

# The `tikz`-`timing` package

Martin Scharrer

[martin@scharrer-online.de](mailto:martin@scharrer-online.de)

<http://www.ctan.org/pkg/tikz-timing>

Version v0.5 – 2009/05/15

## 1 Introduction

This package uses the `tikz`<sup>1</sup> package to produce timing diagrams inside text or `{tikzpicture}` environments. Also a tabular-like environment is provided to produce a larger timing diagram with multiple labeled signals and the possibility to add own drawing material.







































The signal levels of the timing diagram can be given by corresponding characters/letters like ‘H’ for *Logical High* or ‘L’ for *Logical Low*. So e.g. ‘{HLZXD}’ gives ‘’. Lowercase characters only produce a signal with half the width while uppercase characters produce one with the “full” width, called the ‘period width’ in this document and which is by default identical to there height, called ‘signal height’, which defaults to 1.6ex (about the height of the uppercase ‘X’ of the current font). Table 1 shows all by default defined logic characters and Table 2 all possible transitions. Additional functionality is provided by the “modifiers” shown in Table 3.

Table 1: Timing Characters

Character	Description	Full Width (Uppercase)	Half Width (Lowercase)	Transition Example
H	High			
L	Low			
Z	High Impedance			
X	Undefined / Don't Care			
D	Data / Double			
U	Unknown Data			
T	Toggle	 or 	 or 	 or 
C	Clock (no slope)	 or 	 or 	 or 
M	Metastable Condition			
G	Glitch (zero width)			
S	Space (nothing)			

<sup>1</sup>Part of the `pgf` package, CTAN: <http://www.ctan.org/pkg/pgf>

## 2 Usage

### 2.1 Macro for use in Text Mode

`\texttiming[(initial character/TikZ Settings)]{(characters)}`

This macro places a single timing diagram line into the current text. The signals have the same height as a uppercase letter (like ‘X’) of the current font, i.e. they scale with the font size. The macro argument must contain only valid logic characters and modifiers which define the logical levels of the diagram line.

An initial character can be given as an optional argument. No logic level will be drawn for this character. Instead it will be used to define the initial position of the signal so that the diagram line will start with a transition from the initial to the first character. However, if the optional argument holds more than a single character it is taken as TikZ settings for the diagram. The initial character can then be given using the key ‘`timing/initchar=(char)`’.

*Examples:*

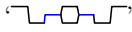
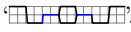
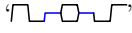
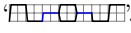

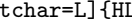
`\texttiming{HLZDZLH}` gives ‘’, with grid: ‘.  
`\texttiming[L]{HLZDZLH}` gives ‘’, with grid: ‘.  
`\texttiming[green]{HLZDZLH}` gives ‘’  
`\texttiming[green,timing/initchar=L]{HLZDZLH}` gives ‘.

Table 2: Overview over all transitions.

$\begin{smallmatrix} \text{to} \\ \text{from} \end{smallmatrix}$	H	L	Z	X	D	U	M	G	S	T	C
H											
L											
Z											
X											
D											
U											
M											
G											
S											
T											
C											

Table 3: Modifiers for Timing Characters

Modifier Syntax	Description	Example Usage	Example Result
D{}D	Produces transition between two data values.	D{}D	
D{<Text>}	Adds text material into a data signal using a node.	D{A}D{B}	
D{[<TikZ Settings>]}<Text>}	Adds text material into a data signal using the given settings.	D{[blue]A}	
<integer>{<characters>}	Repeats the given characters <int> times.	5{h1}	
<number>{<character>}	Sets width of next signal to given number. Half of it if character is in lower case.	2.6H5.21	
<number>B	Subtracts the given number from the width of the next character.	H .5B L	
N[<Settings>](<Name>){<Content>}	Adds node at current position. All three arguments are optional.	H N(a1) L	
[<TikZ Settings>]	Executes given TikZ settings.	H[blue]LH	
;	Renews the internal drawing path which ends the scope of all options given by [].	H;[blue]L;H	
,	Same as ';', but timing specific options (atm.: slopes and line width) are restored for the new path.	H,[lw=1pt]L;H	
!{<code>}	Places given code into the internal {tikzpicture}.	See Example 1 on page 9.	

`\texttimingbefore` (defaults to: `<empty>`)

`\texttimingafter` (defaults to: `<empty>`)

This two macros are executed before and after every timing diagram line created by `\texttiming` macro inside the same `{tikzpicture}` environment and can be used to add drawing macros. The argument of the `\texttiming` macro is already processed before any of these macros are expanded, therefore this macros can access the width of the diagram.

Example: `\let\texttimingbefore\texttiminggrid` adds a grid into the background of the `\texttiming` diagram. A more L<sup>A</sup>T<sub>E</sub>X-stylish way to do this is `\renewcommand*{\textttimingbefore}{\texttiminggrid}`.

`\texttiminggrid`

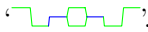
This macro should only be used inside `\texttimingbefore` or `\texttimingafter` and draws a grid of the full size of the `\texttiming` diagram.

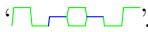
## 2.2 Macro for use inside TikZ-Pictures

```
\timing[TikZ Settings] at (<TikZ Coordinate>) {[<initial character>]}(<characters>);
```

This macro does the same as `\texttiming` but is designed to be used inside a `{tikzpicture}` environment and only there. Like normal TikZ macros (`\path`, `\draw`, `\node`) it allows an optional argument with TikZ settings and an optional TikZ-coordinate. However, a own argument parser, not the one used by TikZ, is used to detect and read these optional arguments. Therefore the order of the arguments is mandatory and might not be reversed. This small limitation might be overcome in future versions of this package.

Please note that the optional initial character may be given *inside* and at the very start of the mandatory argument, not before it. This is necessary because of several technical reasons.

Example: `\tikz \timing [green] at (1,2) {HLZDZLH};` gives .

Example: `\tikz \timing [green] at (1,2) {[L]HLZDZLH};` gives .

### Timing Shape Anchors

Every timing diagram line produced by `\timing`, which includes the rows in `{tikztimingtable}`, is also a PGF shape (node) with several anchors. These are shown in Figure 1. The shape is very similar to the standard `rectangle` shape but does not provide a `text` anchor. In addition to the standard points of the compass anchors of TikZ the three logic levels `low`, `mid` and `high` can be used in combination with `start`, `mid` and `end`. An extra `origin` anchor is located at the lower left, also called `south west` corner where the diagram originates. The two anchors called `start` and `end` are provided to mark the start and end of the timing signal. There are either located at the low, middle or high logic level dependent on the used first (or initial) and last timing character.

In order to use the timing node it has to be named which can be done using the `'name=<name>'` option inside the optional argument. The rows of a `{tikztimingtable}` are automatically named as `'row<row number>'` where the first row has the number 1.

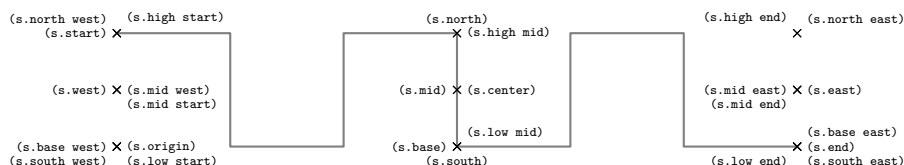


Figure 1: Timing Shape Anchors. The `start` and `end` anchors mark the start and end of the timing signal.

## 2.3 Table for Timing Diagrams

```
\begin{tikztimingtable}[\langle TikZ settings for whole table \rangle]
  \langle Signal Name \rangle & [\langle Init. Char./TikZ Settings for Row \rangle] \langle Characters \rangle \\
  ...
  \extracode % Optional
  \langle additional code \rangle
\end{tikztimingtable}
```

This environment can be used to typeset multi-line timing diagrams. The syntax is like the one for a `{tabular}` environment with two columns. The first column is for the signal name and the second one are the logic characters which would be placed inside the argument of a `\texttiming` or `\timing` macro. If the second column starts with an optional argument it is either taken as initial character if it holds only a single character or as row wide settings otherwise. The whole table will be drawn inside a `{tikzpicture}` environment using multiple `\timing` and `\node` macros for the timing signals and their names, respectively. Additional `tikz` drawing code can be insert at the end of the table using `\extracode`.

### `\extracode`

This macro is only defined inside a `{tikztimingtable}` environment and can only be used after the last table line (i.e. after a `\\`). If used all code between it and the `\end{tikztimingtable}` will be placed inside the same `{tikzpicture}`. This allows to add some drawing lines or a grid to the picture. It is also possible to draw something behind the timing diagram by using the PGF background layer: `\begin{pgfonlayer}{background}... \end{pgfonlayer}`.

### 2.3.1 Macros for `\extracode` Section

The following macros are only defined inside a `{tikztimingtable}` after the macro `\extracode`. They are useful for drawing additional material.

```
\tablegrid[\langle TikZ Settings \rangle]
```

```
\fulltablegrid[\langle TikZ Settings \rangle]
```

After `\extracode` this macros draw a grid in the background of the table. The first one draws a separate grid for each row and the second one a big grid over all rows.

```
\rowdist
```

```
\coldist
```

This macros return the row and column distance. There are useful for drawing additional material relative to the rows and columns. This values can be set (e.g. in the optional argument of the table) using the `timing/rowdist` and `timing/coldist` settings which are explained in Section 3.

### `\nrows`

Returns the number of rows in the current table. Useful for use in `\horlines`.

### `\twidth`

Returns the width (as multiple of the ‘period width’) of the longest timing diagram line in the table.

Example: If the longest line would be ‘H 2.3L z’ than `\twidth` would be  $1 + 2.3 + 0.5 = 3.8$ .

### `\horlines`[*`\TikZ Settings`*]{*`\list`*}

Draws horizontal lines, optionally with the given *`\Settings`*, at the base line of the rows given by *`\list`*. The PGF macro `\foreach`<sup>2</sup> is internally used so the list can include not only row numbers as integer but also fractional numbers and the ‘...’ operator to auto-increment the numbers. Please note that all numbers in the list are multiplied by `\rowdist`. If the list is empty the default ‘1,2,...,\nrows’ is used which draws lines for all rows.

### `\vertlines`[*`\TikZ Settings`*]{*`\list`*}

Like `\horlines` but draws vertical lines and the listed numbers a relative to the basic width. If the list is empty the default ‘0,1,...,\twidth’ is used which draws lines after every period width.

### `\tablerules`[*`\TikZ Settings`*]

This macro adds top and bottom rules to the table in the same (or at least very similar) way as the `booktabs` package is doing it for normal `tabulars`. The current bounding box is used to calculate the needed rule length, which makes this macro position dependent if further code is changing the bounding box.

## Positions & Scalings inside the Table

The first row starts at  $y = 0$  and the next rows are each  $-1 \cdot \text{\code{\rowdist}}$  lower than the previous one. The vertical unit is 1 `signal height` and the default row distance is ‘2’ ( $= 2 \times \text{signal height}$ ). This means that a normal table with three rows goes from  $y = +1$  (base line at  $0 + 1 \text{ signal height}$ ) to  $y = -4$  (first row:  $+0$ , second row:  $-2$ , third row:  $-4$ ). This are relative to the middle of the drawn lines, i.e. the bounding box is  $2 \times \frac{\text{line width}}{2} = 1 \cdot \text{line width}$  higher.

The timing column starts at  $x = 0$  and goes into the positive range while scaled using the period width. Example: `HHHh` has a width of 3.5.

The label column starts at  $x = -\text{\code{\coldist}}$  and the text is right align with the right border at this position.

---

<sup>2</sup>See the `pgf` manual for more details.

