The askmaps package

Jesse op den Brouw*

2020/06/09, version 0.2

1 Abstract

This package generates configurable American style Karnaugh maps for 1, 2, 3, 4 and 5 variables as can be found in numerous books on digital design. Note that there are many ways to draw correct Karnaugh maps[†] and this packages creates only one version.

There are more packages and examples that produce Karnaugh maps. A highly recommendable package is written by Andreas W. Wieland[‡]. This package creates maps that are frequently found in European textbooks[§] but not in American textbooks.

2 Overview

After `\usepackage[`*`option`*`]{askmaps}`, siz new commands are available to draw Karnaugh maps:

`\askmapi`	draws a one-variable Karnaugh map
`\askmapii`	draws a two-variable Karnaugh map
`\askmapiii`	draws a three-variable Karnaugh map
`\askmapiiialt`	draws an alternate style three-variable Karnaugh map
`\askmapiv`	draws a four-variable Karnaugh map
`\askmapv`	draws a five-variable Karnaugh map

These commands have five parameters which have the same meaning for each commands. This will be explained in Section 3. As can be seen later on, many of the parameters can include typesetting commands such as coloring and math notation.

The new dimension `\askmapunitlength` is available and specifies the length and width of the squares. It defaults to 1 cm (about 0.4 in) which is good in combination with a 12 pt font size.

The global definitions `\askmapsversion` and `\askmapsdate` will render to the current version and date respectively.

There are three commands that control the size (and shape) of the rendered text. The `\askmapindexsize` command sets the size and shape of the index number, the default

*The Hague University of Applied Sciences, J.E.J.opdenBrouw@hhs.nl

†E.g., a four-variable Karnaugh map can be drawn in 384 different and correct ways

‡See http://www.ctan.org/tex-archive/macros/latex/contrib/karnaugh

§E.g. Digitale Techniek, A.P. Thijssen

is \scriptsize\slshape. The \askmapcontentsize command sets the size of the contents of the cells. It defaults to \normalsize. The \askmapbitcombinationsize sets the the size of the bit combinations on the edges of the Karnaugh map. It defaults to \footnotesize. These sizes can be changed using the appropriate \renewcommand command.

The command \askmapvarsep contains the length of the variable separator line in de Karnaugh maps. The default is 0.7. Note that it is merely a number, not a length. It can be altered using the appropriate \renewcommand command. See Section 5 for more details.

The command \askmap provides drop-in replacement for the \karnaughmap command available in the kvmacros package and internally calls one of the five commands (the \askmapiiialt command is not supported), but there are some drawbacks on using this command.

3 Outline

All six commands take the form \askmap<spec>{#1}{#2}{#3}{#4}{#5} where <spec> is one of i, ii, iii, iiialt, iv or v and the parameters #1 to #5 have the following meaning:

- #1 is the function output variable.
- #2 is a list of function input variables.
- #3 contains a list op options, see table below.
- #4 is a list of function values.
- #5 can be used to display user defined picture commands.

If a parameter contains fewer elements than needed, you will get empty spaces in the Karnaugh map. If a parameter contains more elements than needed, all elements in excess are not printed. In both situations, you will not be notified.

The third parameter contains a list of options as explained below.

- i index numbers are printed in the lower left corner of each square.
- I no index numbers are printed (default).
- f function output variable is printed at the upper right corner of the Karnaugh map together with a small line extending from the square to the function name (default, see Package Option Handling).
- F no function output variable is printed
- b bit combinations of the function input variables are printed on top of the columns and at the left of the the rows (default).
- B bit combinations are not printed.
- c shortcut for ifb.
- C shortcut for IFB.

Options are evaluated from left to right, so a combination of iI will yield in no index numbers being printed. Please note that macros are not expanded to a list of tokens but to one token.

4 Package Option Handling

As of version 0.2, the option `disablef` disables the default rendering of the function output variable at the upper right corner of the Karnaugh map together with a small line extending from the square to the function name (option `f` in the third parameter). This is more common found in American textbooks.

5 Use

Let's say we have the following truth table for a function S with two variables a and b and the function values 1, 0, 1 and 1 respectively.

Table 1: *A truth table.*

a	b	S
0	0	1
0	1	0
1	0	1
1	1	1

The command

```
\askmapii{S}{ab}{i}{1011}{}
```

will produce the two-variable Karnaugh map as presented in Figure 1.

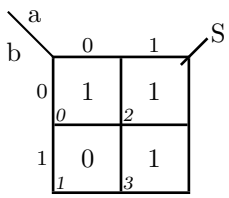


Figure 1: *Karnaugh map for two variables.*

As you can see, the function output variable is printed top right of the Karnaugh map, with a small line from the variable to the top right square. The most significant function input variable is printed on the top left just above the small line extending from the top left square (the variable separator line). The least significant function input variable is printed just below that line. The bit combinations for the most significant function input variables are printed on top of the columns, the bit combinations of the least significant function input variable are printed left of the rows. You can see how the function values are placed in the squares by observing the small index numbers in the lower left corners of the squares.

The command

```
\askmapiii{S}{abc}{}{10110110}{}
```

will produce a three-variable Karnaugh map as presented in Figure 2. In this case, the index numbers are not printed.

c \ ab	00	01	11	10
0	1	1	1	0
1	0	1	0	1

S

Figure 2: *Karnaugh map for three variables.*

Sometimes you need a three-variable Karnaugh map that is printed in the vertical direction. It will leave you room to print accompanying text to the left or right of the map. The command

```
\askmapiiialt{S}{abc}{}{10110110}{}
```

will produce the vertical oriented version of the three-variable Karnaugh map as can been seen in Figure 3.

bc \ a	0	1
00	1	0
01	0	1
11	1	0
10	1	1

S

Figure 3: *Vertical version of the Karnaugh map for three variables.*

Let's draw a Karnaugh map for a logic function with four inputs. The output will become logic 1 if three or four inputs are logic 1, otherwise the output is logic 0. (Of course you can set up a truth table first; this is left as an exercise to the reader). The command

```
\askmapiv{S}{abcd}{}{0000000100010111}{}
```

will produce a four-variable Karnaugh map for the given logic function and is presented in Figure 4.

cd \ ab	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	0

S

Figure 4: *Karnaugh map for four variables.*

Using the command

```
\askmapv{S}{abcde}{}{00101101010010111011110011101011}{}
```

a five-variable Karnaugh map is produced as shown in Figure 5.

de \ abc	000	001	011	010	110	111	101	100
00	0	1	1	0	1	1	1	1
01	0	1	0	1	1	0	1	0
11	0	1	1	0	0	1	0	1
10	1	0	1	0	1	1	0	1

Figure 5: *Karnaugh map for five variables.*

Of course you can do a lot more with the arguments of the commands. You can use inline math to produce sophisticated variable names, you can use coloring for the function values. However, please note that the second and fourth parameter act as a list of tokens, so you have to use braces to separate the tokens in the lists if they consists of multiple characters. A list of tokens such as

```
{abcd}
```

will yield a, b, c and d as separate tokens, but

```
{$m_{0}$$m_{1}$$m_{0}$$m_{1}$}
```

will not work. You have to write

```
{{$m_{0}$}{$m_{1}$}{$m_{0}$}{$m_{1}$}}
```

The fifth parameter can be used to supply user created picture commands such as dashed rectangles or ovals to emphasize the simplifications in the function. Note that the origin of the picture (0,0) is at the lower left corner of the lower left square. Sections 6 and 7 show some examples.

