The ⟨tikz⟩⟨timing⟩ package

Martin Scharrer
martin@scharrer-online.de

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

Version v0.6a – 2009/07/28

1 Introduction

This package uses the tikz[1] 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.

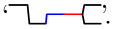
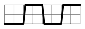
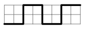
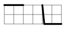
The signal levels of the timing diagram can be given by corresponding characters/letters like ‘H’ for *Logical High* or ‘L’ for *Logical Low*. So e.g. ‘{HLZXD}’ gives ‘’. Lowercase characters only produce a signal with half the width while uppercase characters produce one with the “full” width, called 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	
C	Clock (no slope)	or	or	
M	Metastable Condition			
G	Glitch (zero width)			
S	Space (nothing)			

[1] Part of the pgf package, CTAN: http://www.ctan.org/pkg/pgf

2 Usage

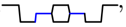
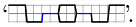
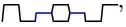
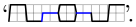
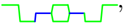
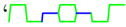
2.1 Macro for use in Text Mode

\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 ''

\texttimingbefore (*defaults to*: ⟨*empty*⟩)

Table 2: Overview over all transitions.

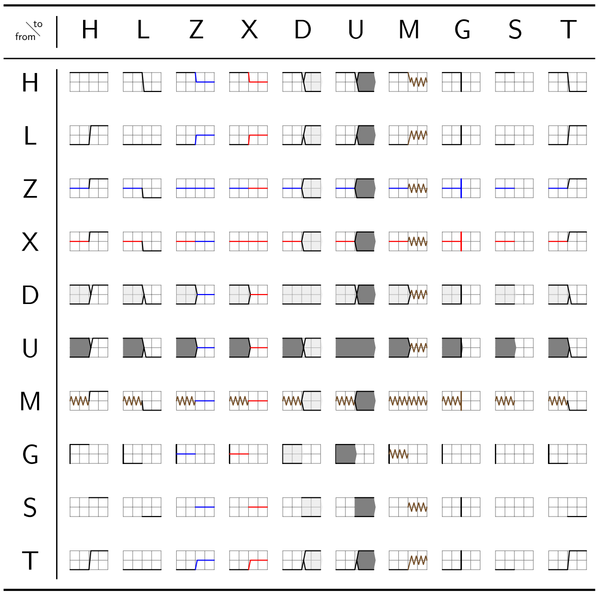
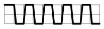
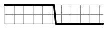
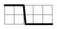
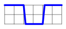
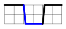
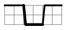


Table 3: Modifiers for Timing Characters

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{hl}	
⟨*number*⟩⟨*character*⟩	Sets width of next signal to given number. Half of it if character is in lower case.	2.6H5.2l	
⟨*number*⟩B	Subtracts the given number from the width of the next character. "*Backwards*"	H .5B L	
⟨*number*⟩F	Adds the given number to the width of the next character. "*Forwards*"	H .5F 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 12.	

`\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 LaTeX-stylish way to do this is `\renewcommand*{\textttimingbefore}{\texttiminggrid}`.

`\texttiminggrid`

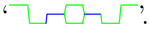
This macro should only be used inside `\texttimingbefore` or `\texttimingafter` and draws a grid of the full size of the `\texttiming` diagram.

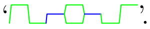
2.2 Macro for use inside TikZ-Pictures

`\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`, `\drawn`, `\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.

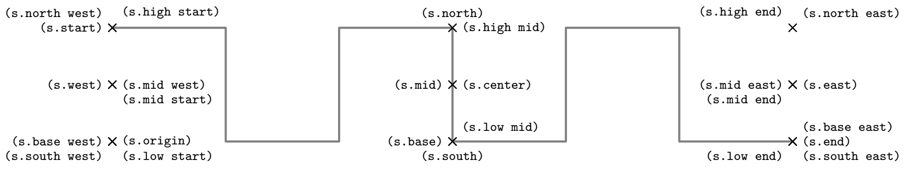


Figure 1: Timing Shape Anchors. The `start` and `end` anchors mark the start and end of the timing signal.

2.3 Table for Timing Diagrams

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

```
\fulltablegrid[⟨TikZ Settings⟩]
```

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[2] 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*\rowdist lower than the previous one. The vertical unit is 1 signal height and the default row distance is '2' (=2×signal height). This means that a normal table with three rows goes from $y = +1$ (base line at 0 + 1 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 $x = -$\coldist and the text is right align with the right border at this position.

[2] 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 than 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 TEX error.