## 2.4 Meta-Characters

It is possible to define recurring groups of characters and modifiers as so called meta-characters. These characters are then expanded to the group whenever they appear inside the character list. Please note that like for groups a numeric factor before such a meta-character is taken as a repetition factor not as a width. The meta-character is case sensitive and the other case is not affected by the definition, i.e. the lower- and uppercase versions of one character can have complete different meanings. It is possible to redefine normal characters (only one or both cases) as meta-characters, which suppresses its normal meaning. Using the meta-character in its own definition group causes a infinite loop which will lead to an  $\text{\TeX}$  error.

```
\tikztimingmetachar{<Meta-Character>}{<Character Group>}
```

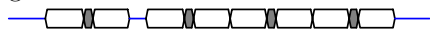
This macro defines the given *<meta-character>* to be identical to the given *<character group>*. Alternatively this can also be done using the TikZ style ‘`timing/metachar={<Meta-Character>}{<Character Group>}`’.

An empty group deletes the meta-character, which might be necessary in cases when normal characters are temporary redefined as meta-characters. However, if the group only contains spaces the meta-character is practically ignored.

*Examples:*

```
\tikztimingmetachar{Y}{2D 0.5U 2D{}} \texttiming{ZZ Y Z 3Y ZZ}
```

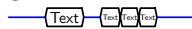
gives:



```
\tikztimingmetachar{Y}{2D{Text}} \tikztimingmetachar{y}{1D{\tiny Text}}
```

```
\texttiming{ZZ Y Z 3y ZZ}
```

gives:



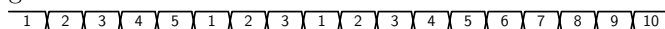
```
\newcounter{mycount}
```

```
\tikztimingmetachar{H}{2D{\stepcounter{mycount}\arabic{mycount}}}
```

```
\tikztimingmetachar{R}{[/utils/exec=\setcounter{mycount}{0}]}
```

```
\texttiming{ 5H R 3H R 10H }
```

gives:



Redefining the glitch ‘G’ character:

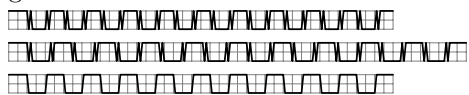
```
\tikztimingmetachar{G}{.1T.1T .2B} \tikztimingmetachar{g}{.1T.1T}
```

```
\texttiming{ 10{H G L G} H } % With correction of width ‘.2B’
```

```
\texttiming{ 10{H g L g} H } % Without correction
```

```
\texttiming{ 10{H L } H } % For comparison
```

gives:



### 3 Styles

The generated logic signals are drawn using the style mechanism provided by `tikz`. This styles are defined and can be redefined using `\tikzset{<style name>/<style=value>,}`. They can also be used in all places where `<TikZ Settings>` is mentioned. Please note that path/draw specific settings might not survive the transition to characters which have there own color, because these start a new drawing path. For a more detailed explanation why this is necessary see the `tikz` manual. However, timing specific settings are saved and restored between internal paths.

The package follows the directory structures approach used by TikZ/PGF and places all styles and other settings under the “subdirectory” ‘`timing`’ in the main “directory” ‘`tikz`’, which is the default when `\tikzset` is used.

Table 4: TikZ Styles and Settings provided and used by this Package.

Style/Setting	Description
<code>timing</code>	Base settings like signal height & period width.
<code>timing/intext</code>	Used for <code>\texttiming</code> . Depends on <code>timing</code> .
<code>timing/picture</code>	Usable for own <code>tikzpictures</code> to set timing settings.
<code>timing/grid</code>	Used for grids. Depends on <code>help lines</code> and <code>timing</code> .
<code>timing/table</code>	Used for <code>{tikztimingtable}</code> . Depends on <code>timing</code> .
<code>timing/table/grid</code>	Used for table grid. Depends on <code>timing/grid</code> .
<code>timing/table/lines</code>	Used for <code>\horlines</code> and <code>\vertlines</code> .
<code>timing/table/rules</code>	Used for <code>\tablerules</code> .
<code>timing/inline node</code>	Used for nodes created by the N character. Defaults to <code>coordinate</code> .
<code>timing/&lt;lowercase char&gt;</code>	Style for character <code>&lt;char&gt;</code> . Not used for ‘H’ and ‘L’.
<code>timing/&lt;lc char&gt;/background</code>	Background style for characters ‘D’ and ‘U’.
<code>timing/&lt;lc char&gt;/text</code>	Text style for character <code>&lt;char&gt;</code> . Only defined for ‘D’.
<code>timing/initchar=&lt;char&gt;</code>	Sets initial character. Only valid as first optional argument in table rows or <code>\texttiming</code> .
<code>timing/metachar=&lt;C&gt;&lt;G&gt;</code>	Sets meta-character <code>&lt;C&gt;</code> to character group <code>&lt;G&gt;</code> .
<code>timing/slope=&lt;0.0–1.0&gt;</code>	Sets slope for logic transitions. This also sets <code>dslope=2*slope</code> , <code>zslope=slope/2</code> .
<code>timing/lslope=&lt;0.0–1.0&gt;</code>	Sets slope for logic transitions only. Default: 0.1
<code>timing/dslope=&lt;0.0–1.0&gt;</code>	Sets slope for data transitions. Default: 0.2
<code>timing/zslope=&lt;0.0–1.0&gt;</code>	Sets slope for Z transitions. Default: 0.05
<code>timing/rowdist=&lt;distance&gt;</code>	Sets (baseline) distance between rows in a <code>tikztimingtable</code> . Default: 2 (=2×signal height)
<code>timing/coldist=&lt;distance&gt;</code>	Sets distance between columns in a <code>tikztimingtable</code> . Default: 1 (=1×period width)



## 4 Examples

This section shows some examples by putting either the full source code or only the needed characters beside the graphical result. Please note that the displayed syntax is the one of `\timing` where the initial character is declared as optional argument (`[<char>]`) *inside/together* with the logic characters. The syntax of `\texttttiming` is identical except the initial character is given as a normal optional argument before the characters argument. All examples except Example 1 are attached in compilable form to this PDF.

Example 1: Initial Characters, Modifiers, TikZ Settings

Characters	Resulting Diagram
HLZXDU TC	
cccc	
tttt	
[c]cccc	
4{c}	
4c4c	
4{1.8c}	
[d] 4{5D{Text}} 0.2D	
3.4H 0.6L	
DDUUUUDD	
DDD{ }DUUDD	
8{2D{\hexcountmacro }} 08 09 0A 0B 0C 0D 0E 0F	
3{2{0.25X 2.5D .25Z}} 0.25X 0.25D 0.25Z	
DDD{ } 3{0.2D{}} DDD	
DDD{ } 3{0.2D{}} 0.4D{ } 0.6D{ } DDD	
HHLLH SSS HLLHHL	
HHGHHGGHLLGLLGH	
ZZ G ZZ G XX G X	
LLL 2{0.1H 0.1L} 0.6H HH	
LLL [timing/slope=0.05] 4{.05H .05L} 0.6H HH	
LLL 0.4U 0.6H HH	
[L] [timing/slope=1.0] HL HL HL HL HL	
LLLLL !{-- +(.5,.5) -- ++(1,0)} HHHHHH	
LLLLL [/utils/exec={\somemacro \code }] HHHHHH	
LL [green] HH [brown] XX LL ZZ [orange] HH	
[ ] [line width=1pt] HLXZDU [line width=0.1pt] HLXZDU	
[ ] [line width=1pt] HLXZDU ; [line width=0.1pt] HLXZDU	

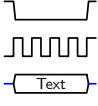
Note: Optional argument must be placed before macro argument if `\texttttiming` is used.

```

\begin{tikztimingtable}
  <<Name>> & hLLLLh      \\
  Clock    & 10{c}       \\
  Signal   & z4D{Text}z     \\
\end{tikztimingtable}

```

«Name»



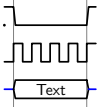
Example 2: `{tikztimingtable}` without `\extracode`.

```

\begin{tikztimingtable}
  <<Name>> & hLLLLh      \\
  Clock    & 10{c}       \\
  Signal   & z4D{Text}z     \\
\extracode
  \draw (0,0) circle (0.2pt); % Origin
  \begin{pgfonlayer}{background}
    \vertlines[help lines]{0.5,4.5}
  \end{pgfonlayer}
\end{tikztimingtable}

```

«Name»

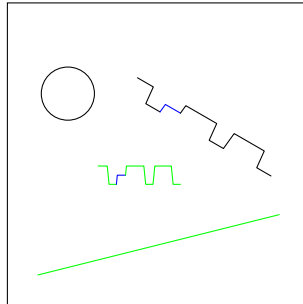


Example 3: `{tikztimingtable}` with `\extracode`.

```

\begin{tikzpicture}[x=4cm,y=4cm]
  \draw (0,0) rectangle (1,1);
  \draw (0.2,0.7) circle (10pt);
  \begin{scope}[green]
    \draw (0.1,0.1) -- +(0.8,0.2);
    \timing at (0.3,0.4) {hlzhhlhhl};
  \end{scope}
  \timing [rotate=-30]
    at (0.4,0.7) {HLZHHLHHL};
\end{tikzpicture}

```

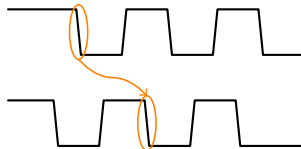


Example 4: `\timing` inside general `{tikzpicture}`.

```

\Huge
\begin{tikzpicture}[timing,thick,
  timing/inline node/.style={coordinate,
  shift={(0.05,-.5)}}]
\timing at (0,2) {hH N(A) LHLHL};
\timing at (0,0) {HLH N(B) LHLl};
\draw [orange,semithick]
  (A) ellipse (.2 and .6) +(0,-0.6) coordinate (Ax)
  (B) ellipse (.2 and .6) +(0,+0.6) coordinate (Bx);
\draw [orange,semithick,->]
  (Ax) parabola[bend pos=0.5] (Bx);
\end{tikzpicture}

```

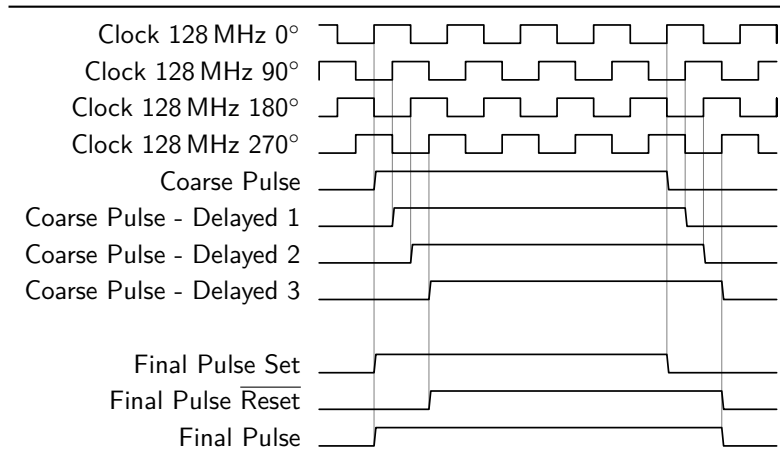


Example 5: Using In-Line Nodes to draw Relationships.

```

\def\degr{${}\^{\circ}}
\begin{tikztimingtable}
  Clock 128\,MHz 0\degr      & H 2C N(A1) 8{2C} N(A5) 3{2C} G\\
  Clock 128\,MHz 90\degr     & [C] 2{2C} N(A2) 8{2C} N(A6) 2{2C} C\\
  Clock 128\,MHz 180\degr    & C 2{2C} N(A3) 8{2C} N(A7) 2{2C} G\\
  Clock 128\,MHz 270\degr    & 3{2C} N(A4) 8{2C} N(A8) 2C C\\
  Coarse Pulse               & 3L 16H 6L \\
  Coarse Pulse - Delayed 1   & 4L N(B2) 16H N(B6) 5L \\
  Coarse Pulse - Delayed 2   & 5L N(B3) 16H N(B7) 4L \\
  Coarse Pulse - Delayed 3   & 6L 16H 3L \\
  \\
  Final Pulse Set            & 3L 16H N(B5) 6L \\
  Final Pulse $\overline{\mbox{Reset}}$ & 6L N(B4) 16H 3L \\
  Final Pulse                & 3L N(B1) 19H N(B8) 3L \\
\extracode
\ablerules
\begin{pgfonlayer}{background}
  \foreach \n in {1,...,8}
    \draw [help lines] (A\n) -- (B\n);
\end{pgfonlayer}
\end{tikztimingtable}