The command `\askmap` provides a drop-in replacement for the command `\karnaughmap` from the Karnaugh package. The command

```
\askmap{4}{S}{abcd}{0100101010100011}{}
```

produces the Karnaugh map in Figure 6.

cd \ ab	00	01	11	10
00	0 (0)	1 (4)	0 (12)	1 (8)
01	1 (1)	0 (5)	0 (13)	0 (9)
11	0 (3)	0 (7)	1 (15)	0 (11)
10	0 (2)	1 (6)	1 (14)	1 (10)

S

Figure 6: *Karnaugh map using drop-in replacement.*

Note that when using this command the output function variable is always printed.

Only Karnaugh maps of one to five variables are supported, any other number will trigger a warning. Please note that `\kvindex` and `\kvnoindex` are supported as can be seen in Figure 7, but `\kvindexsize` and `\kvcontentsize` commands are not supported.

```
\kvindex
\askmap{4}{f(a,b,c,d):}{abcd}{1110011001100110}{}
```

cd \ ab	00	01	11	10
00	1 (0)	0 (4)	0 (12)	0 (8)
01	1 (1)	1 (5)	1 (13)	1 (9)
11	0 (3)	0 (7)	0 (15)	0 (11)
10	1 (2)	1 (6)	1 (14)	1 (10)

f(a,b,c,d):

Figure 7: *Karnaugh map using the* `kvmacros` *package.*

The packages `kvmacros` and `askmaps` can be used in the same document as shown in Figure 8.

```
\kvnoindex
\karnaughmap{4}{f(a,b,c,d):}{abcd}{1110011001100110}{}
```

f(a,b,c,d):

b

d

c

a

1	1	1	0
1	0	0	1
1	0	0	1
0	1	1	0

Figure 8: *Karnaugh map using the* `kvmacros` *package.*

The lenght of the line separating the input variables can be set to a different value as shown in Figure 9.

```
\renewcommand{\askmapvarsep}{0.6}
\askmapiii{S}{abc}{}{00111010}{}
\renewcommand{\askmapvarsep}{0.9}
\askmapiii{F}{xyz}{}{11111010}{}
```

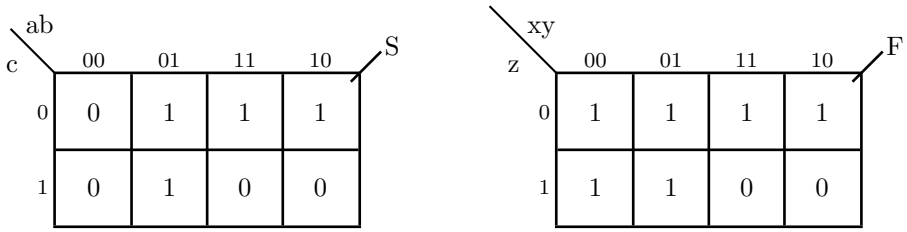


Figure 9: *Different lengths of the variable separator line.*

6 Drawing covers and more

One of the uses of Karnaugh maps is to simplify logic functions. The simplifications are usually shown as (part of) circles or ovals (also squares can be used) combining the function values that make up the simplified terms (either product terms or sum terms).

Let's begin with a simple example. The code is shown below, the map is shown in Figure 10. Note that one function variable is covered by all three ovals, hence it looks as if it is covered by a circle.

```
\askmapiii{s}{abc}{}{00010111}{%
\put(2.5,1.0){\oval(0.8,1.8)}%
\put(2.0,0.5){\oval(1.8,0.8)}%
\put(3.0,0.5){\oval(1.8,0.8)}%
}%
```

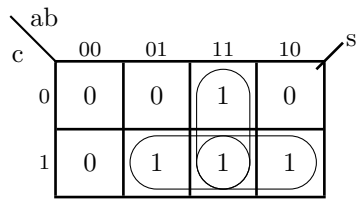


Figure 10: *Karnaugh map for three variables.*

Function values on the edges of Karnaugh maps can be combined in one term. The values are not adjacent in the maps so covering oval or circle has to be split. This is shown in Figure 11. The code is shown left. As can be seen, for marking the edges only half a circle has to be drawn. Note the lines extending the edges of the map.

```
\askmapiii{s}{abc}{}{11100111}{%
\put(2.5,1.0){\oval(0.8,1.8)}%
\put(1.0,1.5){\oval(1.8,0.8)}%
\put(3.5,0.5){\oval(0.8,0.8)[l]}%
\put(3.5,0.9){\line(1,0){0.7}}%
\put(3.5,0.1){\line(1,0){0.7}}%
\put(0.5,0.5){\oval(0.8,0.8)[r]}%
\put(0.5,0.9){\line(-1,0){0.7}}%
\put(0.5,0.1){\line(-1,0){0.7}}%
}%
```

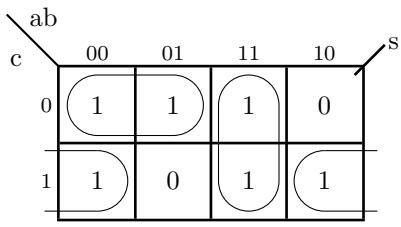


Figure 11: *Example of how to mark edges.*

Making “square” ovals doesn’t always produce a good result as can be seen in Figure 12. The column second to the right seems to be covered by it’s own oval.

```
\askmapiii{s}{abc}{}{00111111}{%
\put(2.5,1.0){\oval(0.8,1.8)[rb]}%
\put(2.5,1.0){\oval(0.8,1.8)[rt]}%
\put(1.5,1.0){\oval(0.8,1.8)[lb]}%
\put(1.5,1.0){\oval(0.8,1.8)[lt]}%
\put(1.5,1.9){\line(1,0){1.0}}%
\put(1.5,0.1){\line(1,0){1.0}}%
\put(3.5,1.0){\oval(0.8,1.8)[rb]}%
\put(3.5,1.0){\oval(0.8,1.8)[rt]}%
\put(2.5,1.0){\oval(0.8,1.8)[lb]}%
\put(2.5,1.0){\oval(0.8,1.8)[lt]}%
\put(2.5,1.9){\line(1,0){1.0}}%
\put(2.5,0.1){\line(1,0){1.0}}%
}%
```

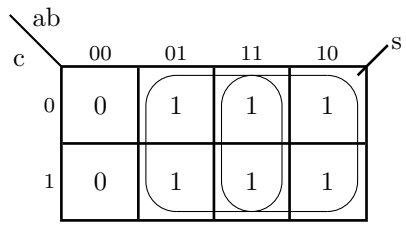


Figure 12: *Example of bad use of ovals.*

Let’s try a four-variable map, see Figure 13 and accompanying code. Here we have deliberately used an oval with a bigger size (see code line marked $*$) otherwise it will interfere with the covering of the lower left and right edges. An example of good use of a “square” oval can be found in Figure 14.

```
\askmapiv{s}{abcd}{}{0111010001110111}{%
\put(2.0,2.5){\oval(3.8,0.8)}%
\put(0.5,1.0){\oval(0.8,1.8)[rb]}%
\put(0.5,1.0){\oval(0.8,1.8)[rt]}%
\put(0.5,1.9){\line(-1,0){0.7}}%
\put(0.5,0.1){\line(-1,0){0.7}}%
\put(3.0,1.0){\oval(1.8,1.8)}% (*)
\put(3.5,1.0){\oval(0.8,1.8)[lb]}%
\put(3.5,1.0){\oval(0.8,1.8)[lt]}%
\put(3.5,1.9){\line(1,0){0.7}}%
\put(3.5,0.1){\line(1,0){0.7}}%
}%
```