\tikztimingmetachar{⟨*Meta-Character*⟩}{⟨*Character Group*⟩}

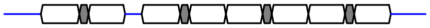
This macro defines the given ⟨*meta-character*⟩ to be identical to the given ⟨*character group*⟩. Alternatively this can also be done using the TikZ style '`timing/metachar={`⟨*Meta-Character*⟩`}{`⟨*Character Group*⟩`}`'.

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:

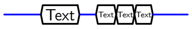
```
\tikztimingmetachar{Y}{2D 0.5U 2D{}}  \texttiming{ZZ Y Z 3Y ZZ}
```

gives:



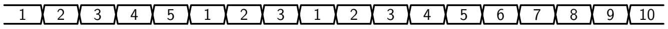
```
\tikztimingmetachar{Y}{2D{Text}}  \tikztimingmetachar{y}{1D{\tiny Text}}
\texttiming{ZZ Y Z 3y ZZ}
```

gives:



```
\newcounter{mycount}
\tikztimingmetachar{Q}{2D{\stepcounter{mycount}\arabic{mycount}}}
\tikztimingmetachar{R}{[/utils/exec=\setcounter{mycount}{0}]}
\texttiming{ 5Q R 3Q R 10Q }
```

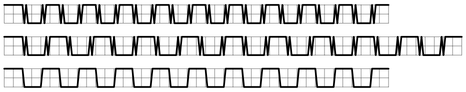
gives:



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

4 Libraries for Further Characters

All default timing characters described in Table 1 are always made available by this package. Further, less-common characters are provided by libraries which are loaded with the macro `\usetikztiminglibrary{⟨library⟩}`. This is done to hold the memory usage of this package small and reduce the risk of collisions with user-defined (meta-)characters. The full syntax for the above macro is `\usetikztiminglibrary[⟨options⟩]{⟨library,...⟩}[⟨date⟩]`, like the one for `\usepackage`. However, no library provides any options so far. The date is used as a version number like for packages.

4.1 Arrows

The library '`arrows`' enables two characters '`A`' and '`W`' which draw vertical up and down *ArroWs*. Such arrows are used in timing diagrams to mark special polarized events, like clock edges of another signal.

The width provided with these character is added as whitespace after the "zero-width" arrow: '`A2AA`' results in '↑↑ ↑ '. This space can be avoided by specifying the width to zero: '`0.A`'. Like the '`C`' and '`T`' characters the subsequent arrow characters are not combined into one.

The arrow tips can be changed using the TikZ styles for this characters. See 3 for more information. The '`A`' character should not be used together with any other characters except with '`S`' (space).

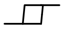
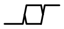
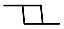
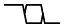
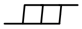
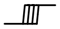
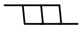
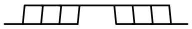
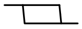
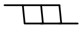
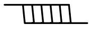
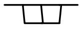
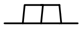
Table 5: Examples for Arrow Characters.

Characters	Resulting Diagram
`AAA`	↑↑↑
`3A`	↑
`3{A}`	↑↑↑
`3A 3A`	↑ ↑
`3a 3a`	↑ ↑
`AW AW`	↑↓↑↓
`3{AW}`	↑↓↑↓↑↓
`3{aw}`	↑↓↑↓↑↓
`2S 2A 3W A W`	↑ ↓ ↑↓

4.2 Either High or Low

The library '`either`' is enabling the '`E`' character which stands for '*either high or low*'. This character is designed to be used with the '`H`' and '`L`' characters to display a uncertainty of a transition. Sometimes a, e.g. low signal can go either to high or stay at low for a certain time before it definitely goes to high. In this case both (or more) possible transitions can be drawn using this character. Like the ‘C’ and ‘T’ characters subsequent ‘E’ characters are not combined into one.

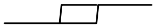
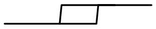
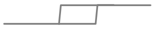
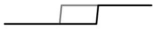
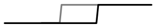
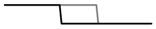
Table 6: Examples for the ‘E’ Character.

Characters	Resulting Diagram
L E H	
L D H	
H E L	
H D L	
L E E H	
L 3{.25E} H	
H E E L	
L EEE HH EEE L	
l e e h	
h e e l	
H 2E L	
H 2{E} L	
H 5{e} L	
H E E H	
L E E L	

4.3 Overlays

The library ‘overlays’ is enabling the ‘O’ character which allows the overlaying of two different groups of timing characters. This character awaits two groups enclosed by braces which are drawn in the given order. The position before the ‘O’ character is saved and restored to draw the second group. The characters of the first group connect to the characters before the ‘O’ modifier but then simply end. The characters following after the groups connect to the characters of the second group as normal. It is the responsibility of the user to make sure that the lines drawn by the first group reconnect to the main lines or do something else useful. The modifier ‘;’ can be used in the first group to restart the drawn line, e.g. to change to a different color. This is not done automatically to give the user the freedom if and where this should happen. It is recommended to start and end both groups with the same character to avoid ugly connection points.