```

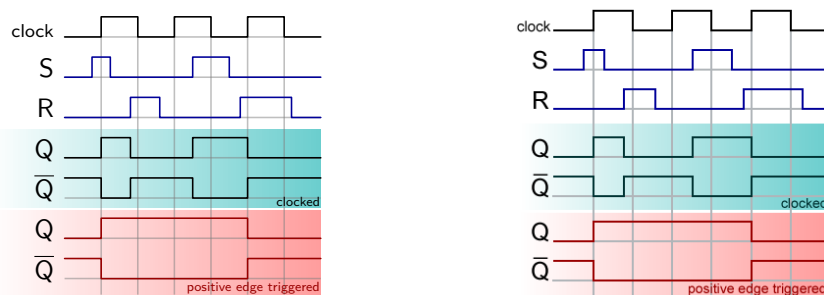


Example 6: Using In-Line Nodes to draw Marker Lines.

```

\definecolor{bgblue}{rgb}{0.41961,0.80784,0.80784}
\definecolor{bgred}{rgb}{1,0.61569,0.61569}
\definecolor{fgblue}{rgb}{0,0,0.6}
\definecolor{fgred}{rgb}{0.6,0,0}
\begin{tikztimingtable}[timing/slope=0,
  timing/coldist=2pt,xscale=2,yscale=1.1,semithick]
  \scriptsize clock & 7{C}\
  S & .75L h 2.25L H LL [fgblue]\
  R & 1.8L .8H 2.2L 1.4H 0.8L [fgblue]\
  Q & L .8H 1.7L 1.5H LL\
  $\overline{\mbox{Q}}$ & H .8L 1.7H 1.5L HH\
  Q & LHHHLL [fgred]\
  $\overline{\mbox{Q}}$ & HLLLLH [fgred]\
\extracode
\begin{pgfonlayer}{background}
  \shade [right color=bgblue,left color=white]
    (7,-8.45) rectangle (-2,-4.6);
  \shade [right color=bgred,left color=white]
    (7,-12.8) rectangle (-2,-8.6);
\begin{scope}[gray,semitransparent,semithick]
  \horlines{}
  \foreach \x in {1,...,6}
    \draw (\x,1) -- (\x,-12.8);
  % similar: \vertlines{1,...,6}
\end{scope}
\node [anchor=south east,inner sep=0pt]
  at (7,-8.45) {\tiny clocked};
\node [anchor=south east,inner sep=0pt,fgred]
  at (7,-12.8) {\tiny positive edge triggered};
\end{pgfonlayer}
\end{tikztimingtable}
\insertoriginalimageforcomparisionifpresent

```

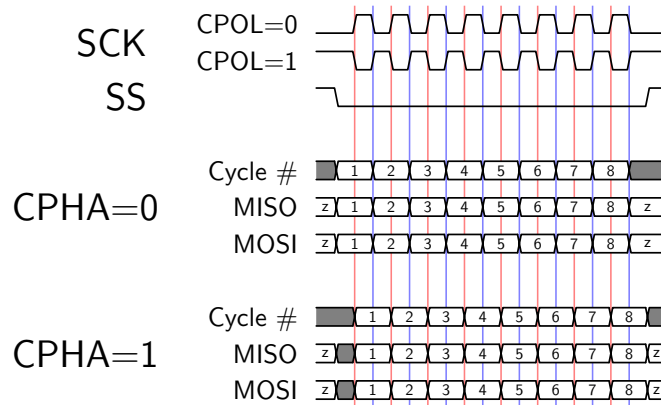


Example 7: SR flip-flop timing diagram (left). Redrawn from image (right)  
[http://commons.wikimedia.org/wiki/File:SR\\_FF\\_timing\\_diagram.png](http://commons.wikimedia.org/wiki/File:SR_FF_timing_diagram.png)

```

\newcounter{countup}
\newcommand*\{countup}\{\addtocounter{countup}{1}\thecountup}
\newcommand*\{crst}\{\setcounter{countup}{0}}
\begin{tikztimingtable}
[timing/d/background/.style={fill=white},
 timing/lslope=0.2]
CPOL=0 & LL 15{T} LL \\\
CPOL=1 & HH 15{T} HH \\\
      & & H 17L H \\\
\\
\crst Cycle \# & U      8{2D{\countup}} 2U \\\
\crst MISO & D{z} 8{2D{\countup}} 2D{z} \\\
\crst MOSI & D{z} 8{2D{\countup}} 2D{z} \\\
\\
\crst Cycle \# & UU      8{2D{\countup}} U \\\
\crst MISO & D{z}U 8{2D{\countup}} D{z} \\\
\crst MOSI & D{z}U 8{2D{\countup}} D{z} \\\
\extracode
\begin{pgfonlayer}{background}
\begin{scope}[semitransparent,semithick]
\vertlines[red]{2.1,4.1,...,17.1}
\vertlines[blue]{3.1,5.1,...,17.1}
\end{scope}
\end{pgfonlayer}
\begin{scope}
[font=\sffamily\Large,shift={(-6em,-0.5)},anchor=east]
\node at ( 0, 0) {SCK}; \node at ( 0,-3 ) {SS};
\node at (1ex,-9) {CPHA=0}; \node at (1ex,-17) {CPHA=1};
\end{scope}
\end{tikztimingtable}%

```



Example 8: SPI Interface Timing. Redrawn from image  
[http://en.wikipedia.org/wiki/File:SPI\\_timing\\_diagram.svg](http://en.wikipedia.org/wiki/File:SPI_timing_diagram.svg)

## 5 Implementation

### 5.1 Package Header

```
1 \langle*package\rangle
2 \RequirePackage{tikz}
3 \usetikzlibrary{calc}
4 \usetikzlibrary{backgrounds}
5 \usetikzlibrary{decorations.pathmorphing}
6 \ifx\collect@body\undefined
7   \IfFileExists{environ.sty}
8     {\RequirePackage{environ}}
9     {\RequirePackage{amsmath}}
10 \fi
11
12 \def\tikztimingwidth{0.0}
13 \newcount\tikztiming@numint
14 \newcount\tikztiming@numfrac
15 \def\tikztiming@num{1.0}%
16 \let\tikztiming@back\empty
17 \newlength\tikztiming@xunit
18 \newlength\tikztiming@yunit
19
20 \newcounter{tikztiming@nrows}%
21 \def\tikztiming@rowdist{2}%
22 \def\tikztiming@colldist{1}%
23 \def\tikztiminglabel#1{#1}%
24
25 \def\tikztiming@prefix{tikztiming@trans@}
```

### 5.2 TikZ Style Settings

```
26 \tikzset{timing/.style={%
27   x=1.6ex, y=1.6ex,
28   line cap=round, line join=round,
29   /utils/exec={\setlength{\tikztiming@xunit}{1.6ex}\setlength{\tikztiming@yunit}{1.6ex}},
30 }%
31 }
32 \tikzset{%
33   timing/.cd,
34   initchar/.value required,
35   initchar/.code={\ifx\lastchar\empty\uppercase{\def\lastchar{#1}}\fi},
36   metachar/.code 2 args={\tikztimingmetachar{#1}{#2}},
37   grid/.style={timing,help lines},
38   picture/.style={timing,line width=0.15ex},
39   intext/.style={timing,line width=0.15ex},
40   inline node/.style={shape=coordinate},
41   table/.style={timing,line width=0.15ex,font=\sffamily},
42   coord/.style={inner sep=0pt,outer sep=0pt},
43   save/.style={inner sep=0pt,outer sep=0pt,/utils/exec=\tikztiming@savesettings},
44   restore/.style={/utils/exec=\tikztiming@restoresettings},
```

```

45 name/.style={inner sep=0pt,outer sep=0pt},
46 d/text/.style={timing,scale=0.6,font=\sffamily},
47 d/background/.style={},
48 h/.style={},
49 l/.style={},
50 d/.style={},
51 m/.style={black!40!brown},
52 m/decoration/.style={decorate,decoration={zigzag,segment
53 length=.25\tikztiming@xunit,amplitude=.225\tikztiming@yunit}},
54 k/.style={black!40!blue,semitransparent},
55 u/background/.style={fill=gray},
56 u/.style={},
57 o/background/.style={},
58 o/.style={timing/d,line width=0.10ex,dotted},
59 g/.style={},
60 z/.style={blue},
61 t/.style={},
62 c/.style={timing/slope=0.0},
63 x/.style={red},
64 table/grid/.style={timing/grid},
65 table/lines/.style={},
66 table/rules/.style={line width=0.08em,line cap=butt},
67 slope/.code={%
68   \tikztimingsetslope{#1}%
69   \tikztimingsetdslope{2*#1}%
70   \tikztimingsetzslope{#1/2}%
71 },
72 lslope/.code={\tikztimingsetslope{#1}},
73 dslope/.code={\tikztimingsetdslope{#1}},
74 zslope/.code={\tikztimingsetzslope{#1}},
75 coldist/.store in=\tikztiming@coldist,
76 rowdist/.store in=\tikztiming@rowdist,
77 }

```

### 5.3 Macros

#### `\texttimingbefore`

This macro is executed inside the `tikzpicture` environment of `\texttiming` before the timing diagram is drawn.

```
78 \def\texttimingbefore{}
```

#### `\texttimingafter`

This macro is executed inside the `tikzpicture` environment of `\texttiming` after the timing diagram is drawn.

```
79 \def\texttimingafter{}
```

#### `\texttiminggrid`

Draws a background grid with the ‘timing/grid’ setting. Should be used inside `\texttimingbefore`.



```

80 \def\texttiminggrid{%
81   \draw[xstep={\timingwidth/2.},ystep={\timingheight/2.},timing/grid] (0,0) grid
82   (\timingwidth*\tikztimingwidth,\timingheight);
83 }

```

## `\texttiming`

```

84 \DeclareRobustCommand*\texttiming[2] [] {%
85   \begingroup
86     \tikztiming@init
87     \ifx\relax#1\relax\else
88       \tikztiming@testoptarg#1\relax\relax%
89     \fi
90     \def\@tempa{\begin{tikzpicture}[timing/intext,}%][
91     \expandafter\@tempa\settings]
92     \@ifundefined{tikztiming@initcode@\lastchar}%
93     {}%
94     {\@nameuse{tikztiming@initcode@\lastchar}}%
95     \ifx\lastchar\empty\else
96       \@ifundefined{tikztiming@prefix\lastchar @start}%
97       {\PackageWarning{tikz-timing}{Start value for timing character '\lastchar'
98       is not defined and will be ignored!}}{}{}%
99       {\tikztiming@nameaddtostr{\lastchar @start}{}%
100     \fi
101     \tikztiming@#2\relax
102     \%message{^^J\meaning\tikztiming@str^^J}%
103     \path[use as bounding box] (0,0) rectangle
104     (\timingwidth*\tikztimingwidth,\timingheight);%
105     \texttimingbefore
106     \tikztiming@str;%
107     \node [shape=tikztiming@shape,anchor=origin,name=last texttiming] at (0,0)
108     {};
109     \texttimingafter
110     \end{tikzpicture}%
111   \endgroup
112 }

```

## `\tikztiming@testoptarg`

```

113 \def\tikztiming@testoptarg#1#2\relax{%
114   \ifx\relax#2\relax
115     \uppercase{\def\lastchar{#1}}%
116   \else
117     \def\settings{#1#2}%
118   \fi
119 }

```

## `\tikztiming@init`

```

120 \def\tikztiming@init{%
121     \let\lastchar\empty%
122     \let\currentchar\empty
123     \let\settings\empty
124     \def\tikztimingwidth{0.0}%
125     \setcounter{tikztimingtrans}{-1}%
126     \def\tikztiming@str{\draw (0,0) coordinate (timing/start base) }%
127 }