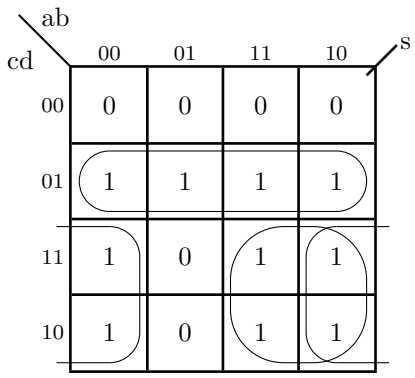


Figure 13: *Example of how to mark edges.*

Every textbook on using Karnaugh maps shows how to combine the corners of a four-variable map. The complete code is shown below. The result is shown in Figure 14. Each corner marking consists of three drawing command: two lines (to the left, right up or down) and one (part of a) oval (right-bottom, left-bottom, right-top and right-bottom).

```
\begin{figure}[H]
\centering
\askmapiv{s}{abcd}{}{1010000010100000}{%
\put(0.5,3.5){\oval(0.8,0.8)[rb]}% left-top
\put(0.5,3.1){\line(-1,0){0.7}}%
\put(0.9,3.5){\line(0,1){0.7}}%
\put(3.5,3.5){\oval(0.8,0.8)[lb]}% right-top
\put(3.5,3.1){\line(1,0){0.7}}%
\put(3.1,3.5){\line(0,1){0.7}}%
\put(0.5,0.5){\oval(0.8,0.8)[rt]}% left-bottom
\put(0.5,0.9){\line(-1,0){0.7}}%
\put(0.9,0.5){\line(0,-1){0.7}}%
```

```
\put(3.5,0.5){\oval(0.8,0.8)[lt]}% right-bottom
\put(3.5,0.9){\line(1,0){0.7}}%
\put(3.1,0.5){\line(0,-1){0.7}}%
\put(2.5,2.0){\oval(0.8,1.8)[rb]}% the square oval
\put(2.5,2.0){\oval(0.8,1.8)[rt]}% in the middle
\put(1.5,2.0){\oval(0.8,1.8)[lb]}% of the map
\put(1.5,2.0){\oval(0.8,1.8)[lt]}%
\put(1.5,2.9){\line(1,0){1.0}}%
\put(1.5,1.1){\line(1,0){1.0}}%
}%
\caption{The corners of the Karnaugh map are connected.}
\label{fig:kmapforfourvarcorner}
\end{figure}
```

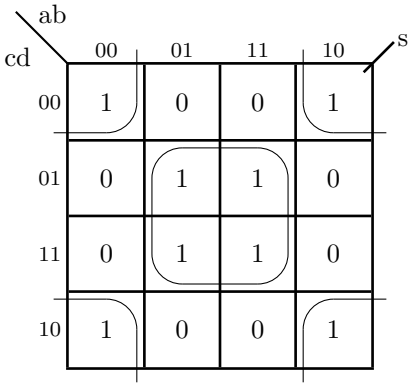
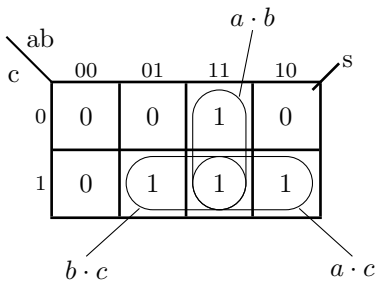


Figure 14: *The corners of the Karnaugh map are connected.*

When writing a textbook on the use of Karnaugh maps, it sometimes helps if a cover is accompanied with the term it covers. Below is a fine example of explaining the simplification of a logic function by showing the product term at each cover. The truth table is shown at the right.

```
\begin{figure}[H]
\begin{subfigure}[c]{0.5\textwidth}
\centering
\askmapiii{s}{abc}{}{00010111}{%
\put(2.5,1.0){\oval(0.8,1.8)}%
\put(3.0,3.0){\makebox(0,0){$a\cdot b$}}%
\put(3.0,2.8){\line(-1,-5.0){0.2}}%
\put(2.0,0.5){\oval(1.8,0.8)}%
\put(0.5,-0.8){\makebox(0,0){$b\cdot c$}}%
\put(0.5,-0.6){\line(1,0.9){0.8}}%
\put(3.0,0.5){\oval(1.8,0.8)}%
\put(4.5,-0.8){\makebox(0,0){$a\cdot c$}}%
\put(4.5,-0.6){\line(-1,0.9){0.8}}%
}%
\end{subfigure}%
\begin{subfigure}[c]{0.5\textwidth}
\centering
\begin{tabular}{ccc|c}
$a$ & $b$ & $c$ & $s$ \\ \hline
 0  &  0  &  0  &  0  \\
 0  &  0  &  1  &  0  \\
```

```
 0  &  1  &  0  &  0  \\
 0  &  1  &  1  &  1  \\
 1  &  0  &  0  &  0  \\
 1  &  0  &  1  &  1  \\
 1  &  1  &  0  &  1  \\
 1  &  1  &  1  &  1  \\
\end{tabular}
\end{subfigure}%
\bigskip\caption{Karnaugh map for three variables.}
\label{fig:karnaughmajorfunctie}
\end{figure}
```



a	b	c	s
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

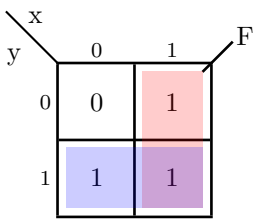
Figure 15: *Karnaugh map for three variables.*

A very notable feature is when you use Ti*k*Z. When you use the package with

```
\usepackage{tikz}
```

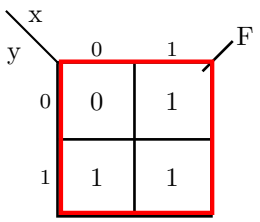
you can use Ti*k*Z' opacity control to add opaque solid rectangles in the back:

```
\askmapii{F}{xy}{f}{0111}{
\setlength\fboxsep{0pt}\linethickness{0pt}
\put(1.1,0.1){\pgfsetfillopacity{0.2}\colorbox{red}{\framebox(0.8,1.8){}}\pgfsetfillopacity{1}}%
\put(0.1,0.1){\pgfsetfillopacity{0.2}\colorbox{blue}{\framebox(1.8,0.8){}}\pgfsetfillopacity{1}}%
}
```



In fact, you can use a `tikzpicture` environment to use Ti*k*Z commands but you have to set an invisible point to coordinate to (0,0):

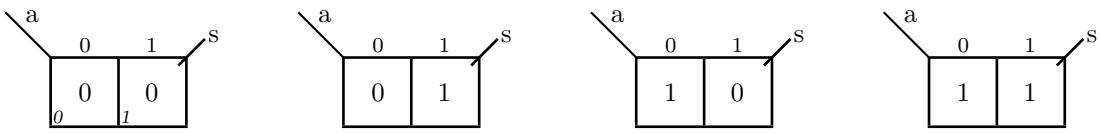
```
\askmapii{F}{xy}{f}{0111}{%
\begin{tikzpicture}[x=\askmapunitlength,y=\askmapunitlength]%
\draw[draw=none,fill=none] (0,0) -- (0,0); % Set default corner
\draw[red, ultra thick] (0,0) rectangle (2,2);%
\end{tikzpicture}%
}
```