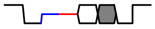
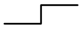
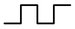
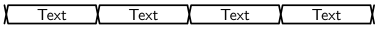
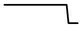
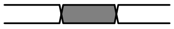
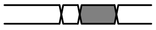
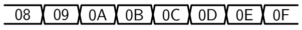
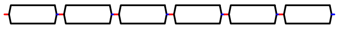
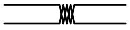
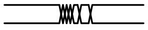
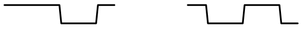
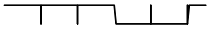
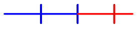
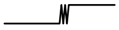
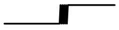
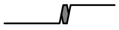
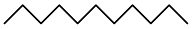
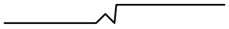
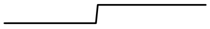
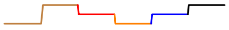
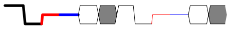
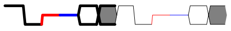
Table 7: Examples for the ‘O’ Overlay Character.

Characters	Resulting Diagram
`LLL O{HH}{LL} HHH`	
`LLL O{HHH}{LL} HHH`	
`LLL O{[gray]HHH}{LLH} HH`	
`LLL O{;[gray]HH.1H;}{LLH} HH`	
`LL O{L;[gray]HH.1H;}{LLLH} HH`	
`HHH O{;[gray]HH.1L;}{LL} LLL`	

5 Examples

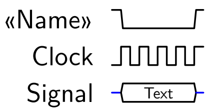
This section shows some examples by putting either the full source code or only the needed characters beside the graphical result. Please note that the displayed syntax is the one of `\timing` where the initial character is declared as optional argument ([⟨*char*⟩]) *inside/together* with the logic characters. The syntax of `\texttiming` 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
`HLZXDUTC`	
`cccc`	
`tttt`	
`[c]cccc`	
`4{c}`	
`4c4c`	
`4{1.8c}`	
`[d] 4{5D{Text}} 0.2D`	
`3.4H 0.6L`	
`DDDUUUDDD`	
`DDD{}DUUDD`	
`8{2D{\hexcountmacro }}`	
`3{2{0.25X 2.5D .25Z}}`	
`DDD{} 3{0.2D{}} DDD`	
`DDD{} 3{0.2D{}} 0.4D{} 0.6D{} DDD`	
`HHHLLH SSSS HLLHHL`	
`HHGHHGGHHLLGLLGH`	
`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`	

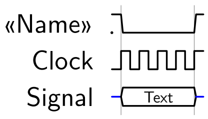
Note: Optional argument must be placed before macro argument if `\texttiming` is used.

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



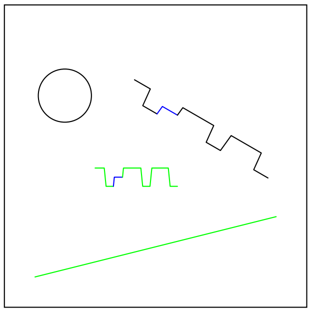
Example 2: {tikztimingtable} without \extracode.

```
\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}
```



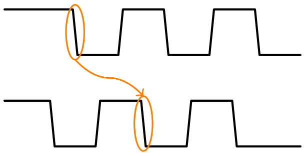
Example 3: {tikztimingtable} with \extracode.

```
\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}
```



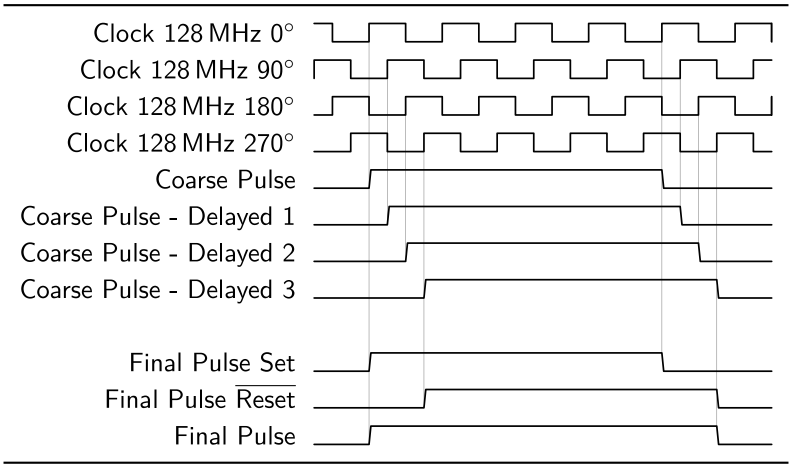
Example 4: \timing inside general {tikzpicture}.

```
\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}
```