```

### **\timing**

```

128 \def\timing{%
129     \@ifnextchar{[]}%
130     {\timing@}%
131     {\timing@[]}%
132 }

```

### **\timing@**

```

133 \def\timing@[#1]{%
134     \@ifnextchar{a}%
135     {\timing@at{#1}}%
136     {\PackageWarning{tikz-timing}{The \string\timing\space command awaits now an
137     'at' before the coordinate, like the \string\node\space command is doing.
138     Please update your source code to ensure compatibility with newer
139     versions}{-}{-}}%
140     \@ifnextchar{+}%
141     {\timing@@{#1}}%
142     {\@ifnextchar(%
143     {\timing@@{#1}}%
144     {\timing@@{#1}++(0,0)}}%
145     }%
146     }%
147 }

```

### **\timing@at**

```

148 \def\timing@at#1at#2(#3){%
149     \timing@@@{#1}{#2(#3)}%
150 }

```

### **\timing@@**

```

151 \def\timing@@#1#2(#3){%
152     \timing@@@{#1}{#2(#3)}%
153 }

```

`\timing@@@`

```
154 \def\timing@@@#1#2#3{%
155   \begingroup
156     \tikztiming@init
157     \let\tikz@alias=\pgfutil@empty%
158     \begin{scope}[shift={#2},timing,#1]%
159       \ifnextchar{[%
160         {\timing@@@init}%
161         {\timing@@@init[]}%
162         #3\relax
163         %\message{^^J\meaning\tikztiming@str^^J}%
164         \tikztiming@str;%
165         \node [shape=tikztiming@shape,anchor=origin,#1] at (0,0) {};
166       \end{scope}%
167     \endgroup
168   \timing@@@end
169 }
```

`\timing@@@end`

```
170 \def\timing@@@end#1;%
171   \ifx;#1;\else
172     \PackageError{tikz-package}{Can not parse timing path}{-}{-}{-}%
173   \fi
174 }
```

`\timing@@@init`

```
175 \def\timing@@@init[#1]{%
176   \ifx\relax#1\relax\else
177     \uppercase{\def\lastchar{#1}}%
178   \fi
179   \ifundefined{tikztiming@initcode@\lastchar}%
180     {}%
181     {\@nameuse{tikztiming@initcode@\lastchar}}%
182   \ifx\lastchar\empty\else
183     \ifundefined{tikztiming@prefix\lastchar @start}%
184       {\PackageWarning{tikz-timing}{Start value for timing character '\lastchar'
185         is not defined and will be ignored!}}{-}{-}{-}%
186       {\tikztiming@nameaddtostr{\lastchar @start}}{-}{-}%
187     \fi
188   \tikztiming@
189 }
```

#### `\tikztiming@trans@`

The empty transition gets defined to avoid errors if it is used by the generic code, e.g. if a non-combinable character like ‘C’ is the last one.

```
190 \let\tikztiming@trans@\@gobble
```

#### `\tikztiming@aftercode@T`

```
191 \def\tikztiming@aftercode@T{%
192   \tikztiming@output@flush
193 }
```

#### `\tikztiming@aftercode@t`

```
194 \def\tikztiming@aftercode@t{%
195   \tikztiming@aftercode@T
196 }
```

#### `\tikztiming@aftercode@C`

```
197 \def\tikztiming@aftercode@C{%
198   %\tikztiming@output@flush
199 }
```

#### `\tikztiming@aftercode@c`

```
200 \def\tikztiming@aftercode@c{%
201   \tikztiming@aftercode@C
202 }
```

#### `\tikztiming@aftercode@G`

```
203 \def\tikztiming@aftercode@G{%
204   \let\lastchar\secondlastchar
205   \let\tikztimingwidth\lasttikztimingwidth
206 }
```

#### `\tikztiming@aftercode@g`

```
207 \def\tikztiming@aftercode@g{%
208   \let\lastchar\secondlastchar
209   \let\tikztimingwidth\lasttikztimingwidth
210 }
```

#### `\tikztiming@aftercode@S`

```
211 \def\tikztiming@aftercode@S{%
212   \let\lastchar\secondlastchar
213 }
```

#### `\tikztiming@aftercode@s`

```
214 \def\tikztiming@aftercode@s{%
215   \let\lastchar\secondlastchar
216 }
```

`\tikztiming@beforenextcode@D@edge@`

```
217 \def\tikztiming@beforenextcode@D@edge@{%
218   \if D\currentchar\else
219     \if d\currentchar\else
220       \def\lastchar{D}%
221     \fi
222   \fi
223 }
```

`\tikztiming@beforecode@d@edge@`

```
224 \def\tikztiming@beforenextcode@d@edge@{%
225   \if D\currentchar\else
226     \if d\currentchar\else
227       \def\lastchar{D}%
228     \fi
229   \fi
230 }
```

`\tikztiming@initcode@D`

```
231 \def\tikztiming@initcode@D{%
232   \def\lastchar{D@edge@}%
233 }
```

`\tikztiming@initcode@d`

```
234 \def\tikztiming@initcode@d{%
235   \def\lastchar{d@edge@}%
236 }
```

`\tikztiming@`

The `\@ifnextchar\bgroup` is a trick to remove following spaces which would break the number test.

```
237 \def\tikztiming@{%
238   \@ifnextchar\bgroup
239     {\tikztiming@testfornum}%
240     {\tikztiming@testfornum}%
241 }
```

`\tikztiming@eaddtostr`

```
242 \def\tikztiming@eaddtostr#1{%
243   \begingroup
244     \tikztiming@internaldefs{}%
245     \@temptokena\expandafter{\tikztiming@str}%
246     \xdef\tikztiming@str{%
247       \the\@temptokena
248       #1%
249     }%
```

```

250 \endgroup
251 }

\tikztiming@addtostr
252 \def\tikztiming@addtostr{%
253 \g@addto@macro\tikztiming@str
254 }

\tikztiming@output

255 \def\tikztiming@output#1#2{%
256 \ifx\relax#2\relax
257 \tikztiming@nameaddtostr{#1}%
258 \else
259 \ifcase0%
260 \ifx\tikztiming@output@bufchara\empty
261 \ifx\tikztiming@output@bufcharb\empty
262 1%
263 \fi
264 \fi\relax
265 % not empty
266 \edef\tikztiming@output@currentchar{#2}%
267 \ifcase0%
268 \expandafter\ifx\csize tikztiming@nocombine@#2\endcsize\relax
269 \ifx\tikztiming@output@currentchar\tikztiming@output@bufcharb
270 1%
271 \fi\fi
272 \relax
273 \tikztiming@output@flush
274 \edef\tikztiming@output@bufchara{#1}%
275 \edef\tikztiming@output@bufcharb{#2}%
276 \or
277 \pgfmathparse{\tikztiming@output@bufnum + \tikztiming@num}%
278 \let\tikztiming@output@bufnum\pgfmathresult
279 \def\tikztiming@num{1.0}%
280 \fi
281 \else % empty
282 \edef\tikztiming@output@bufchara{#1}%
283 \edef\tikztiming@output@bufcharb{#2}%
284 \let\tikztiming@output@bufnum\tikztiming@num
285 \def\tikztiming@num{1.0}%
286 \fi
287 \fi
288 }

Init buffer macros:
289 \def\tikztiming@output@bufchara{}%
290 \def\tikztiming@output@bufcharb{}%
291 \def\tikztiming@output@bufnum{0}%

```

### `\tikztiming@output@flush`

```
292 \def\tikztiming@output@flush{%
293   \begingroup
294     \let\tikztiming@num\tikztiming@output@bufnum
295     \tikztiming@nameaddtostr{%
296       \tikztiming@output@bufchara
297       \tikztiming@output@bufcharb
298     }%
299   \endgroup%
300   \gdef\tikztiming@output@bufchara{}%
301   \gdef\tikztiming@output@bufcharb{}%
302   \global\let\tikztiming@output@bufnum\tikztiming@num
303   \gdef\tikztiming@num{1.0}%
304 }
```

### `\tikztiming@nameaddtostr`

```
305 \def\tikztiming@nameaddtostr#1{%
306   \begingroup
307     \edef\@tempa{\tikztiming@num}%
308     \expandafter\g@addto@macro
309     \expandafter\tikztiming@str
310     \expandafter{\csname\tikztiming@prefix#1\expandafter\endcsname
311     \expandafter{\@tempa} }%
312   \endgroup
313   \def\tikztiming@num{1.0}%
314 }
```

### `\tikztiming@nameedef`

Defines internal tikztiming macro with name `\<prefix>\<name (#2)>`. The macro definition (`#3`) is expanded while the internal drawing definitions are active.

```
315 \newcommand\tikztiming@nameedef[3][A]{%
316   \def\@gtempa##1{#3}%
317   \expandafter\let\csname\tikztiming@prefix#2\general\endcsname\@gtempa
318   \begingroup
319     \tikztiming@internaldefs{#1}%
320     \xdef\@gtempa##1{\@gtempa{\width}}%
321   \endgroup
322   \expandafter\let\csname\tikztiming@prefix#2\endcsname\@gtempa
323   \let\@gtempa\empty
324 }
```

### `\tikztiming@namelet`

Only execute `\let` if the original macro is defined or the destination macro is defined and would now set to undefined.

```

325 \newcommand\tikztiming@namelet[2]{%
326   \ifcase0%
327     \@ifundefined{\tikztiming@prefix#2}%
328       {\@ifundefined{\tikztiming@prefix#1}%
329         {0}{1}%
330       }%
331       {1}%
332     \relax
333   \else
334     \expandafter\let
335     \csname\tikztiming@prefix#1\expandafter\endcsname
336     \csname\tikztiming@prefix#2\endcsname
337   \fi
338 }
```

`\tikztiming@@end`

```

339 \def\tikztiming@@end{%
340   \tikztiming@output@flush
341   \global\let\tikztimingwidth\tikztimingwidth
342   \tikztiming@addtostr{ coordinate (timing/end)
343     let \p1 = (timing/start base), \p2 = (timing/end) in
344     coordinate (timing/end base) at (\x2,\y1)
345     coordinate (timing/end top) at (\x2,1+\y1)
346   }%
347 }
```

`\tikztiming@@`

```

348 \def\tikztiming@@#1{%
349   \ifx\relax#1\empty
350     \expandafter\tikztiming@@end
351   \else
352     \let\lasttikztimingwidth\tikztimingwidth
353     \tikztiming@iflower{#1}%
354     {\pgfmathparse{\tikztiming@num/2.0}\let\tikztiming@num\pgfmathresult}%
355     {}%
356   \ifx\tikztiming@back\empty\else
357     \pgfmathparse{\tikztiming@num-\tikztiming@back}%
358     \let\tikztiming@num\pgfmathresult
359     \let\tikztiming@back\empty
360   \fi
361   \pgfmathparse{\tikztimingwidth + \tikztiming@num}%
362   \let\tikztimingwidth\pgfmathresult
363   \def\currentchar{#1}%
364   \uppercase{\def\currentcharuc{#1}}%
```



```

365 \ifundefined{tikztiming@beforenextcode@lastchar}%
366 {}%
367 {\@nameuse{tikztiming@beforenextcode@lastchar}}%
368 \ifundefined{tikztiming@beforecode@currentchar}%
369 {}%
370 {\@nameuse{tikztiming@beforecode@currentchar}}%
371 \ifundefined{tikztiming@prefixlastcharcurrentchar}%
372 {\ifundefined{tikztiming@prefixlastcharcurrentcharuc}%
373 {\PackageWarning{tikz-timing}{Timing transition 'lastcharcurrentchar'
374 is not defined and will be ignored!}}{}{}}%
375 {\tikztiming@output{lastchar}{currentcharuc}}%
376 }%
377 {\tikztiming@output{lastchar}{currentchar}}%
378 \letsecondlastcharlastchar
379 \letlastcharcurrentcharuc
380 \ifundefined{tikztiming@aftercode@currentcharuc}%
381 {}%
382 {\@nameuse{tikztiming@aftercode@currentcharuc}}%
383 \expandafter
384 \tikztiming@testfortext
385 \fi
386 }