7 Examples

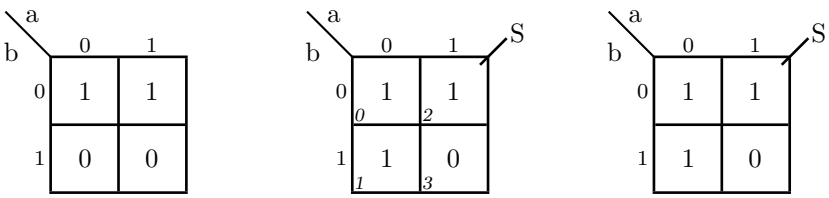
The most simple Karnaugh maps you can make with only one variable.

```
\askmapi{s}{a}{i}{00}{}
\askmapi{s}{a}{}{01}{}
\askmapi{s}{a}{}{10}{}
\askmapi{s}{a}{}{11}{}
```



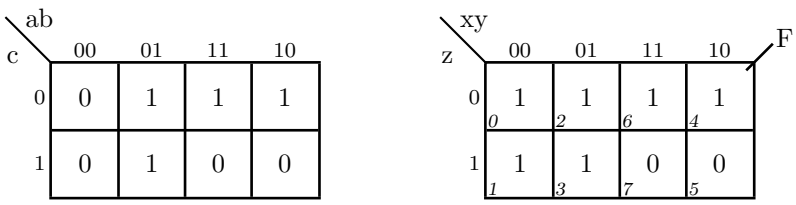
The Karnaugh maps for two variables: plain, with indexes and output variable.

```
\askmapii{S}{ab}{F}{1010}{}
\askmapii{S}{ab}{i}{1110}{}
\askmapii{S}{ab}{}{1110}{}
```



The Karnaugh maps for three variables: plain, with indexes and function output variable.

```
\askmapiii{S}{abc}{F}{00111010}{}
\askmapiii{F}{xyz}{i}{11111010}{}
```



The Karnaugh maps for four variables: plain, with function output variable and indexes.

```
\askmapiv{f}{wxyz}{F}{1110001101000101}{}
\askmapiv{S}{abcd}{i}{1100100100011101}{}
```

yz \ wx	00	01	11	10
00	1	0	0	0
01	1	0	1	1
11	0	1	1	0
10	1	1	0	0

cd \ ab	00	01	11	10
00	1 (0)	1 (4)	1 (12)	0 (8)
01	1 (1)	0 (5)	1 (13)	0 (9)
11	0 (3)	1 (7)	1 (15)	1 (11)
10	0 (2)	0 (6)	0 (14)	0 (10)

S

You can also do math things by using the known $ signs... and you can make an empty map for your exams...

```
\askmapii{$S_{0}$}{{$a_{1}$}{$a_{0}$}}{}{1010}{}
\askmapii{}{}{F}{}{}
```

a_0 \ a_1	0	1
0	1	1
1	0	0

S_0

	0	1
0		
1		

You can do the math thing in roman upright font... and of course you can make a *really* empty map for your exams...

```
\askmapiii{$\mathrm{M^{n+1}_{0}}$}{{$\mathrm{M^{n}_{2}}$}{$\mathrm{M^{n}_{1}}$}
          {$\mathrm{M^{n}_{0}}$}}{}{11100111}{}
\askmapiii{}{}{BF}{}{}
```

$\mathrm{M^{n}_{0}}$ \ $\mathrm{M^{n}_{2}M^{n}_{1}}$	00	01	11	10
0	1	1	1	0
1	1	0	1	1

$\mathrm{M^{n+1}_{0}}$

The vertical version of the three variable map, used in a `minipage` environment:

```
\begin{minipage}[c]{0.25\textwidth}
\askmapiiialt{s}{{c$_{in}$}ab}{}{01101001}{}
\end{minipage}
```

```
\begin{minipage}[c]{0.25\textwidth}
\askmapiiialt{c$_{out}$}{{c$_{in}$}ab}{}{00010111}{}
\end{minipage}\hfill%
\begin{minipage}[c]{0.40\textwidth}
As you can see on the left, the function of the sum of the full adder cannot
be simplified as all 1's are placed diagonal in the map. This means that the
function for the sum can be expressed with XOR ports.

The carry-out function on the otherhand can be simplified.
\end{minipage}
```

ab \ c_{in}	0	1
00	0	1
01	1	0
11	0	1
10	1	0

S

ab \ c_{in}	0	1
00	0	0
01	0	1
11	1	1
10	0	1

c_{out}

As you can see on the left, the function of the sum of the full adder cannot be simplified as all 1's are placed diagonal in the map. This means that the function for the sum can be expressed with XOR ports.
The carry-out function on the otherhand can be simplified.

You can do things with don't cares...

```
\askmapii{S}{ab}{}{011-}{}
\askmapiii{S}{abc}{}{001--10-}{}
```

b \ a	0	1
0	0	1
1	1	-

S

c \ ab	00	01	11	10
0	0	1	0	-
1	0	-	-	1

S

You can use colors and empty function values and variables as values too...

```
\askmapiii{S}{abc}{}{{\color{blue}{0}}{\color{blue}{0}}{\color{red}{1}}{ }{ }
            {\color{red}{1}}{\color{blue}{0}}{ }}{}
\askmapii{S}{ab}{}{{$i_{0}$}{$i_{1}$}{$i_{2}$}{$i_{3}$}}{}
```

c \ ab	00	01	11	10
0	0	1	0	
1	0			1

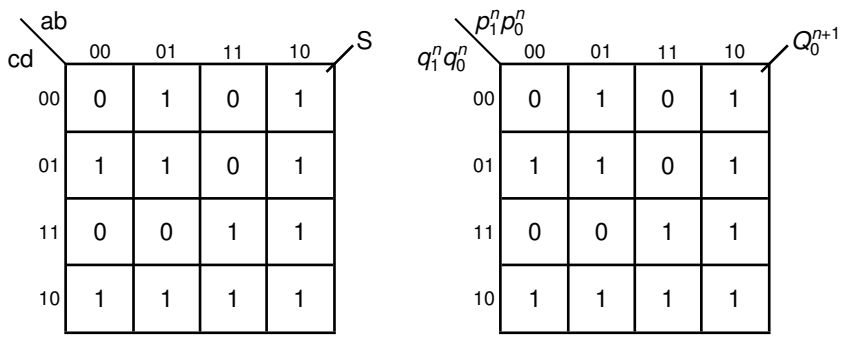
S

b \ a	0	1
0	i_0	i_2
1	i_1	i_3

S

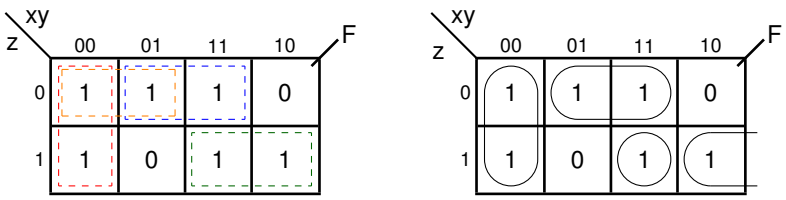
You can set the font to something else ... and use sans math font ...

```
{\fontfamily{phv}\selectfont%
\askmapiv{S}{abcd}{}{0110111011110011}{}%
}
{\fontfamily{phv}\selectfont\sansmath
\askmapiv{$Q^{n+1}_{0}$}{{$p^{n}_{1}$}{$p^{n}_{0}$}{$q^{n}_{1}$}{$q^{n}_{0}$}}
         {}{0110111011110011}{}%
}
```