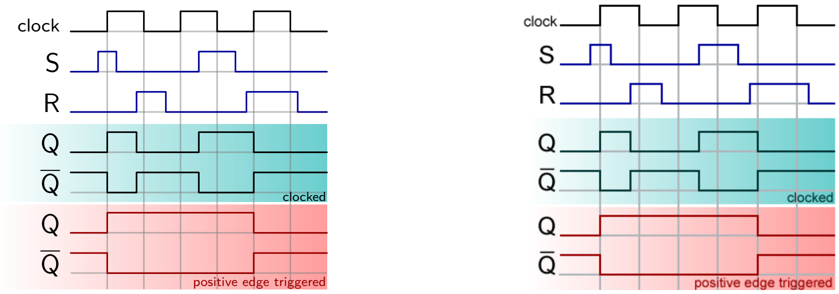
Example 5: Using In-Line Nodes to draw Relationships.

```
\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
  \tablerules
  \begin{pgfonlayer}{background}
    \foreach \n in {1,...,8}
      \draw [help lines] (A\n) -- (B\n);
  \end{pgfonlayer}
\end{tikztimingtable}
```



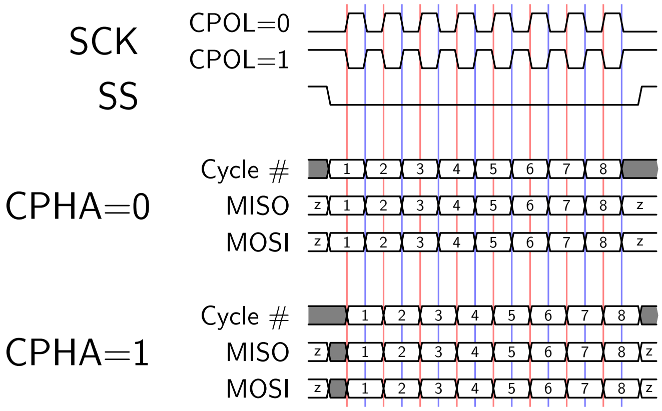
Example 6: Using In-Line Nodes to draw Marker Lines.

```
\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 LLl [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 & LHHHHLL[fgred]\\
  $\overline{\mbox{Q}}$ & HLLLLHH[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

```
\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

6 Implementation

6.1 Package Header

```
⟨*package⟩
\RequirePackage{tikz}

\usetikzlibrary{calc}
\usetikzlibrary{backgrounds}
\usetikzlibrary{decorations.pathmorphing}
\ifx\collect@body\@undefined
  \IfFileExists{environ.sty}
    {\RequirePackage{environ}}
    {\RequirePackage{amsmath}}
\fi

\newcommand*\usetikztiminglibrary[2][]{%
  \edef\tikztiming@library@options{#1}%
  \@ifnextchar{[}%]
    {\tikztiming@library{#2}}%
    {\tikztiming@library{#2}[]}%
}
\def\tikztiming@library#1[#2]{%
  \edef\tikztiming@library@date{#2}%
  \tikztiming@library@#1,\relax
}

\def\tikztiming@library@#1,{%
  \IfFileExists{tikz-timing-#1.sty}%
    {\RequirePackage[\tikztiming@library@options]{tikz-timing-#1}%
      [\tikztiming@library@date]}%
    {\PackageError{tikz-timing}%
      {No tikz-timing library `#1' found!}%
      {}{}{}{}%
    }%
  \@ifnextchar\relax{}{\tikztiming@library@}%
}

\def\tikztimingwidth{0.0}
\newcount\tikztiming@numint
\newcount\tikztiming@numfrac
\def\tikztiming@num{1.0}%
\let\tikztiming@back\empty
\newlength\tikztiming@xunit
\newlength\tikztiming@yunit

\newcounter{tikztiming@nrows}%
\def\tikztiming@rowdist{2}%
\def\tikztiming@coldist{1}%
\def\tikztiminglabel#1{#1}%
```

```

\def\tikztiming@prefix{tikztiming@trans@}
```