```

#### **\tikztiming@testfortext**

```

387 \def\tikztiming@testfortext{%
388 \ifnextchar\bgroup
389 {\tikztiming@handletext}%
390 {\tikztiming@}%
391 }

```

#### **\tikztiming@handletext**

```

392 \def\tikztiming@handletext#1{%
393 \ifnextchar{[%
394 {\tikztiming@handletext@}%
395 {\tikztiming@handletext@[[]}%
396 #1\relax
397 }

```

#### **\tikztiming@handletext@**

```

398 \def\tikztiming@handletext@[#1]#2\relax{%
399 \begingroup
400 \expandafter\lowercase\expandafter{%
401 \expandafter\def\expandaftercurrentcharlc
402 \expandafter{currentchar}%
403 }%

```

```

404 \pgfkeysifdefined{/tikz/timing/\currentcharlc/text/.@cmd}%
405 {%
406 \tikztiming@output@flush
407 \tikztiming@eaddtostr{%
408   node (timing@dend) at +(\dslope/2.0,\height/2.0) {}
409   node [%]
410     shift={($ (timing@dstart)!0.5!(timing@dend) $)},%
411     timing/\currentcharlc/text,%
412 }%
413 \endgroup
414 \tikztiming@addtostr{[%
415   #1%
416   ] {#2}%
417 }%
418 \def\lastchar{D@edge@}%
419 {%
420   \endgroup
421   \PackageWarning{tikz-timing}{Ignoring text for character
422     '\currentchar'!}{-}{-}{-}%
423 }%
424 \tikztiming@
425 }

```

**\tikztiming@defcode**

```

426 \def\tikztiming@defcode#1{%
427   \@namedef{tikztiming@code@meaning#1}%
428 }

```

**\tikztiming@defcode,**

```

429 \tikztiming@defcode{,}{%
430   \tikztiming@output@flush
431   \tikztiming@eaddtostr{%
432     \newdraw
433   }%
434   \tikztiming@
435 }

```

**\tikztiming@defcode;**

```

436 \tikztiming@defcode{;}{%
437   \tikztiming@output@flush
438   \tikztiming@eaddtostr{%
439     \newdrawns
440   }%
441   \tikztiming@
442 }

```

**\tikztiming@defcodeN**

```

443 \tikztiming@defcode{N}{%
444   \@ifnextchar[%
445     {\tikztiming@addnode@getoptions}%
446     {\tikztiming@addnode@getoptions[]}%
447 }

```

#### `\tikztiming@addnode`

```

448 \def\tikztiming@addnode@getoptions[#1]{%
449   \@ifnextchar(%
450     {\tikztiming@addnode@getname{#1}}%
451     {\tikztiming@addnode@getname{#1}()})%
452 }

```

#### `\tikztiming@addnode@getname`

```

453 \def\tikztiming@addnode@getname#1(#2){%
454   \@ifnextchar\bgroup
455     {\tikztiming@addnode@{#1}{#2}}%
456     {\tikztiming@addnode@{#1}{#2}{}}%
457 }

```

#### `\tikztiming@addnode@`

```

458 \def\tikztiming@addnode@#1#2#3{%
459   \tikztiming@output@flush
460   \begingroup
461     \def\@tempa{#2}%
462     \ifx\@tempa\empty
463       \def\@tempa{ node [timing/inline node,#1] }%
464     \else
465       \def\@tempa{ node [timing/inline node,#1] (#2) }%
466     \fi
467     \expandafter\tikztiming@addtostr\expandafter{\@tempa {#3} }%
468   \endgroup
469   \tikztiming@
470 }

```

#### `\tikztiming@testforcode`

```

471 \def\tikztiming@testforcode{%
472   \@ifnextchar{!}%
473     {\tikztiming@testforcode@}%
474     {\@ifundefined{tikztiming@code@meaning\@let@token}%
475       {\tikztiming@@}%

```

```

476      {\csname tikztiming@code@\meaning\@let@token\expandafter
477       \endcsname\@gobble}%
478    }%
479 }

```

#### `\tikztiming@testforcode@`

```

480 \def\tikztiming@testforcode@#1{%
481   \@ifnextchar\bgroup
482   {\tikztiming@handlecode}%
483   {%
484     \PackageWarning{tikz-timing}{Missing braces after '!' character. Ignoring
485     this character}{-}{-}{-}%
486     \tikztiming@
487   }%
488 }

```

#### `\tikztiming@defcodeB`

```

489 \tikztiming@defcode{B}{%
490   \pgfmathparse{\tikztiming@back+\tikztiming@num}%
491   \let\tikztiming@back\pgfmathresult
492   \tikztiming@
493 }

```

#### `\tikztiming@handlecode`

```

494 \def\tikztiming@handlecode#1{%
495   \tikztiming@output@flush
496   \tikztiming@addtostr{ #1 }%
497   \tikztiming@
498 }

```

#### `\tikztiming@defcode []`

```

499 \tikztiming@defcode[#1]{%
500   \tikztiming@addtostr{ [#1] }%
501   \tikztiming@
502 }

```

#### `\tikztiming@testfornum`

```

503 \def\tikztiming@testfornum{%
504   \let\tikztiming@numchars\empty
505   \tikztiming@numfrac0\relax
506   \afterassignment
507   \tikztiming@testfornum@
508   \tikztiming@numint0%
509 }

```

**\tikztiming@testfornumfrac**

```
510 \def\tikztiming@testfornumfrac{%
511   \afterassignment
512   \tikztiming@testfornum@@@
513   \tikztiming@numfrac1%
514 }
```

**\tikztiming@numloop**

```
515 \def\tikztiming@numloop{%
516   \ifnum\tikztiming@numint>0%
517     \toks@\expandafter{\tikztiming@numchars}%
518     \xdef\tikztiming@numchars{%
519       \the\toks@
520       \the\@temptokena
521     }%
522     \advance\tikztiming@numint by -1\relax
523     \expandafter\tikztiming@numloop
524   \fi
525 }
```

**\tikztiming@testfornum@**

```
526 \def\tikztiming@testfornum@{%
527   \ifnum0<\tikztiming@numint
528     \let\tikztiming@next\tikztiming@testfornum@@
529   \else
530     \def\tikztiming@next{%
531       \@ifnextchar{.}%
532       {\expandafter\tikztiming@testfornumfrac\@gobble}%
533       {%
534         \tikztiming@numint1\relax
535         \tikztiming@numfrac0\relax
536         \def\tikztiming@num{1.0}%
537         \if\tikztiming@metachar
538           {\def\@tempa{\expandafter\expandafter\expandafter
539             \tikztiming@testfornum@@@
540             \expandafter\expandafter\expandafter{%
541               \csname tikztiming@metachar@\meaning\@let@token\endcsname}}%
542             \expandafter\@tempa\@gobble
543           }%
544           {\tikztiming@testforcode}%
545         }%
546       }%
547     \fi
548     \tikztiming@next
549 }
```

**\tikztiming@testfornum@@**

```
550 \def\tikztiming@testfornum@@{%
```

```

551 \@ifnextchar{.}%
552 {\expandafter\tikztiming@testfornumfrac\@gobble}%
553 {\tikztiming@testfornum@@@}%
554 }

```

#### **\tikztiming@testfornum@@@**

```

555 \def\tikztiming@testfornum@@@{%
556 \xdef\tikztiming@num{\the\tikztiming@numint.\expandafter\@gobble\the\tikztiming@numfrac}%
557 \ifnextchar\bgroup
558 {%
559 \expandafter\tikztiming@numfrac\expandafter0\expandafter
560 \@gobble\the\tikztiming@numfrac\relax
561 \ifnum0=\tikztiming@numfrac\else
562 \pgfmathparse{round(\tikztiming@num)}%
563 \PackageWarning{tikz-timing}%
564 {Can not repeat group by a non-integer factor!^^J%
565 Rounding '\tikztiming@num' to '\pgfmathresult'.}{-}{-}%
566 \let\tikztiming@num\pgfmathresult
567 \fi
568 \tikztiming@testfornum@@@
569 }%
570 {%
571 \if\tikztiming@metachar
572 {\def\@tempa{\expandafter\expandafter\expandafter
573 \tikztiming@testfornum@@@
574 \expandafter\expandafter\expandafter{%
575 \csname tikztiming@metachar@\meaning\@let@token\endcsname}}%
576 \expandafter\@tempa\@gobble
577 }%
578 {\tikztiming@testforcode}%
579 }%
580 }

```

#### **\tikztiming@metachar**

```

581 \def\tikztimingmetachar#1#2{%
582 \ifx\relax#2\relax
583 \expandafter\let\csname tikztiming@metachar@\meaning#1\endcsname\undefined
584 \else
585 \@namedef{tikztiming@metachar@\meaning#1}{#2}%
586 \fi
587 }

```

#### **\if\tikztiming@metachar**

```

588 \def\if\tikztiming@metachar#1#2{%

```

```

589 \@ifundefined{tikztiming@metachar@meaning\@let@token}{#2}{#1}%
590 }

```

**\tikztiming@testfornum@@@**

```

591 \def\tikztiming@testfornum@@@#1{%
592   \begingroup
593     \@temptokena{#1}%
594     \tikztiming@numloop%
595   \endgroup
596   \tikztiming@numint1\relax
597   \tikztiming@numfrac0\relax
598   \expandafter\tikztiming@\tikztiming@numchars
599 }

```

## 5.4 Table environment

```

600 %\usetikzlibrary{backgrounds}
601 \newcounter{tikztimingrows}

```

**\tikztiming@extracode**

```

602 \def\tikztiming@extracode{\@gobble{EXTRACODE}}%

```

**tikztimingtable**

```

603 \newenvironment{tikztimingtable}[1][1]{%
604   \begingroup
605   \setcounter{tikztiming@nrows}{0}%
606   \def\tikztiming@maxwidth{0.0}%
607   \let\extracode\tikztiming@extracode
608   \let\tablegrid\tikztiming@tablegrid
609   \let\fulltablegrid\tikztiming@fulltablegrid
610   \let\horlines\tikztiming@horlines
611   \let\vertlines\tikztiming@vertlines
612   \let\ablerules\tikztiming@ablerules
613   \def\rowdist{\tikztiming@rowdist}%
614   \def\coldist{\tikztiming@coldist}%
615   \def\nrows{\the\c@tikztiming@nrows}%
616   \def\twidth{\tikztiming@maxwidth}%
617   \tikzpicture[timing/table,#1]%
618     \coordinate (last row) at (0,\rowdist);
619     \coordinate (label pos) at (-1*\coldist,0);
620     \coordinate (timing/table/top left) at (0,1);
621     \coordinate (timing/table/bottom right) at (0,0);
622     \collect@body\tikztiming@table
623 }{%
624 }

```

**\tikztiming@table**

```
625 \def\tikztiming@table#1{%
626   \tikztimingtable@row#1\endtikztimingtable@
627   \endtikzpicture
628   \endgroup
629 }
```

**\endtikztimingtable@**

```
630 \def\endtikztimingtable@{\@gobble{ENDTIKSTIMING}}
```

**\tikztimingextracode**

```
631 \long\def\tikztimingextracode#1#2\endtikztimingtable@{%
632   \path let
633     \p1 = (timing/table/top left),
634     \p2 = (timing/table/bottom right)
635   in
636     coordinate (timing/table/bottom left) at (\x1,\y2)
637     coordinate (timing/table/top right) at (\x2,\y1)
638     coordinate (timing/table/size) at (\x2-\x1,\y1-\y2)
639   ;
640   #2%
641 }
```

**\tikztiming@emptycell**

Just used as marker. Needs unique definition.

```
642 \def\tikztiming@emptycell{%
643   \@gobble{tikztiming@emptycell}%
644 }
```