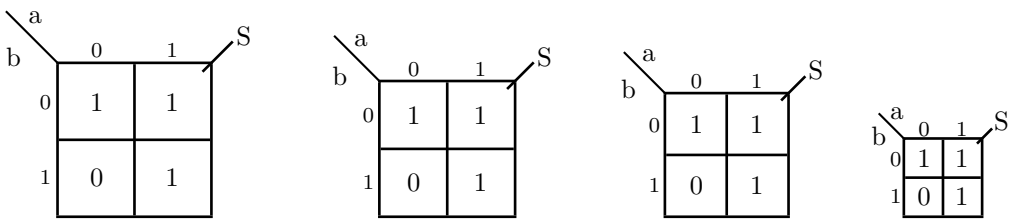
You can use the last parameter to create to your own picture commands. Note that the origin of the picture (0,0) is at the lower left corner of the lower left square. It is also possible to use `\raisebox` in conjuction with horizontal spacing to adjust the placing of the left (least significant) variable(s).

```
{\fontfamily{phv}\selectfont\sansmath
\askmapiii{F}{xy{\raisebox{2ex}{\ \ z}}}{f}{11100111}{% raise Z input
\color{red}\put(0.1,0.1){\dashbox{0.1}(0.8,1.8){}}%
\color{blue}\put(1.1,1.1){\dashbox{0.1}(1.8,0.8){}}%
\color{darkgreen}\put(2.1,0.1){\dashbox{0.1}(1.8,0.8){}}%
\color{orange}\put(0.15,1.15){\dashbox{0.1}(1.7,0.7){}}%
}}%
{\fontfamily{phv}\selectfont\sansmath
\askmapiii{F}{xyz}{}{11100111}{%
\put(0.5,1.0){\oval(0.8,1.8)}%
\put(2.0,1.5){\oval(1.8,0.8)}%
\put(2.5,0.5){\oval(0.8,0.8)}% circle
\put(3.5,0.5){\oval(0.8,0.8)[l]}% these three create
\put(3.5,0.9){\line(1,0){0.6}}% a oval open to the left
\put(3.5,0.1){\line(1,0){0.6}}%
}}%
```



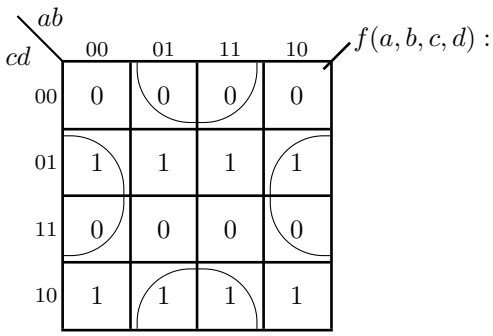
You can set the length of the squares, but please note that this will also affect the rendering of the text...

```
\askmapunitlength=1cm
\askmapii{S}{ab}{}{1011}{}
\askmapunitlength=0.88cm
\askmapii{S}{ab}{}{1011}{}
\askmapunitlength=0.8cm
\askmapii{S}{ab}{}{1011}{}
\askmapunitlength=0.5cm
\askmapii{S}{ab}{}{1011}{}
```



An example directly from the Karnaugh package:

```
\kvnoindex
\askmap{4}{$f(a,b,c,d):$}{{$a$}{$b$}{$c$}{$d$}}%
{0110011001100110}%
{%
\put(0,2){\oval(1.9,1.9)[r]}
\put(4,2){\oval(1.9,1.9)[l]}
\put(2,0){\oval(1.9,1.9)[t]}
\put(2,4){\oval(1.9,1.9)[b]}
}
```



but please note that the ovals do not cover the logic 1's in the Karnaugh map because the function values are placed in a different order.

One can use Ti*k*Z. Please note you have to set a default corner to (0,0), because Ti*k*Z will create a bounding box around the the `tikzpicture`:

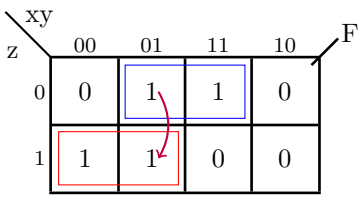
```
A static hazard can be found by inspecting the Karnaugh map and see if a change
of minterms will cross implicants:

\askmapiii{F}{xyz}{}{011100100}{%
\begin{tikzpicture}[x=\askmapunitlength,y=\askmapunitlength]%
```

```
\draw[draw=none,fill=none] (0,0) -- (0,0); % Set default corner
\draw[red] (0.1,0.1) rectangle (1.9,0.9);
\draw[blue] (1.1,1.1) rectangle (2.9,1.9);
\draw[purple,thick,->] (1.6,1.5) to[bend left] (1.6,0.5);
\end{tikzpicture}%
}

The function is $F = \overline{x}z + y\overline{z}$. But if the circuit is
changing from minterm $m_2$ to $m_3$, a static hazard will occur as can
be seen by the arrow.
```

A static hazard can be found by inspecting the Karnaugh map and see if a change of minterms will cross implicants:



The function is $F = \overline{x}z + y\overline{z}$. But if the circuit is changing from minterm m_2 to m_3, a static hazard will occur as can be seen by the arrow.

8 Changelog

```
 4-dec-2013 -- v0.1
   initial release
19-aug-2020 -- v0.2
   added option disablef
   added askmapi - a K-map for one variable
   added askmapiiialt - alternate (vertical) version op K-map III
   added formatting index size, contents size and bit combination size.
   added support for \kvindex and \nokvindex macros when kvmacros.tex is loaded
   added formatting for left-placed variables
   implemented a debug possibility
   implemented linelength of variable separator
   resolved an issue with multiple overfull \hbox'es
   added a section on creating covers
   added examples with Tikz
```