6.2 TikZ Style Settings

```
\tikzset{timing/.style={%
    x=1.6ex, y=1.6ex,
    line cap=round, line join=round,
    /utils/exec={\setlength{\tikztiming@xunit}{1.6ex}\setlength{\tikztiming@yunit}{1.6ex}},
  }%
}
\tikzset{%
  timing/.cd,
  initchar/.value required,
  initchar/.code={\ifx\lastchar\empty\uppercase{\def\lastchar{#1}}\fi},
  metachar/.code 2 args={\tikztimingmetachar{#1}{#2}},
  grid/.style={timing,help lines},
  picture/.style={timing,line width=0.15ex},
  intext/.style={timing,line width=0.15ex},
  inline node/.style={shape=coordinate},
  table/.style={timing,line width=0.15ex,font=\sffamily},
  coord/.style={coordinate},
  save/.style={coordinate,/utils/exec=\tikztiming@savesettings},
  restore/.style={/utils/exec=\tikztiming@restoresettings},
  name/.style={inner sep=0pt,outer sep=0pt},
  d/text/.style={timing,scale=0.6,font=\sffamily},
  d/background/.style={},
  h/.style={},
  l/.style={},
  d/.style={},
  e/.style={},
  e/background/.style={},
  a/.style={->},
  w/.style={<-},
  m/.style={black!40!brown},
  m/decoration/.style={decorate,decoration={zigzag,segment
  length=.25\tikztiming@xunit,amplitude=.225\tikztiming@yunit}},
  k/.style={black!40!blue,semitransparent},
  u/background/.style={fill=gray},
  u/.style={},
  o/background/.style={},
  o/.style={timing/d,line width=0.10ex,dotted},
  g/.style={},
  z/.style={blue},
  t/.style={},
  c/.style={timing/slope=0.0},
  x/.style={red},
  table/grid/.style={timing/grid},
  table/lines/.style={},
  table/rules/.style={line width=0.08em,line cap=butt},
  slope/.code={%
    \tikztimingsetslope{#1}%
    \tikztimingsetdslope{2*#1}%
    \tikztimingsetzslope{#1/2}%
  },
  lslope/.code={\tikztimingsetslope{#1}},
  dslope/.code={\tikztimingsetdslope{#1}},
  zslope/.code={\tikztimingsetzslope{#1}},
  coldist/.store in=\tikztiming@coldist,
  rowdist/.store in=\tikztiming@rowdist,
}
```

6.3 Macros

\texttimingbefore

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

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

\texttimingafter

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

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

\texttiminggrid

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

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

\texttiming

```
\DeclareRobustCommand*\texttiming[2][]{%
  \begingroup
    \tikztiming@init
    \ifx\relax#1\relax\else
      \tikztiming@testoptarg#1\relax\relax%
    \fi
    \def\@tempa{\begin{tikzpicture}[timing/intext,}%]}[
    \expandafter\@tempa\settings]
      \@ifundefined{tikztiming@initcode@\lastchar}%
        {}%
        {\@nameuse{tikztiming@initcode@\lastchar}}%
      \ifx\lastchar\empty\else
```

```
   \@ifundefined{\tikztiming@prefix\lastchar @start}%
    {\PackageWarning{tikz-timing}{Start value for timing character '\lastchar'
    is not defined and will be ignored!}{}{}{}}%
    {\tikztiming@nameaddtostr{\lastchar @start}{}}%
   \fi
   \tikztiming@#2\relax
   %\message{^^J\meaning\tikztiming@str^^J}%
   \path[use as bounding box] (0,0) rectangle
   (\timingwidth*\tikztimingwidth,\timingheight);%
   \texttimingbefore
   \tikztiming@str;%
   \node [shape=tikztiming@shape,anchor=origin,name=last texttiming] at (0,0)
   {};
   \texttimingafter
  \end{tikzpicture}%
 \endgroup
}
```

\tikztiming@testoptarg

```
\def\tikztiming@testoptarg#1#2\relax{%
 \ifx\relax#2\relax
   \uppercase{\def\lastchar{#1}}%
 \else
   \def\settings{#1#2}%
 \fi
}
```

\tikztiming@init

```
\def\tikztiming@init{%
  \let\lastchar\empty%
  \let\currentchar\empty
  \let\settings\empty
  \def\tikztimingwidth{0.0}%
  \setcounter{tikztimingtrans}{-1}%
  \setcounter{tikztimingtranspos}{0}%
  \def\tikztiming@str{\draw (0,0) coordinate (timing/start base) }%
}
```

\timing

```
\def\timing{%
 \@ifnextchar{[}%
   {\timing@}%
   {\timing@[]}%
}
```