**\tikztimingtable@row**

```
645 \def\tikztimingtable@row#1\\{%
646   \tikztimingtable@row@#1&\tikztiming@emptycell&\\
647 }
```

**\tikztimingtable@row@**

```
648 \def\tikztimingtable@row@#1&#2&#3\\{%
649   \ifx\\#3\\ \else
650     \begingroup
651       \def\@tempa{\tikztiming@emptycell&}%
652       \def\@tempb{#3}%
653     \endgroup
654   \fi
655 }
```



```

653     \ifx\@tempa\@tempb\else
654         \PackageWarning{tikz-timing}{%
655             To many columns in tikztimingtable row! Only two are allowed%
656             }-{}-{}-%
657     \fi
658 \endgroup
659 \fi
660 \ifx\tikztiming@emptycell#2%
661     \def\next{\tikztimingtable@row@{#1}-{}}%
662 \else
663     \def\next{\tikztimingtable@row@{#1}-{#2}}%
664 \fi
665 \next
666 }%

```

**\tikztimingtable@row@**

```

667 \def\tikztimingtable@row@{#1#2{%
668     \addtocounter{tikztiming@nrows}{1}%
669     \path ($ (last row) + (0,-1*\rowdist) $) coordinate (last row);
670     \path ($ (last row) + (-1*\colldist,0) $) node [anchor=base east,timing/name]
671         {\tikztiminglabel{#1}};
672     \@ifnextchar{[%
673         {\tikztiming@tabletiming}%
674         {\tikztiming@tabletiming[]}%
675     #2\relax
676     \path let \p1 = (timing/table/bottom right), \p2 = (timing/end base) in
677         coordinate (timing/table/bottom right) at ({max(\x1,\x2)},\y2);
678     %
679     \pgfmathparse{max(\tikztiming@maxwidth,\tikztimingwidth)}%
680     \let\tikztiming@maxwidth\pgfmathresult
681     \@ifnextchar\extracode
682     {%
683         \let\extracode\relax
684         \tikztimingextracode
685     }%
686     {%
687         \@ifnextchar\endtikztimingtable@
688         {\@gobble}{\tikztimingtable@row}%
689     }%
690 }

```

**\tikztiming@tabletiming**

```

691 \def\tikztiming@tabletiming[#1]#2\relax{%
692     \let\lastchar\empty
693     \let\settings\empty
694     \ifx\relax#1\relax\else
695         \tikztiming@testoptarg#1\relax\relax

```

```

696 \fi
697 \edef\@tempa{\noexpand\timing[name=row\the\c@tikztiming@nrows,\settings]
698   at (last row)}%
699 \expandafter\@tempa\expandafter{\expandafter[\lastchar]#2};
700 }

```

#### `\tikztiming@fulltablegrid`

```

701 \newcommand*\tikztiming@fulltablegrid[1][]{%
702   \begin{pgfonlayer}{background}
703     \scope[xstep={\timingwidth/2.},ystep={\timingheight/2.},
704     shift={(\timing/table/bottom left)},timing/table/grid,#1]
705     \draw (0,0) grid
706       ($(\timing/table/top right) - (\timing/table/bottom left)$);
707   \endscope
708 \end{pgfonlayer}
709 }

```

#### `\tikztiming@tablegrid`

```

710 \newcommand*\tikztiming@tablegrid[1][]{%
711   \begin{pgfonlayer}{background}
712     \scope[xstep={\timingwidth/2.},ystep={\timingheight/2.},timing/table/grid,#1]
713     \foreach \y in {1,...,\nrows} {%
714       \draw {[shift={($(\timing/table/bottom left) + (0,\y*\rowdist) -
715         (0,\rowdist)$)}] let \p1 = (\timing/table/bottom right) in (0,0) grid
716         (\x1,1)};
717     }%
718   \endscope
719 \end{pgfonlayer}
720 }

```

#### `\tikztiming@tablerules`

```

721 \newcommand*\tikztiming@tablerules[1][]{%
722   \draw [timing/table/rules,#1] let
723     \p1 = (current bounding box.north west),
724     \p2 = (current bounding box.south east),
725     \p3 = (last row)
726   in
727     (\x1-\tabcolsep,\rowdist) -- (\x2+\tabcolsep,\rowdist)
728     ($(\x1-\tabcolsep,\y3) - (0,\rowdist-1)$) --
729     ($(\x2+\tabcolsep,\y3) - (0,\rowdist-1)$)
730   ;
731 }

```

#### `\tikztiming@horlines`

```

732 \newcommand*\tikztiming@horlines[2] [] {%
733   \begingroup
734     \def\list{#2}%
735     \ifx\list\empty
736       \def\list{1,2,...,\nrows}%
737     \fi
738     \foreach \row in \list%
739       \draw [timing/table/lines,#1] let
740         \p1 = (timing/table/bottom right)
741         in
742         (0,\rowdist-\row*\rowdist) -- +(\x1,0);
743   \endgroup
744 }

```

### **\tikztiming@vertlines**

```

745 \newcommand*\tikztiming@vertlines[2] [] {%
746   \begingroup
747     \def\list{#2}%
748     \ifx\list\empty
749       \def\list{0,1,...,\twidth}%
750     \fi
751     \draw [timing/table/lines,#1] let
752       \p1 = ($ (timing/table/bottom right) - (0,2) $)
753       in
754       \foreach \clk in \list {
755         (\clk,+1.5) -- +(0,\y1)
756       }
757     ;
758   \endgroup
759 }

```

## **5.5 Shape**

```

760 \pgfdeclareshape{tikztiming@shape}{%
761   \savedanchor\northeast{\pgfpointxy{.5*\tikztimingwidth}{.5}}%
762   \savedanchor\northeastborder{\pgfpointxy{.5*\tikztimingwidth+0.1}{.6}}%
763   \savedanchor\startpoint{%
764     \pgfpointanchor{timing/start}{center}%
765     \@tempdima=\pgf@x
766     \@tempdimb=\pgf@y
767     \pgfpointxy{-.5*\tikztimingwidth}{-.5}%
768     \advance\pgf@x by \@tempdima
769     \advance\pgf@y by \@tempdimb
770   }%
771   \savedanchor\endpoint{%
772     \pgfpointanchor{timing/end}{center}%
773     \@tempdima=\pgf@x

```

```

774 \tempdimb=\pgf@y
775 \pgfpointxy{-.5*\tikztimingwidth}{-.5}%
776 \advance\pgf@x by \tempdima
777 \advance\pgf@y by \tempdimb
778 }%
779 \anchor{center}{\pgfpointorigin}%
780 \anchor{start}{\startpoint}%
781 \anchor{end}{\endpoint}%
782 %
783 \anchor{origin}{\northeast \pgf@x=-\pgf@x \pgf@y=-\pgf@y }%
784 \anchor{east}{\northeast \pgf@y=0pt }%
785 \anchor{west}{\northeast \pgf@y=0pt \pgf@x=-\pgf@x }%
786 \anchor{north}{\northeast \pgf@x=0pt }%
787 \anchor{north west}{\northeast \pgf@x=-\pgf@x }%
788 \anchor{north east}{\northeast }
789 \anchor{high mid}{\northeast \pgf@x=0pt }%
790 \anchor{high start}{\northeast \pgf@x=-\pgf@x }%
791 \anchor{high end}{\northeast }
792 \anchor{south}{\northeast \pgf@x=0pt \pgf@y=-\pgf@y }%
793 \anchor{south west}{\northeast \pgf@x=-\pgf@x \pgf@y=-\pgf@y }%
794 \anchor{south east}{\northeast \pgf@y=-\pgf@y }%
795 \anchor{low mid}{\northeast \pgf@x=0pt \pgf@y=-\pgf@y }%
796 \anchor{low start}{\northeast \pgf@x=-\pgf@x \pgf@y=-\pgf@y }%
797 \anchor{low end}{\northeast \pgf@y=-\pgf@y }%
798 \anchor{mid}{\pgfpointorigin}%
799 \anchor{mid east}{\northeast \pgf@y=0pt }%
800 \anchor{mid west}{\northeast \pgf@y=0pt \pgf@x=-\pgf@x }%
801 \anchor{mid end}{\northeast \pgf@y=0pt }%
802 \anchor{mid start}{\northeast \pgf@y=0pt \pgf@x=-\pgf@x }%
803 \anchor{base}{\northeast \pgf@x=0pt \pgf@y=-\pgf@y }%
804 \anchor{base west}{\northeast \pgf@x=-\pgf@x \pgf@y=-\pgf@y }%
805 \anchor{base east}{\northeast \pgf@y=-\pgf@y }%
806 \anchorborder{%
807 \tempdima=\pgf@x
808 \tempdimb=\pgf@y
809 \pgfpointborderrectangle{\pgfpoint{\tempdima}{\tempdimb}}{\northeastborder}%
810 }%
811 }

```

## 5.6 Other Macros

### `\tikztiming@iflower`

```

812 \def\tikztiming@iflower#1{%
813 \begingroup
814 \edef\@tempa{‘#1}%
815 \ifnum\@tempa=\lccode\@tempa
816 \endgroup
817 \expandafter

```

```

818     \@firstoftwo
819   \else
820     \endgroup
821     \expandafter
822     \@secondoftwo
823   \fi
824 }

\timingwidth
\timingheight

825 \def\timingwidth{1}%
826 \def\timingheight{1}%

\tikztiming@internaldefs

827 \def\tikztiming@internaldefs#1{%
828   \def\draw{\noexpand\draw}%
829   \def\path{\noexpand\path}%
830   \def\fill{\noexpand\fill}%
831   \def\width{###1*\noexpand\timingwidth}%
832   \def\fwidht{\noexpand\timingwidth}%
833   \def\height{\noexpand\timingheight}%
834   \def\slope{\noexpand\timingslope}%
835   \def\zslope{\noexpand\timingzslope}%
836   \def\dslope{\noexpand\timingdslope}%
837   \def\gslope{0}%
838   \lowercase{%
839     \def\style{timing/#1}%
840     \def\bgstyle{timing/#1/background}%
841   }%
842   \def\newdraw{\tikztiming@newdraw}%
843   \def\newdrawns{\tikztiming@newdraw@nosave}%
844   \def\code##1{ [/utils/exec={\unexpanded{##1}}] }%
845 }

\tikztimingsetslope

846 \def\tikztimingsetslope#1{%
847   \pgfmathparse{min(1.0,{max(0.0,#1)})}%
848   \let\tikztiming@slope\pgfmathresult
849   \edef\timingslope{\tikztiming@slope*\noexpand\timingwidth}%
850 }

\tikztimingsetdslope

851 \def\tikztimingsetdslope#1{%
852   \pgfmathparse{min(1.0,{max(0.0,#1)})}%

```

```

853 \let\tikztiming@dslope\pgfmathresult
854 \edef\timingdslope{\tikztiming@dslope*\noexpand\timingwidth}%
855 }

```

### **\tikztimingsetzslope**

```

856 \def\tikztimingsetzslope#1{%
857   \pgfmathparse{min(1.0,{max(0.0,#1)})}%
858   \let\tikztiming@zslope\pgfmathresult
859   \edef\timingzslope{\tikztiming@zslope*\noexpand\timingwidth}%
860 }
861 \tikztimingsetslope{0.10}%
862 \tikztimingsetdslope{0.20}%
863 \tikztimingsetzslope{0.05}%

```

### **\tikztiminguse**

```

864 \def\tikztiminguse#1{%
865   \ifundefined{\tikztiming@prefix#1@general}%
866     {\PackageWarning{Can not use transition macro for '#1'.}{-}{-}}%
867     {\@nameuse{\tikztiming@prefix#1@general}}%
868 }