9 The source code

The source code is pretty straight forward. It uses a lot of picture primitives for drawing the maps. The package uses three macros for processing variable-length parameters, which were completely reused from the `kvmacros` package.

```
%%
%% askmaps.sty
%%
%% American Style K-maps (1, 2, 3, 4 and 5 variables)
%%
%% (c)2020 Jesse op den Brouw, <J.E.J.opdenBrouw@hhs.nl>
%%
%% LPPL LaTeX Public Project License
%%
%%
%% Original rel 0.1 : 2013-12-04
%% Update   rel 0.2 : 2020-06-09
%%   added option oldstylef
%%   added option newstylef
%%   added askmapi - a K-map for one variable
%%   added askmapiiialt - alternate (vertical) version op K-map III
%%   added formatting index size, contents size and bit combination size.
%%   added support for \kvindex and \nokvindex macros when kvmacros.tex is
%%     loaded
%%   added formatting for left-placed variables
%%   implemented a debug possibility
%%   implemented linelength of variable separator
%%   resolved an issue with multiple overfull \hbox'es
%%   added a section on creating covers
%%   changed the use of \askmapindexsize et al. to include the size and
%%     shape of the fonts used
%%
%%
\NeedsTeXFormat{LaTeX2e}[1994/06/01]
%% Version and date
\gdef\askmapsversion{0.2}
\gdef\askmapsdate{2020/06/09}
%
\ProvidesPackage{askmaps}[\askmapsdate\space Version \askmapsversion\space
      LaTeX package for drawing American Style Karnaugh maps]

%% This package provides 1, 2, 3, 4 and 5 variable Karnaugh maps that can
%% be found in numerous american text books on digital design. Note
%% that it is possible to draw many different K-maps that are all
%% legitimate as long as a square (minterm) differs only in one
%% literal with its adjacent squares. There is an alternate form of the
%% 3 variable K-map available.
%% This package draws K-maps where the most significant input variables
%% are placed on top of the columns and the least significant variables
%% are placed left of the rows.

%% Packages needed
\RequirePackage{pict2e}

%% Get the only possible option
\newcommand{\askmap@usecapf}{\askmap@optftrue}
\DeclareOption{disablef}{\renewcommand{\askmap@usecapf}{\askmap@optffalse}}
\DeclareOption*{\PackageWarning{askmaps}{Unknown '\CurrentOption' in package
    askmaps}}
\ProcessOptions\relax

%% For debugging purposes, currently draw rectangles around the maps
%% Use only for debugging purposes
```

```
\newif\ifaskmap@debug\askmap@debugfalse

%% The macros \askmapargumentstring, \askmapgetchar and \askmapgetonechar are
%% needed to process the variable-length parameters in \askmapxxx:
%% These routine politely reused from karnaugh package
\def\askmapargumentstring#1{\gdef\askmapdummystring{#1{}\noexpand\end}}
\def\askmapgetchar{\expandafter\askmapgetonechar\askmapdummystring}
\def\askmapgetonechar#1#2\end{{#1}\gdef\askmapdummystring{#2\noexpand\end}}%

%% Dimension for length of one square
\newdimen\askmapunitlength
\askmapunitlength=10mm

%% The sizes (and shapes) of the text in the maps
\newcommand{\askmapindexsize}{\scriptsize\slshape}
\newcommand{\askmapcontentsize}{\normalsize}
\newcommand{\askmapbitcombinationsize}{\footnotesize}

%% Length of the line separating the input variables top-right and
%% middle-left
\newcommand{\askmapvarsep}{0.70}

%% Some use options
\newif\ifaskmap@opti% want indexes
\newif\ifaskmap@optf% want output function printed
\newif\ifaskmap@optb% want input variable bit combinations at columns and rows

%% Process user options
\def\askmap@processoptions#1{%
    \askmap@optifalse\askmap@usecapf\askmap@optbtrue%
    \@tfor\askmap@opt:=#1\do{%
        \if i\askmap@opt\askmap@optitrue\fi%
        \if I\askmap@opt\askmap@optifalse\fi%
        \if f\askmap@opt\askmap@optftrue\fi%
        \if F\askmap@opt\askmap@optffalse\fi%
        \if b\askmap@opt\askmap@optbtrue\fi%
        \if B\askmap@opt\askmap@optbfalse\fi%
        \if c\askmap@opt\askmap@optitrue\askmap@optftrue\askmap@optbtrue\fi%
        \if C\askmap@opt\askmap@optifalse\askmap@optffalse\askmap@optbfalse\fi%
    }%
}%

%% Replacement for macros from karnaugh package
%% #1 = number of input variables
%% #2 = function output variable
%% #3 = function input variables
%% #4 = list of function values
%% #5 = user picture commands
%\newcommand{\askmap}[5]{}%
% Starred version or not
%\@ifstar{\gdef\askmap@printoutputname{f}\askmap@askmap}{\gdef\
    askmap@printoutputname{F}\askmap@askmap}%
%}
%\newcommand{\askmap@askmap}[5]{%
\newcommand{\askmap}[5]{%
\def\askmap@printoutputname{f}
%% Check if the \kvcurrentindex is set to using indexes
\def\askmap@printindex{i}
\ifx\kvcurrentindex\undefined
\def\askmap@printindex{I}
\else
\ifx\kvcurrentindex\empty
\def\askmap@printindex{I}
\fi
\fi
%% Sort out the 'call' to the right macro
```

```
\ifcase#1\relax
    \PackageWarning{askmaps}{Sorry, no K-maps with #1 variables supported by
package askmaps!}
\or
    \askmapi{#2}{#3}{\askmap@printoutputname\askmap@printindex}{#4}{#5}%
\or
    \askmapii{#2}{#3}{\askmap@printoutputname\askmap@printindex}{#4}{#5}%
\or
    \askmapiii{#2}{#3}{\askmap@printoutputname\askmap@printindex}{#4}{#5}%
\or
    \askmapiv{#2}{#3}{\askmap@printoutputname\askmap@printindex}{#4}{#5}%
\or
    \askmapv{#2}{#3}{\askmap@printoutputname\askmap@printindex}{#4}{#5}%
\else
    \PackageWarning{askmaps}{Sorry, no K-maps with #1 variables supported by
package askmaps!}
\fi
}%

%% #1= output function variable
%% #2= 1 input variable
%% #3= options list
%% #4= 2 function values
%% #5= user defined picture commands
\newcommand{\askmapi}[5]{%
{\unitlength\askmapunitlength%
\askmap@processoptions{#3}%
\ifaskmap@debug\fboxsep0pt\fbox{\else\fi%
\begin{picture}(4.4,2)(-1.2,0)%
\linethickness{1pt}%
\put(0,0){\framebox(2,1){}}%
\put(1,0){\line(0,1){1}}%
\ifaskmap@optf%
\put(2.35,1.25){\askmapcontentsize#1}%
\put(1.9,0.9){\line(1,1){0.4}}%
\fi%
\askmapargumentstring{#2}%
\put(-0.4,1.55){\askmapcontentsize\askmapgetchar}%
\ifaskmap@opti%
\put(0.01,0.03){\askmapindexsize 0}%
\put(1.03,0.03){\askmapindexsize 1}%
\fi%
\ifaskmap@optb%
\put(0.5,1.1){\makebox(0,0)[b]{\askmapbitcombinationsize 0}}%
\put(1.5,1.1){\makebox(0,0)[b]{\askmapbitcombinationsize 1}}%
\fi%
\askmapargumentstring{#4}%
\put(0.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\thicklines%
\put(0,1){\line(-1,1){\askmapvarsep}}%
\thinlines%
#5%
\end{picture}%
\ifaskmap@debug}\else\fi%
}%
}% end \askmapi

%% #1= output function variable
%% #2= 2 input variables
%% #3= options list
%% #4= 4 function values
%% #5= user defined picture commands
\newcommand{\askmapii}[5]{%
{\unitlength\askmapunitlength%
\askmap@processoptions{#3}%
```

```
\ifaskmap@debug\fboxsep0pt\fbox{\else\fi%
\begin{picture}(4.4,3)(-1.2,0)%
\linethickness{1pt}%
\put(0,0){\framebox(2,2){}}%
\put(1,0){\line(0,1){2}}%
\put(0,1){\line(1,0){2}}%