\timing@

```
\def\timing@[#1]{%
```

```
  \@ifnextchar{a}%
    {\timing@at{#1}}%
    {\PackageWarning{tikz-timing}{The \string\timing\space command awaits now an
    ‘at’ before the coordinate, like the \string\node\space command is doing.
    Please update your source code to ensure compatibility with newer
    versions}{}{}{}%
     \@ifnextchar{+}%
      {\timing@@{#1}}%
      {\@ifnextchar(%)
        {\timing@@{#1}}%
        {\timing@@{#1}++(0,0)}%
      }%
    }%
}
```

\timing@at

```
\def\timing@at#1at#2(#3){%
  \timing@@@{#1}{#2(#3)}%
}
```

\timing@@

```
\def\timing@@#1#2(#3){%
  \timing@@@{#1}{#2(#3)}%
}
```

\timing@@@

```
\def\timing@@@#1#2#3{%
  \begingroup
    \tikztiming@init
    \let\tikz@alias=\pgfutil@empty%
    \begin{scope}[shift={#2},timing,#1]%
      \@ifnextchar{[}%
        {\timing@@@init}%
        {\timing@@@init[]}%
      #3\relax
      %\message{^^J\meaning\tikztiming@str^^J}%
      \tikztiming@str;%
      \node [shape=tikztiming@shape,anchor=origin,#1] at (0,0) {};
    \end{scope}%
  \endgroup
  \timing@@@end
}
```

\timing@@@end

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

\timing@@@init

```
\def\timing@@@init[#1]{%
  \ifx\relax#1\relax\else
    \uppercase{\def\lastchar{#1}}%
  \fi
  \@ifundefined{tikztiming@initcode@\lastchar}%
    {}%
    {\@nameuse{tikztiming@initcode@\lastchar}}%
  \ifx\lastchar\empty\else
  \@ifundefined{\tikztiming@prefix\lastchar @start}%
    {\PackageWarning{tikz-timing}{Start value for timing character '\lastchar'
    is not defined and will be ignored!}{}{}{}}%
    {\tikztiming@nameaddtostr{\lastchar @start}{}}%
  \fi
  \tikztiming@
}
```

\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.

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

\tikztiming@aftercode@T

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

\tikztiming@aftercode@t

```
\def\tikztiming@aftercode@t{%
  \tikztiming@aftercode@T
}
```

\tikztiming@aftercode@C

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

\tikztiming@aftercode@c

```
\def\tikztiming@aftercode@c{%
  \tikztiming@aftercode@C
}
```

\tikztiming@aftercode@G

```
\def\tikztiming@aftercode@G{%
  \let\lastchar\secondlastchar
  \let\tikztimingwidth\lasttikztimingwidth
}
```

\tikztiming@aftercode@g

```
\def\tikztiming@aftercode@g{%
  \let\lastchar\secondlastchar
  \let\tikztimingwidth\lasttikztimingwidth
}
```

\tikztiming@aftercode@S

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

\tikztiming@aftercode@s

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

\tikztiming@beforenextcode@D@edge@

```
\def\tikztiming@beforenextcode@D@edge@{%
  \if D\currentchar\else
    \if d\currentchar\else
      \def\lastchar{D}%
    \fi
  \fi
}
```

\tikztiming@beforecode@d@edge@

```
\def\tikztiming@beforenextcode@d@edge@{%
  \if D\currentchar\else
    \if d\currentchar\else
      \def\lastchar{D}%
    \fi
  \fi
}
```

\tikztiming@initcode@D

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

\tikztiming@initcode@d

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

\tikztiming@

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

```
\def\tikztiming@{%
  \@ifnextchar\bgroup
    {\tikztiming@testfornum}%
    {\tikztiming@testfornum}%
}
```

\tikztiming@eaddtostr

```
\def\tikztiming@eaddtostr#1{%
  \begingroup
    \tikztiming@internaldefs{}%
    \@temptokena\expandafter{\tikztiming@str}%
    \xdef\tikztiming@str{%
      \the\@temptokena
      #1%
    }%
  \endgroup
}
```

\tikztiming@addtostr

```
\def\tikztiming@addtostr{%
  \g@addto@macro\tikztiming@str
}
```