```

### **\tikztimingdef**

```

869 \def\tikztimingdef#1{%
870   \tikztimingdef@#1\relax%
871 }

```

### **\tikztimingdef@**

```

872 \def\tikztimingdef@#1#2\relax#3{%
873   \ifx\relax#2\relax
874     \tikztiming@nameedef[#1]{#1}{#3}%
875   \else
876     \tikztiming@nameedef[#2]{#1#2}{#3}%
877   \fi
878 }

```

### **\tikztiminglet**

```

879 \def\tikztiminglet#1#2{%
880   \tikztiminglet@#1\relax#2\relax
881 }

```

`\tikztiminglet@`

```
882 \def\tikztiminglet@#1#2\relax#3#4\relax{%
883   \tikztiming@namelet{#1#2}{#3#4}%
884   \tikztiming@namelet{#1#2@general}{#3#4@general}%
885   \tikztiming@iflower{#1}{}%
886   {\tikztiming@iflower{#2}%
887    {%
888     \lowercase{\tikztiminglet@{#1}{#2}\relax{#3}{#4}}\relax
889    }%
890    {%
891     \uppercase{\lowercase{%
892      \uppercase{\lowercase{\tikztiminglet@{#1}{#2}}\relax{#3}{#4}}\relax
893      \lowercase{\uppercase{%
894       \lowercase{\uppercase{\tikztiminglet@{#1}{#2}}\relax{#3}{#4}}\relax
895      }%
896     }%
897 }
```

`\tikztiming@chars`

Initial definition of character list. Will gobble the separation comma in front of the first character which is added to the list.

```
898 \def\tikztiming@chars#1{}
```

`\tikztiming@ifcharexists`

```
899 \def\tikztiming@ifcharexists#1{%
900   \def\tikztiming@ifcharexists@##1,#1,##2\relax{%
901     \ifx\relax##2\relax%
902       \expandafter\@firstoftwo
903     \else
904       \expandafter\@secondoftwo
905     \fi
906   }%
907   \expandafter\tikztiming@ifcharexists@
908   \expandafter,\tikztiming@chars,#1,\relax%
909 }
```

`\tikztiming@addchar`

```
910 \def\tikztiming@addchar#1{%
911   \tikztiming@ifcharexists{#1}{%
```

```

912 \edef\tikztiming@chars{\tikztiming@chars,#1}%
913 }{}%
914 }

```

#### **\tikztimingchar**

```

915 \def\tikztimingchar#1{%
916 \uppercase{%
917 \tikztiming@addchar{#1}%
918 \tikztimingchar@{#1}}%
919 }

920 \expandafter\def\csname\tikztiming@prefix @start\endcsname#1{%

```

#### **\tikztimingchar@**

```

921 \def\tikztimingchar@#1#2#3{%
922 \tikztiming@nameedef[#1]{#1@start}{#2 coordinate (timing/start) }%
923 \tikztiming@nameedef[#1]{#1}{#2 coordinate (timing/start) #3}%
924 \tikztimingdef{#1#1}{#3}%
925 }

```

#### **\tikztimingalias**

```

926 \def\tikztimingalias#1#2{%
927 \uppercase{\tikztimingalias@{#1}{#2}}%
928 }

```

#### **\tikztimingalias@**

```

929 \def\tikztimingalias@#1#2{%
930 \tikztiming@namelet{#1}{#2}%
931 \tikztiming@namelet{#1@start}{#2@start}%
932 \lowercase{%
933 \tikztiming@namelet{#1}{#2}%
934 \tikztiming@namelet{#1@start}{#2@start}%
935 }%
936 \tikztiminglet{#1#1}{#2#2}%
937 \@for\@tempa:=\tikztiming@chars\do{%
938 \expandafter\tikztiminglet@@
939 \expandafter{\@tempa}{#1}{#2}%
940 }%
941 }

```



`\tikztimingecopy`

```
942 \def\tikztimingecopy#1#2{%
943   \uppercase{\tikztimingecopy@{#1}{#2}}%
944 }
```

`\tikztimingecopy@`

```
945 \def\tikztimingecopy@#1#2{%
946   \tikztimingchar{#1}{}%
947   \tikztimingdef{#1}{\tikztiminguse{#2}{##1}}%
948   \tikztiming@nameedef[#1]{#1@start}{\tikztiminguse{#2@start}{##1}}%
949   \lowercase{%
950     \@ifundefined{\tikztiming@prefix#2}{%
951       \tikztimingdef{#1}{\tikztiminguse{#2}{##1}}%
952       \tikztiming@nameedef[#1]{#1@start}{\tikztiminguse{#2@start}{##1}}%
953     }%
954   }%
955   \tikztimingdef{#1#1}{\tikztiminguse{#2#2}{##1}}%
956   \@for\@tempa:=\tikztiming@chars\do{%
957     \expandafter\tikztimingdef@@
958     \expandafter{\@tempa}{#1}{#2}%

```

Handle lowercase macros:

```
959   \expandafter\lowercase\expandafter{\expandafter\def\expandafter\@tempb
960   \expandafter{\@tempa}}%
961   \@ifundefined{\tikztiming@prefix#2\@tempb}{%
962     \expandafter\tikztimingdef@@
963     \expandafter{\@tempb}{#1}{#2}%
964   }%
965   }%
966 }
```

`\tikztiminglet@@`

```
967 \def\tikztiminglet@@#1#2#3{%
968   \tikztiminglet@@@#1#2#3%
969   % Should stay, cause no harm:
970   \lowercase{\tikztiminglet@@@#1}{#2#3}%
971   \lowercase{\tikztiminglet@@@#1#2#3}%
972   \lowercase{\uppercase{\tikztiminglet@@@#1}{#2#3}}%
973 }
```

`\tikztiminglet@@`

```
974 \def\tikztiminglet@@#1#2#3{%
975   \tikztiminglet{#1#2}{#1#3}%
976   \tikztiminglet{#2#1}{#3#1}%
977 }
```

`\tikztimingdef@@`

```
978 \def\tikztimingdef@@#1#2#3{%
979   \tikztimingdef{#1#2}{\tikztiminguse{#1#3}{##1}}%
980   \tikztimingdef{#2#1}{\tikztiminguse{#3#1}{##1}}%
981 }
```

`\tikztiming@savesettings`

```
982 \def\tikztiming@savesettings{%
983   \xdef\tikztiming@saved@settings{%
984     {\tikztiming@slope}%
985     {\tikztiming@dslope}%
986     {\tikztiming@zslope}%
987     {\the\pgflinewidth}%
988   }%
989 }
```

`\tikztiming@restoresettings`

```
990 \def\tikztiming@restoresettings{%
991   \expandafter\tikztiming@restoresettings@
992   \tikztiming@saved@settings\relax
993 }
```

`\tikztiming@restoresettings@`

```
994 \def\tikztiming@restoresettings@#1#2#3#4\relax{%
995   \tikztimingsetslope{#1}%
996   \tikztimingsetdslope{#2}%
997   \tikztimingsetzslope{#3}%
998   \pgfsetlinewidth{#4}%
999 }
```

`\tikztiming@newdraw`

```

1000 \def\tikztiming@newdraw{%
1001   node [timing/save] (timing@save) {};%
1002   \draw [timing/restore] (timing@save) ++(0,0)
1003 }

```

`\tikztiming@newdraw`

```

1004 \def\tikztiming@newdraw@nosave{%
1005   node [timing/coord] (timing@save) {};%
1006   \draw (timing@save) ++(0,0)
1007 }

```

## 5.7 Definition of Timing Characters

```

1008 \tikztimingchar{H}{++(0,\height)}{-- ++(#1,0)}
1009
1010 \tikztimingchar{L}{++(0,0)}{-- ++(#1,0)}
1011
1012 \tikztimingchar{Z}{++(0,\height/2.)}{%
1013   \newdraw [\style]
1014   -- ++(#1,0)
1015 }
1016
1017 \tikztimingchar{X}{}{}%
1018 \tikztimingchar{D}{}{}%
1019 \tikztimingchar{U}{}{}%
1020 %\tikztimingchar{O}{}{}%
1021 \tikztimingchar{M}{}{}%
1022
1023 \tikztimingchar{G}{++(0,0)}{-- ++(\gslope,\height) -- ++(\gslope,-\height)}
1024 \tikztimingchar{S}{++(0,0)}{++(#1,0)}
1025
1026 \tikztimingdef{DD}{
1027   node [timing/save] (timing@save) {}; \path [\bgstyle] (timing@save) ++(0,0)
1028     +(0.5*\dslope,0.5*\height) -- +(\dslope,0)
1029   -- +(#1,0)
1030   -- +($ (#1,0) + 0.5*(\dslope,\height) $)
1031   -- +(#1,\height)
1032   -- +(\dslope,\height) -- cycle;
1033   \draw [timing/restore,\style] (timing@save) ++(0,0)
1034   node [timing/save] (timing@dstart) at +(\dslope/2.,\height/2.) {}
1035   -- +(\dslope,+\height) -- +(#1,+\height) ++(0,+\height)
1036   -- +(\dslope,-\height) -- ++(#1,-\height)
1037 }
1038 \tikztiming@namelet{D@edge@D}{DD}
1039 \tikztiming@namelet{D@edge@D@general}{DD@general}
1040
1041 \tikztimingchar{D}{++(0,0)}{
1042   node [timing/save] (timing@save) {}; \path [\bgstyle] (timing@save) ++(0,0)

```

```

1043 -- +(#1,0)
1044 -- +($ (#1,0) + 0.5*(\dslope,\height) $)
1045 -- +(#1,\height)
1046 -- +(0,\height)
1047 -- cycle;
1048 \draw [timing/restore,\style] (timing@save) ++(0,0)
1049 node [timing/save] (timing@dstart) at +(-\dslope/2.,\height/2.0) {}
1050 -- +(#1,0) ++(0,+\height)
1051 -- ++(#1,0) ++(0,-\height)
1052 }
1053
1054 \tikztimingdef{DD}{
1055 node [timing/save] (timing@save) {}; \path [\bgstyle] (timing@save) ++(0,0)
1056 -- +(#1,0)
1057 -- +($ (#1,0) + 0.5*(\dslope,\height) $)
1058 -- +(#1,\height)
1059 -- +(0,\height)
1060 -- cycle;
1061 \newdraw [\style]
1062 -- +(#1,0) ++(0,+\height)
1063 -- ++(#1,0) ++(0,-\height)
1064 }
1065
1066 \tikztiming@namelet{D@edge@@start}{D@start}
1067 \tikztiming@namelet{d@edge@@start}{d@start}
1068

```

**\tikztiming@trans@D@fill**

```

1069 \def\tikztiming@trans@D@fill#1#2{%
1070 node [timing/save] (timing@save) {}; \path [\bgstyle] (timing@save) ++(0,0)
1071 -- +(0.5*\dslope,-0.5*\height)
1072 -- +($ (#1,-0.5*\height) - (#2,0) $)
1073 -- +(0.5*\dslope,0.5*\height)
1074 -- +(0,\height)
1075 -- +($ (#2,\height) - (#1,0) + (0.5*\dslope,0) $)
1076 -- cycle;
1077 \draw [timing/restore,\style] (timing@save) ++(0,0)
1078 node [timing/save] (timing@dstart) {}
1079 }
1080 \tikztimingdef{HH}{-- ++(#1,0)}
1081 \tikztimingdef{LL}{-- ++(#1,0)}
1082 \tikztimingdef{HL}{-- ++(\slope,-\height) \tikztiminguse{HH}{#1-\slope}}
1083 \tikztimingdef{LH}{-- ++(\slope, \height) \tikztiminguse{LL}{#1-\slope}}
1084
1085 \tikztimingdef{HG}{-- ++(\gslope,-\height) -- ++(\gslope,+\height)}
1086 \tikztimingdef{LG}{-- ++(\gslope,+\height) -- ++(\gslope,-\height)}
1087 \tikztimingdef{ZG}{