\ifaskmap@optf%
\put(2.35,2.25){\askmapcontentsize#1}%
\put(1.9,1.9){\line(1,1){0.4}}%
\fi%
\askmapargumentstring{#2}%
\put(-0.4,2.55){\askmapcontentsize\askmapgetchar}%
%\put(-0.9,2.1){\askmapcontentsize\askmapgetchar}%
%\put(-1.0,1.9){\framebox(0.5,0.5)[r]{\askmapcontentsize\askmapgetchar}}%
\put(-0.5,2.1){\makebox(0.0,0.0)[r]{\askmapcontentsize\askmapgetchar\
    askmapgetchar}}%
%\put(-0.9,2.1){\makebox(0.5,0){\askmapcontentsize\askmapgetchar}}%
\ifaskmap@opti%
\put(0.01,1.05){\askmapindexsize 0}% a little bit to the edge because
\put(0.01,0.03){\askmapindexsize 1}% the edge line is thick
\put(1.03,1.05){\askmapindexsize 2}%
\put(1.03,0.03){\askmapindexsize 3}%
\fi%
\ifaskmap@optb%
\put(-0.6,1.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 0}}%
\put(-0.6,0.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 1}}%
\put(0.5,2.1){\makebox(0,0)[b]{\askmapbitcombinationsize 0}}%
\put(1.5,2.1){\makebox(0,0)[b]{\askmapbitcombinationsize 1}}%
\fi%
\askmapargumentstring{#4}%
\put(0.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(0.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\thicklines%
\put(0,2){\line(-1,1){\askmapvarsep}}%
\thinlines%
#5%
\end{picture}%
\ifaskmap@debug}\else\fi%
}% end \askmapii
}

%% #1= output function variable
%% #2= 3 input variables
%% #3= options list
%% #4= 8 function values
%% #5= user defined picture commands
\newcommand{\askmapiii}[5]{%
{\unitlength\askmapunitlength%
\askmap@processoptions{#3}%
\ifaskmap@debug\fboxsep0pt\fbox{\else\fi%
\begin{picture}(6.4,3)(-1.2,0)%
\linethickness{1pt}%
\put(0,0){\framebox(4,2){}}%
\multiput(1,0)(1,0){3}{\line(0,1){2}}%
\put(0,1){\line(1,0){4}}%
\ifaskmap@optf%
\put(4.35,2.25){\askmapcontentsize#1}%
\put(3.9,1.9){\line(1,1){0.4}}%
\fi%
\askmapargumentstring{#2}%
\put(-0.4,2.55){\askmapcontentsize\askmapgetchar\askmapgetchar}%
%\put(-0.9,2.1){\framebox(0.5,0.5)[rt]{\askmapcontentsize\askmapgetchar}}%
%\put(-1.0,1.9){\framebox(0.5,0.5)[rt]{\askmapcontentsize\askmapgetchar}}%
%\put(-1.0,1.9){\framebox(0.5,0.5)[r]{\askmapcontentsize\askmapgetchar}}%
```

```
%\put(-0.9,2.1){\askmapcontentsize\askmapgetchar}%
%\put(-0.9,2.1){\makebox(0.5,0){\askmapcontentsize\askmapgetchar}}%
\put(-0.5,2.1){\makebox(0.0,0.0)[r]{\askmapcontentsize\askmapgetchar\
    askmapgetchar}}%
%\put(-1.0,1.9){\makebox(0.5,0.45)[rt]{\askmapcontentsize\askmapgetchar}}%
\ifaskmap@opti%
\put(0.01,1.05){\askmapindexsize 0}%
\put(0.01,0.03){\askmapindexsize 1}%
\put(1.03,1.05){\askmapindexsize 2}%
\put(1.03,0.03){\askmapindexsize 3}%
\put(3.03,1.05){\askmapindexsize 4}%
\put(3.03,0.03){\askmapindexsize 5}%
\put(2.03,1.05){\askmapindexsize 6}%
\put(2.03,0.03){\askmapindexsize 7}%
\fi%
\askmapargumentstring{#4}%
\put(0.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(0.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(3.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(3.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(2.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(2.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\ifaskmap@optb%
\put(-0.6,1.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 0}}%
\put(-0.6,0.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 1}}%
\put(0.5,2.1){\makebox(0,0)[b]{\askmapbitcombinationsize 00}}%
\put(1.5,2.1){\makebox(0,0)[b]{\askmapbitcombinationsize 01}}%
\put(2.5,2.1){\makebox(0,0)[b]{\askmapbitcombinationsize 11}}%
\put(3.5,2.1){\makebox(0,0)[b]{\askmapbitcombinationsize 10}}%
\fi%
\thicklines%
\put(0,2){\line(-1,1){\askmapvarsep}}%
\thinlines%
#5%
\end{picture}%
\ifaskmap@debug}\else\fi%
}%
}% end \askmapiii

%% #1= output function variable
%% #2= 4 input variables
%% #3= options list
%% #4= 16 function values
%% #5= user defined picture commands
\newcommand{\askmapiiialt}[5]{%
{\unitlength\askmapunitlength%
\askmap@processoptions{#3}%
\ifaskmap@debug\fboxsep0pt\fbox{\else\fi%
\begin{picture}(4.4,5)(-1.2,0)%
\linethickness{1pt}%
\put(0,0){\framebox(2,4){}}%
\put(1,0){\line(0,1){4}}%
\multiput(0,1)(0,1){3}{\line(1,0){2}}%
\ifaskmap@optf%
\put(2.35,4.25){\askmapcontentsize#1}%
\put(1.9,3.9){\line(1,1){0.4}}%
\fi%
\askmapargumentstring{#2}%
\put(-0.4,4.55){\askmapcontentsize\askmapgetchar}%
%\put(-1.0,4.1){\framebox(0.5,0){\askmapcontentsize\askmapgetchar\askmapgetchar
    }}%
%\put(-1.0,3.9){\framebox(0.5,0.5)[r]{\askmapcontentsize\askmapgetchar\
    askmapgetchar}}%
\put(-0.5,4.1){\makebox(0.0,0.0)[r]{\askmapcontentsize\askmapgetchar\
```

```
    askmapgetchar}}%
%\put(-0.9,4.1){\makebox(0.5,0){\askmapcontentsize\askmapgetchar\askmapgetchar}}%
\ifaskmap@opti%
\put(0.01,3.05){\askmapindexsize 0}%
\put(0.01,2.05){\askmapindexsize 1}%
\put(0.01,0.03){\askmapindexsize 2}%
\put(0.01,1.05){\askmapindexsize 3}%
\put(1.03,3.05){\askmapindexsize 4}%
\put(1.03,2.05){\askmapindexsize 5}%
\put(1.03,0.03){\askmapindexsize 6}%
\put(1.03,1.05){\askmapindexsize 7}%
\fi%
\askmapargumentstring{#4}%
\put(0.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(0.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(0.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(0.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\ifaskmap@optb%
\put(-0.6,3.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 00}}%
\put(-0.6,2.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 01}}%
\put(-0.6,1.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 11}}%
\put(-0.6,0.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 10}}%
\put(0.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 0}}%
\put(1.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 1}}%
\fi
\thicklines%
\put(0,4){\line(-1,1){\askmapvarsep}}%
\thinlines%
#5%
\end{picture}%
\ifaskmap@debug}\else\fi%
}%
}% end \askmapiiialt

%% #1= output function variable
%% #2= 4 input variables
%% #3= options list
%% #4= 16 function values
%% #5= user defined picture commands
\newcommand{\askmapiv}[5]{%
{\unitlength\askmapunitlength%
\askmap@processoptions{#3}%
\ifaskmap@debug\fboxsep0pt\fbox{\else\fi%
\begin{picture}(6.4,5)(-1.2,0)%
\linethickness{1pt}%
\put(0,0){\framebox(4,4){}}%
\multiput(1,0)(1,0){3}{\line(0,1){4}}%
\multiput(0,1)(0,1){3}{\line(1,0){4}}%
\ifaskmap@optf%
\put(4.35,4.25){\askmapcontentsize#1}%
\put(3.9,3.9){\line(1,1){0.4}}%
\fi%
\askmapargumentstring{#2}%
\put(-0.4,4.55){\askmapcontentsize\askmapgetchar\askmapgetchar}%
%\put(-1.0,4.1){\framebox(0.5,0){\askmapcontentsize\askmapgetchar\askmapgetchar}}%
\put(-0.5,4.1){\makebox(0.0,0.0)[r]{\askmapcontentsize\askmapgetchar\askmapgetchar}}%
%\put(-1.0,4.1){\makebox(0.5,0){\askmapcontentsize\askmapgetchar\askmapgetchar}}%
\ifaskmap@opti%
```

```
\put(0.01,3.05){\askmapindexsize  0}%
\put(0.01,2.05){\askmapindexsize  1}%
\put(0.01,0.03){\askmapindexsize  2}%
\put(0.01,1.05){\askmapindexsize  3}%
\put(1.03,3.05){\askmapindexsize  4}%