\tikztiming@output

```
\def\tikztiming@output#1#2{%
  \ifx\relax#2\relax
    \tikztiming@nameaddtostr{#1}%
  \else
    \ifcase0%
      \ifx\tikztiming@output@bufchara\empty
        \ifx\tikztiming@output@bufcharb\empty
          1%
        \fi
      \fi\relax
      % not empty
      \edef\tikztiming@output@currentchar{#2}%
      \ifcase0%
      \expandafter\ifx\csname tikztiming@nocombine@#2\endcsname\relax
      \ifx\tikztiming@output@currentchar\tikztiming@output@bufcharb
        1%
      \fi\fi
      \relax
        \tikztiming@output@flush
        \edef\tikztiming@output@bufchara{#1}%
        \edef\tikztiming@output@bufcharb{#2}%
      \or
        \pgfmathparse{\tikztiming@output@bufnum + \tikztiming@num}%
        \let\tikztiming@output@bufnum\pgfmathresult
        \def\tikztiming@num{1.0}%
      \fi
    \else % empty
      \edef\tikztiming@output@bufchara{#1}%
      \edef\tikztiming@output@bufcharb{#2}%
      \let\tikztiming@output@bufnum\tikztiming@num
      \def\tikztiming@num{1.0}%
    \fi
  \fi
}
```

Init buffer macros:

```
\def\tikztiming@output@bufchara{}%
\def\tikztiming@output@bufcharb{}%
\def\tikztiming@output@bufnum{0}%
```

\tikztiming@output@flush

```
\def\tikztiming@output@flush{%
  \begingroup
    \let\tikztiming@num\tikztiming@output@bufnum
    \tikztiming@nameaddtostr{%
      \tikztiming@output@bufchara
      \tikztiming@output@bufcharb
    }%
  \endgroup%
  \gdef\tikztiming@output@bufchara{}%
  \gdef\tikztiming@output@bufcharb{}%
  \global\let\tikztiming@output@bufnum\tikztiming@num
  \gdef\tikztiming@num{1.0}%
}
```

\tikztiming@nameaddtostr

```
\def\tikztiming@nameaddtostr#1{%
  \begingroup
    \edef\@tempa{\tikztiming@num}%
    \expandafter\g@addto@macro
    \expandafter\tikztiming@str
    \expandafter{\csname\tikztiming@prefix#1\expandafter\endcsname
    \expandafter{\@tempa} }%
  \endgroup
  \def\tikztiming@num{1.0}%
}
```

\tikztiming@nameedef

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

```
\newcommand\tikztiming@nameedef[3][A]{%
  \def\@gtempa##1{#3}%
  \expandafter\let\csname\tikztiming@prefix#2@general\endcsname\@gtempa
  \begingroup
    \tikztiming@internaldefs{#1}%
    \xdef\@gtempa##1{\@gtempa{\width}}%
  \endgroup
  \expandafter\let\csname\tikztiming@prefix#2\endcsname\@gtempa
  \let\@gtempa\empty
}
```

\tikztiming@namelet

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

```
\newcommand\tikztiming@namelet[2]{%
  \ifcase0%
    \@ifundefined{\tikztiming@prefix#2}%
      {\@ifundefined{\tikztiming@prefix#1}%
        {0}{1}%
      }%
      {1}%
    \relax
  \else
    \expandafter\let
    \csname\tikztiming@prefix#1\expandafter\endcsname
    \csname\tikztiming@prefix#2\endcsname
  \fi
}
```

\tikztiming@@end

```
\def\tikztiming@@end{%
  \tikztiming@output@flush
  \global\let\tikztimingwidth\tikztimingwidth
  \tikztiming@addtostr{ coordinate (timing/end)
    let \p1 = (timing/start base), \p2 = (timing/end) in
      coordinate (timing/end base) at (\x2,\y1)
      coordinate (timing/end top)  at (\x2,1+\y1)
  }%
}
```

\tikztiming@@

```
\def\tikztiming@@#1{%
  \ifx\relax#1\empty
    \expandafter\tikztiming@@end
  \else
    \let\lasttikztimingwidth\tikztimingwidth
    \tikztiming@iflower{#1}%
      {\pgfmathparse{\tikztiming@num/2.0}\let\tikztiming@num\pgfmathresult}%
      {}%
    \ifx\tikztiming@back\empty\else
      \pgfmathparse{\tikztiming@num-\tikztiming@back}%
      \let\tikztiming@num\pgfmathresult
      \let\tikztiming@back\empty
    \fi
    \pgfmathparse{\tikztimingwidth + \tikztiming@num}%
    \let\tikztimingwidth\pgfmathresult
    \def\currentchar{#1}%
    \uppercase{\def\currentcharuc{#1}}%
    \@ifundefined{tikztiming@beforenextcode@\lastchar}%
      {}%
      {\@nameuse{tikztiming@beforenextcode@\lastchar}}%
    \@ifundefined{tikztiming@beforecode@\currentchar}%
      {}%
      {\@nameuse{tikztiming@beforecode@\currentchar}}%
    \@ifundefined{\tikztiming@prefix\lastchar\currentchar}%
      {\@ifundefined{\tikztiming@prefix\lastchar\currentcharuc}%
        {\PackageWarning{tikz-timing}{Timing transition ’\lastchar\currentchar’
        is not defined and will be ignored!}{}{}{}}%
        {\tikztiming@output{\lastchar}{\currentcharuc}}%
      }%
      {\tikztiming@output{\lastchar}{\currentchar}}%
    \let\secondlastchar\lastchar
    \let\lastchar\currentcharuc
    \@ifundefined{tikztiming@aftercode@\currentcharuc}%
      {}%
      {\@nameuse{tikztiming@aftercode@\currentcharuc}}%
    \expandafter
    \tikztiming@testfortext
  \fi
}
```