```

```

1088 -- ++(\gslope,-\height/2.0)
1089 -- ++(\gslope,+\height)
1090 -- ++(\gslope,-\height/2.0)
1091 }
1092 \tikztiminglet{DG}{LG}
1093
1094 \tikztiminglet{HS}{S}
1095 \tikztiminglet{LS}{S}
1096 \tikztiminglet{ZS}{S}
1097 \tikztiminglet{DS}{S}
1098 \tikztiminglet{TS}{S}
1099
1100 \tikztimingdef{LZ}{
1101   \newdraw [\style]
1102   -- ++(\zslope,+\height/2.) -- ++($ (#1,0) - (\zslope,0) $)
1103 }
1104 \tikztimingdef{HZ}{%
1105   \newdraw [\style]
1106   -- ++(\zslope,-\height/2.) -- ++($ (#1,0) - (\zslope,0) $)
1107 }
1108 \tikztimingdef{ZH}{
1109   \newdraw
1110   -- ++(\zslope,+\height/2.) -- ++($ (#1,0) - (\zslope,0) $)
1111 }
1112 \tikztimingdef{ZL}{%
1113   \newdraw
1114   -- ++(\zslope,-\height/2.) -- ++($ (#1,0) - (\zslope,0) $)
1115 }
1116
1117 \tikztimingdef{DZ}{
1118   -- ++(\dslope/2.,+\height/2.)
1119   ++(-\dslope/2.,+\height/2.)
1120   -- ++(\dslope/2.,-\height/2.)
1121   \newdraw [\style]
1122   -- ++($ (#1,0) - (\dslope/2.,0) $)
1123 }
1124
1125 \tikztimingdef{ZD}{
1126   \tikztiming@trans@D@fill{#1}{0}%
1127   -- ++(\dslope/2.,\height/2.) -- ++($ (#1,0) - (\dslope/2.,0) $)
1128   ++($ -1*(#1,0) + (0,-\height/2.) $)
1129   -- ++(\dslope/2.,-\height/2.) -- ++($ (#1,0) - (\dslope/2.,0) $)
1130 }
1131
1132 \tikztimingdef{LD}{
1133   -- ++(0.5*\dslope,0.5*\height)
1134   \tikztiming@trans@D@fill{#1}{0.5*\dslope}%
1135   -- ++(0.5*\dslope,0.5*\height)
1136   -- ++($ (#1,0) - (\dslope,0) $)
1137   ++($ -1*(#1,0) + (0,-\height) $)          ++(\dslope/2.,+\height/2.)

```

```

1138 -- ++(\dslope/2.,-\height/2.) -- ++($ (#1,0) - (\dslope,0) $)
1139 }
1140
1141 \tikztimingdef{DL}{
1142 -- ++( \dslope/2.,+\height/2.)
1143 ++(-\dslope/2.,+\height/2.)
1144 -- ++(\dslope/2.,-\height/2.)
1145 \newdraw [\style]
1146 -- ++(\dslope/2.,-\height/2.)
1147 -- ++($ (#1,0) - (\dslope,0) $)
1148 }
1149
1150 \tikztimingdef{HD}{
1151 -- ++(0.5*\dslope,-0.5*\height)
1152 \tikztiming@trans@D@fill{#1}{0.5*\dslope}%
1153 -- ++(0.5*\dslope,-0.5*\height)
1154 -- ++($ (#1,0) - (\dslope,0) $)
1155 ++($ -1*(#1,0) + (0,+\height) $) ++(\dslope/2.,-\height/2.)
1156 -- ++(\dslope/2.,+\height/2.) -- ++($ (#1,0) - (\dslope,0) $)
1157 ++(0,-\height)
1158 }
1159
1160 \tikztimingdef{DH}{
1161 ++(0,+\height)
1162 -- ++(+\dslope/2.,-\height/2.)
1163 ++(-\dslope/2.,-\height/2.)
1164 -- ++(\dslope/2.,+\height/2.)
1165 \newdraw [\style]
1166 -- ++(\dslope/2.,+\height/2.)
1167 -- ++($ (#1,0) - (\dslope,0) $)
1168 }
1169
1170
1171 \tikztimingalias{M}{Z}
1172 \tikztimingchar{M}{++(0,\height/2.)}{
1173 \newdraw [\style/decoration,\style]
1174 -- ++(#1,0)
1175 }
1176 \tikztimingdef{MG}{
1177 \newdraw [timing/m]
1178 -- ++(\gslope,-\height/2.0)
1179 -- ++(\gslope,+\height)
1180 -- ++(\gslope,-\height/2.0)
1181 }
1182
1183 \tikztimingdef{MZ}{
1184 \newdraw [\style]
1185 \tikztiminguse{ZZ}{#1}
1186 }
1187

```

```

1188 \tikztimingdef{ZM}{
1189   \newdraw [\style]
1190   \tikztiminguse{MM}{#1}
1191 }
1192
1193 \tikztimingdef{LM}{
1194   \newdraw [\style]
1195   -- ++($ (1/8,0) + (0,\height/2) $)
1196   \newdraw [\style/decoration,\style]
1197   -- ++($ (-1/8,0) + (#1,0) $)
1198 }
1199
1200 \tikztimingdef{HM}{
1201   \newdraw [\style]
1202   -- ++($ (1/8,0) + (0,-1*\height/2) $)
1203   \newdraw [\style/decoration,\style]
1204   -- ++($ (-1/8,0) + (#1,0) $)
1205 }
1206
1207 \tikztimingdef{DM}{
1208   -- +($ (1/8,0) + (0,\height*.50) $)
1209   + (0,\height)
1210   -- ++($ (1/8,0) + (0,\height*.50) $)
1211   \newdraw [\style/decoration,\style]
1212   -- ++($ (-1/8,0) + (#1,0) $)
1213 }
1214
1215 \newcounter{tikztimingtrans}
1216 \newcounter{tikztimingtranspos}
1217
1218 \tikztimingchar{T}{++(0,0)}{
1219   -- ++(#1,0)
1220 }
1221
1222 \tikztimingdef{HT}{%
1223   {[\style]
1224     \code{\setcounter{tikztimingtrans}{-1}}
1225     -- ++(\slope,\value{tikztimingtrans}*\height) -- ++($ (#1,0) - (\slope,0) $)
1226   }
1227 }
1228
1229 \tikztimingdef{LT}{%
1230   {[\style]
1231     \code{\setcounter{tikztimingtrans}{+1}}
1232     -- ++(\slope,\value{tikztimingtrans}*\height) -- ++($ (#1,0) - (\slope,0) $)
1233   }
1234 }
1235
1236 \tikztimingdef{TL}{%
1237   \code{\setcounter{tikztimingtranspos}{\value{tikztimingtrans}}}%

```

```

1238 \addtocounter{tikztimingtranspos}{+1}}
1239 -- ++(\slope, -0.5*\value{tikztimingtranspos}*\height) -- ++($ (#1,0) - (\slope,0) $)
1240 }
1241
1242 \tikztimingdef{TH}{%
1243 \code{\setcounter{tikztimingtranspos}{\value{tikztimingtrans}}}%
1244 \addtocounter{tikztimingtranspos}{-1}}
1245 -- ++(\slope, -0.5*\value{tikztimingtranspos}*\height) -- ++($ (#1,0) - (\slope,0) $)
1246 }
1247
1248 \tikztimingdef{TZ}{%
1249 \newdraw [\style]
1250 \code{\setcounter{tikztimingtrans}{-\value{tikztimingtrans}}}%
1251 -- ++(\slope,\value{tikztimingtrans}*\height/2.)
1252 -- ++($ (#1,0) - (\slope,0) $)
1253 }
1254
1255 \tikztimingdef{TG}{%
1256 -- +(\gslope,-1*\value{tikztimingtrans}*\height)
1257 -- +(\gslope,0)
1258 }
1259
1260 \tikztimingdef{ZT}{%
1261 \newdraw
1262 {[\style]
1263 \code{\setcounter{tikztimingtrans}{-\value{tikztimingtrans}}}%
1264 -- ++(\slope,\value{tikztimingtrans}*\height/2.)
1265 -- ++($ (#1,0) - (\slope,0) $)
1266 }
1267 }
1268
1269 \tikztimingdef{TT}{%
1270 {[\style]
1271 \code{\setcounter{tikztimingtrans}{-\value{tikztimingtrans}}}%
1272 -- ++(\slope,\value{tikztimingtrans}*\height) -- ++($ (#1,0) - (\slope,0) $)
1273 }
1274 }
1275
1276 \tikztimingdef{TD}{
1277 \code{\setcounter{tikztimingtrans}{-\value{tikztimingtrans}}}%
1278 \code{\setcounter{tikztimingtranspos}{\value{tikztimingtrans}}}%
1279 \addtocounter{tikztimingtranspos}{-1}}
1280 -- ++(0.5*\dslope,+0.5*\value{tikztimingtrans} * \height)
1281 \tikztiming@trans@D@fill{#1}{0.5*\dslope}%
1282 -- ++(0.5*\dslope,+0.5*\value{tikztimingtrans} * \height)
1283 -- ++($ (#1,0) - (\dslope,0) $)
1284 ++($ -1*(#1,\value{tikztimingtrans}*\height) $)
1285 ++(\dslope/2.,+1*\value{tikztimingtrans}*\height/2.)
1286 -- ++(\dslope/2.,-1*\value{tikztimingtrans}*\height/2.)
1287 -- ++($ (#1,0) - (\dslope,0) $)

```



```

1288      ++(0,\value{tikztimingtranspos}*\height/2.)
1289 }
1290
1291 \tikztimingdef{DT}{
1292   \code{\setcounter{tikztimingtrans}{-1}}
1293   \tikztiminguse{DL}{#1}%
1294 }
1295
1296 \tikztimingdef{MT}{%
1297   \newdraw
1298   {[\style]
1299   -- ++(\slope,\value{tikztimingtrans}*\height/2.) -- ++($ (#1,0) - (\slope,0) $)
1300   }
1301 }
1302
1303 \tikztimingdef{TM}{%
1304   \newdraw [\style]
1305   \code{\setcounter{tikztimingtrans}{-\value{tikztimingtrans}}}
1306   -- ++($ (1/8,0) + (0,\value{tikztimingtrans}*\height/2) $)
1307   \newdraw [\style/decoration,\style]
1308   -- ++($ (-1/8,0) + (#1,0) $)
1309 }
1310
1311 \tikztimingecopy{C}{T}
1312 \def\tikztiming@nocombine@T{%
1313 \def\tikztiming@nocombine@C{%
1314 \def\tikztiming@nocombine@t{%
1315 \def\tikztiming@nocombine@c{%
1316 \def\tikztiming@nocombine@M{%
1317 \def\tikztiming@nocombine@m{%
1318
1319
1320 \tikztimingecopy{U}{D}
1321 \tikztimingdef{UD}{\tikztiminguse{D@edge@D}{#1}}
1322 \tikztimingdef{DU}{\tikztiminguse{D@edge@D}{#1}}
1323
1324 %\tikztimingecopy{O}{D}
1325 \tikztimingecopy{X}{Z}

```

## Change History

v0.3		
General:	First released version	.. 1
v0.4		
General:	Added output routine which combines successive occurrences of the same character.	Added parser for rows in <code>{tikztimingtable}</code> . This makes the syntax much more

stable. Also replaced row counter with TikZ coordinates which is more user-friendly. . .	1	v0.4a	General: Added <code>\tablerules</code> macro. Changed default style of inline nodes to <code>coordinate</code> .	1
In-line Nodes, e.g. to mark positions inside the diagram. . . . .	1	v0.5	General: Added PGF shape for timing diagrams. Added meta-characters. Changed ‘M’ character to use PGF decorations. Added special ‘B’ character to reduce width of next character. Changed <code>\timing</code> syntax to include an ‘at’ before the coordinate. Bug fix for use with the ‘calc’ package. . . . .	1
Released as v0.4 . . . . .	1			
Removed own macros for lowercase characters. They are now handled by the uppercase macros which receive half of the width. Exceptions are possible like for the ‘m’ character. . . .	1			
User macros to draw grids and lines inside table. . . . .	1			