlines 环境可以用于绘制线网图, 其定义和示例如下:

```
\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,{-(\pm1)});\".\"
    \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'){}; \lambda \lambda
        \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,\{-(\sharp 1)\});%%
         \node [not gate US,draw,rotate=270] at ({\bsum {1},-1} (not##1){}; %%
         \draw (\basex,-0.5)-|(not##1.input); %%
         \label{lower_section} $$ \operatorname{not}_{1.output} --({\textstyle \text{\basex+}$2}, {-($1)$}); \hdots \\
         \plparamdef{##1'}%%
    }
    \newcommand\plparamboth[1]{%%
         \plparambothlabel{##1}{$##1$}%%
    \newcommand\pllink[2]{\"."
         \plsetbasex{##1}%%
         \filldraw let \p1=(\#2) in (\p1)--(\basex,\y1) circle (2pt); \%
    }
}
{}
```

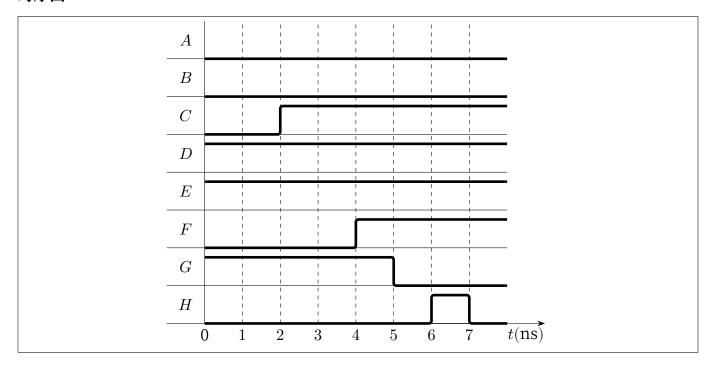


```
\begin{center}
\begin{tikzpicture}
\begin{paramlines}[11][0.6]
\plparaminv{A}\plparam{B}\plparamboth{C}\plparaminv{D}
\foreach \x in \{1,...,5\} {\node [or gate US,draw,scale=2] at (4,-\x^2) (or\x) \{\};}
\node (and) [and gate US, draw, anchor=input 3, logic gate inputs=nnnn, scale=2] at ($(or3.output)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



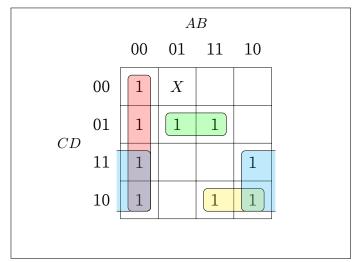
```
\begin{center}
 \usepackage{tikz}
 \usetikzlibrary{arrows.meta,shapes} % 导言区
\begin{tikzpicture}
 \begin{paramlines}[5.5][1]
 \left[ \begin{array}{c} \left[ 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.015, 0.0
\left[0\right] \left[0\right
 \plparamboth{X}
 \node [draw, shape=rectangle, minimum height=2cm, minimum width=1.8cm] at (0.5,1){};
\node at (0,1.7) {$S_1$};
\node at (1,1.7) {$$_0$};
 \node (clk) [left] at (-1,1) {$Clk$};
\draw (clk) -- (-0.4,1) --+ (0,0.2) --+ (0.35,0) --+(0,-0.2);
% gates
\def\andbasex{5}
\def\orbasex{5}
\node [nand gate US,logic gate inputs=nnn,draw,scale=2] at (\orbasex,-3) (nand1) {};
\node [and gate US,draw,scale=2] at (\andbasex,-5) (and1) {};
 \pllink{X}{and1.input 1}
\pllink{Q1}{and1.input 2}
\pllink{X ' }{nand1.input 1}
 \pllink{Q1}{nand1.input 2}
\pllink{Q0}{nand1.input 3}
 \gatelabel*{and1}{$Z$}
 \plsetbasex{X}
\draw (nand1.output) --++(0.5,0) |- (1,2.5) -- (1,2);
 \draw (\basex,-4) --++(5.5,0) |-(0,3) --(0,2);
\fill (\basex,-4) circle (2pt);
 \end{paramlines}
 \end{tikzpicture}
 \end{center}
```

时序图



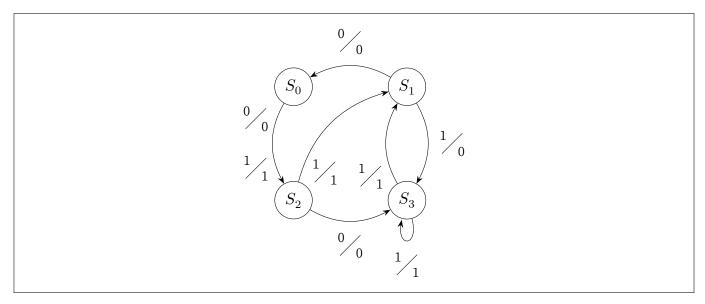
```
\begin{center}
\begin{tikzpicture}[>={Stealth[length=2mm]}]
\def\lshift{-0.3}
                    % yshift of label on axies
\def\rlimit{8}
                     % max x can time line reach
\def\hlevel{0.75}
                    % yshift of high level signal
% variables
\foreach \label [count=\cnt] in {H,...,A} {%
\node at (-0.5,\cnt-0.5) {\label\}; \draw (-1,\cnt-1) --++ (\rlimit+1,0);
};
% axies
draw (0,0)--(0,8); draw[->] (0,0)--(\rlimit+1,0);
\node at (0, \ \{0\}; \ \text{at } (\ \text{mathrm}\{ns\});
% verticle lines
\foreach \x in \{1, \ldots, 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
(0,2)--++(4,0)|-(\rlimit,2+\hlevel); ", F
(0,1+\hlevel)--++(5,0)|-(\rlimit,1); \% G
\text{draw } (0,0) --++(6,0)|-++(1,\hlevel)|-(\rlimit,0); % H
\end{scope}
\end{tikzpicture}
\end{center}
```

卡诺图