\put(1.03,2.05){\askmapindexsize  5}%
\put(1.03,0.03){\askmapindexsize  6}%
\put(1.03,1.05){\askmapindexsize  7}%
\put(3.03,3.05){\askmapindexsize  8}%
\put(3.03,2.05){\askmapindexsize  9}%
\put(3.03,0.03){\askmapindexsize 10}%
\put(3.03,1.05){\askmapindexsize 11}%
\put(2.03,3.05){\askmapindexsize 12}%
\put(2.03,2.05){\askmapindexsize 13}%
\put(2.03,0.03){\askmapindexsize 14}%
\put(2.03,1.05){\askmapindexsize 15}%
\fi%
\askmapargumentstring{#4}%
\put(0.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(0.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(0.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(0.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(3.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(3.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(3.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(3.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(2.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(2.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(2.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(2.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\ifaskmap@optb%
\put(-0.6,3.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 00}}%
\put(-0.6,2.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 01}}%
\put(-0.6,1.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 11}}%
\put(-0.6,0.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 10}}%
\put(0.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 00}}%
\put(1.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 01}}%
\put(2.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 11}}%
\put(3.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 10}}%
\fi
\thicklines%
\put(0,4){\line(-1,1){\askmapvarsep}}%
\thinlines%
#5%
\end{picture}%
\ifaskmap@debug}\else\fi%
}%
}% end \askmapiv

%% #1= output function variable
%% #2= 5 input variables
%% #3= options list
%% #4= 32 function values
%% #5= user defined picture commands
\newcommand{\askmapv}[5]{%
{\unitlength\askmapunitlength%
\askmap@processoptions{#3}%
\ifaskmap@debug\fboxsep0pt\fbox{\else\fi%
\begin{picture}(10.4,5)(-1.2,0)%
\linethickness{1pt}%
\put(0,0){\framebox(8,4){}}%
\multiput(1,0)(1,0){7}{\line(0,1){4}}%
```

```
\multiput(0,1)(0,1){3}{\line(1,0){8}}%
\ifaskmap@optf%
\put(8.35,4.25){\askmapcontentsize#1}%
\put(7.9,3.9){\line(1,1){0.4}}%
\fi%
\askmapargumentstring{#2}%
\put(-0.4,4.55){\askmapcontentsize\askmapgetchar\askmapgetchar\askmapgetchar}%
%\put(-1.0,3.9){\framebox(0.5,0.5)[r]{\askmapcontentsize\askmapgetchar\askmapgetchar}}%
%\put(-1.0,4.1){\framebox(0.5,0)[r]{\askmapcontentsize\askmapgetchar\askmapgetchar}}
\put(-0.5,4.1){\makebox(0.0,0.0)[r]{\askmapcontentsize\askmapgetchar\askmapgetchar}}%
%\put(-1.0,4.1){\makebox(0.5,0){\askmapcontentsize\askmapgetchar\askmapgetchar}}%
\ifaskmap@opti%
\put(0.01,3.05){\askmapindexsize  0}%
\put(0.01,2.05){\askmapindexsize  1}%
\put(0.01,0.03){\askmapindexsize  2}%
\put(0.01,1.05){\askmapindexsize  3}%
\put(1.03,3.05){\askmapindexsize  4}%
\put(1.03,2.05){\askmapindexsize  5}%
\put(1.03,0.03){\askmapindexsize  6}%
\put(1.03,1.05){\askmapindexsize  7}%
\put(3.03,3.05){\askmapindexsize  8}%
\put(3.03,2.05){\askmapindexsize  9}%
\put(3.03,0.03){\askmapindexsize 10}%
\put(3.03,1.05){\askmapindexsize 11}%
\put(2.03,3.05){\askmapindexsize 12}%
\put(2.03,2.05){\askmapindexsize 13}%
\put(2.03,0.03){\askmapindexsize 14}%
\put(2.03,1.05){\askmapindexsize 15}%
\put(7.03,3.05){\askmapindexsize 16}%
\put(7.03,2.05){\askmapindexsize 17}%
\put(7.03,0.03){\askmapindexsize 18}%
\put(7.03,1.05){\askmapindexsize 19}%
\put(6.03,3.05){\askmapindexsize 20}%
\put(6.03,2.05){\askmapindexsize 21}%
\put(6.03,0.03){\askmapindexsize 22}%
\put(6.03,1.05){\askmapindexsize 23}%
\put(4.03,3.05){\askmapindexsize 24}%
\put(4.03,2.05){\askmapindexsize 25}%
\put(4.03,0.03){\askmapindexsize 26}%
\put(4.03,1.05){\askmapindexsize 27}%
\put(5.03,3.05){\askmapindexsize 28}%
\put(5.03,2.05){\askmapindexsize 29}%
\put(5.03,0.03){\askmapindexsize 30}%
\put(5.03,1.05){\askmapindexsize 31}%
\fi%
\askmapargumentstring{#4}%
\put(0.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(0.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(0.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(0.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(1.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(3.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(3.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(3.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(3.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(2.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(2.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(2.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(2.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
```

```
\put(7.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(7.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(7.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(7.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(6.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(6.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(6.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(6.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(4.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(4.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(4.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(4.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(5.5,3.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(5.5,2.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(5.5,0.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\put(5.5,1.5){\makebox(0,0){\askmapcontentsize\askmapgetchar}}%
\ifaskmap@optb%
\put(-0.6,3.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 00}}%
\put(-0.6,2.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 01}}%
\put(-0.6,1.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 11}}%
\put(-0.6,0.5){\makebox(0.5,0)[r]{\askmapbitcombinationsize 10}}%
\put(0.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 000}}%
\put(1.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 001}}%
\put(2.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 011}}%
\put(3.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 010}}%
\put(4.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 110}}%
\put(5.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 111}}%
\put(6.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 101}}%
\put(7.5,4.1){\makebox(0,0)[b]{\askmapbitcombinationsize 100}}%
\fi%
\thicklines%
\put(0,4){\line(-1,1){\askmapvarsep}}%
\thinlines%
#5%
\end{picture}%
\ifaskmap@debug}\else\fi%
}%
}% end \askmapv

\endinput
%%
%% End of file 'askmaps.sty'.
```

askmaps.sty