```
\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}
  \implicant411}{10}
  \implicantedge{12}{8}{14}{10}
  \end{karnaugh-map}
\end{center}
```

自动机



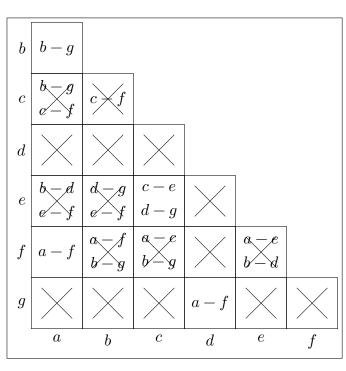
```
\usepackage{tikz}
\usetikzlibrary{arrows.meta,calc,graphs,positioning,quotes,shapes} % 导言区
\begin{center}
\begin{tikzpicture}[>={Stealth[length=2mm]}]
\graph[math nodes, nodes={draw,circle}, edge quotes={auto,circle solidus,scale=0.8}, grow right=3cm,
                → branch down=3cm] {
S_0 \leftarrow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



implicantion 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\cellsizelength{#1 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=\cellsizelength,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 {\pmu1}{\%\'\'.
            \node [anchor=east] at ({-\cellsize*0.5},{-\cellsize*\n}) {\label};%
        }%
    }
    % \drawlabelh{labels}
```

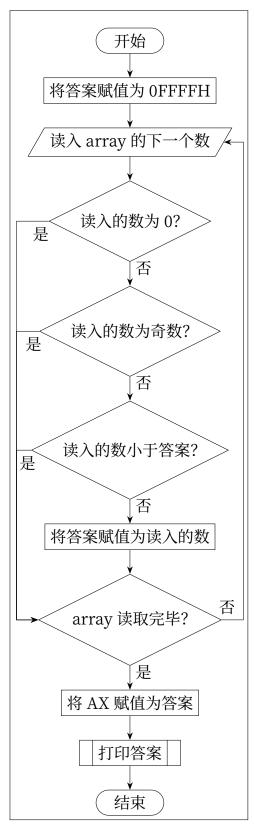
```
% draw horizontal labels
  \newcommand*\drawlabelh[1]{%
      \foreach \label [count=\n from 0] in {##1}{%
            \node [anchor=north] at ({\cellsize*\n},{-\cellsize*(\arabic{impt@@linecount}-0.5)}) {
      \label\};%
      }%
    }
}{
}
makeatother
```



```
\usepackage{tikz} % 导言区
\begin{center}
\begin{tikzpicture}
\begin{implication}[1.35]
%S_1
\drawcell*{$b-g$}
%с
\displaystyle \frac{\$b-g\$}{\c-f\$}
\drawcell{$c-f$}
%d
\drawcell{}
\drawcell{}
\drawcell{}
%е
\displaystyle \frac{\$b-d}{\$e-f}}
\displaystyle \frac{1}{\$d-g\$}\
\drawcell*{$c-e$\\\sd-g$}
\drawcell{}
%f
\drawcell*{$a-f$}
\displaystyle \frac{\$a-f}{\$b-g}}
\drawcell{$a-c$\\\b-g$}
\drawcell{}
\displaystyle \frac{\$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}
```

6.2.2 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={-- ++(\pmu1,0) \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}
```

6.2.3 操作系统

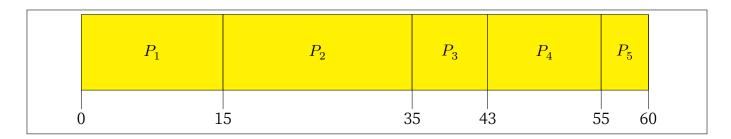
调度结果图

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 ({\pmu 1 * \dgxscale}, -0.75) {\pmu 1};\psi.
    \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]<mark>%</mark>
}{%
    \end{tikzpicture}%
}
```



```
\usepackage{tikz}
\dgset{
    height = 2,
    xscale = 0.25,
    fill = yellow
} % 导言区
\begin{center}
\begin{dispatchgraph}
    \drawslice{15}{$P_1$}
    \drawslice{35}{$P_2$}
    \drawslice{43}{$P_3$}
    \drawslice{55}{$P_4$}
    \drawslice{60}{$P_5$}
\end{dispatchgraph}
\end{center}
```