\tikztiming@testfortext

```
\def\tikztiming@testfortext{%
  \@ifnextchar\bgroup
    {\tikztiming@handletext}%
    {\tikztiming@}%
}
```

\tikztiming@handletext

```
\def\tikztiming@handletext#1{%
  \@ifnextchar{[}%
    {\tikztiming@handletext@}%
    {\tikztiming@handletext@[]}%
  #1\relax
}
```

\tikztiming@handletext@

```
\def\tikztiming@handletext@[#1]#2\relax{%
  \begingroup
  \expandafter\lowercase\expandafter{%
    \expandafter\def\expandafter\currentcharlc
    \expandafter{\currentchar}%
  }%
  \pgfkeysifdefined{/tikz/timing/\currentcharlc/text/.@cmd}%
  {%
  \tikztiming@output@flush
  \tikztiming@eaddtostr{%
    node (timing@dend) at +(\dslope/2.0,\height/2.0) {}
    node [%]
      shift={($ (timing@dstart)!0.5!(timing@dend) $)},%
      timing/\currentcharlc/text,%
  }%
  \endgroup
  \tikztiming@addtostr{%[
      #1%
      ] {#2}%
  }%
  \def\lastchar{D@edge@}%
  }{%
    \endgroup
    \PackageWarning{tikz-timing}{Ignoring text for character
    '\currentchar'!}{}{}{}%
  }%
  \tikztiming@
}
```

\tikztiming@defcode

```
\def\tikztiming@defcode#1{%
  \@namedef{tikztiming@code@\meaning#1}%
}
```

\tikztiming@defcode,

```
\tikztiming@defcode{,}{%
  \tikztiming@output@flush
  \tikztiming@eaddtostr{%
    \newdraw
  }%
  \tikztiming@
}
```

\tikztiming@defcode;

```
\tikztiming@defcode{;}{%
  \tikztiming@output@flush
  \tikztiming@eaddtostr{%
    \newdrawns
  }%
  \tikztiming@
}
```

\tikztiming@defcodeN

```
\tikztiming@defcode{N}{%
  \@ifnextchar[%]
    {\tikztiming@addnode@getoptions}%
    {\tikztiming@addnode@getoptions[]}%
}
```

\tikztiming@defcoden

```
\tikztiming@defcode{n}{%
  \@ifnextchar[%]
    {\tikztiming@addnode@getoptions}%
    {\tikztiming@addnode@getoptions[]}%
}
```

\tikztiming@addnode

```
\def\tikztiming@addnode@getoptions[#1]{%
  \@ifnextchar(%)
    {\tikztiming@addnode@getname{#1}}%
    {\tikztiming@addnode@getname{#1}()}%
}
```

\tikztiming@addnode@getname

```
\def\tikztiming@addnode@getname#1(#2){%
  \@ifnextchar\bgroup
    {\tikztiming@addnode@{#1}{#2}}%
    {\tikztiming@addnode@{#1}{#2}{}}%
}
```

\tikztiming@addnode@

```
\def\tikztiming@addnode@#1#2#3{%
  \tikztiming@output@flush
  \begingroup
    \def\@tempa{#2}%
    \ifx\@tempa\empty
      \def\@tempa{ node [timing/inline node,#1] }%
    \else
      \def\@tempa{ node [timing/inline node,#1] (#2) }%
    \fi
    \expandafter\tikztiming@addtostr\expandafter{\@tempa {#3} }%
  \endgroup
  \tikztiming@
}
```

\tikztiming@testforcode

```
\def\tikztiming@testforcode{%
  \@ifnextchar{!}%
    {\tikztiming@testforcode@}%
    {\@ifundefined{tikztiming@code@\meaning\@let@token}%
      {\tikztiming@@}%
      {\csname tikztiming@code@\meaning\@let@token\expandafter
       \endcsname\@gobble}%
    }%
}
```

\tikztiming@testforcode@

```
\def\tikztiming@testforcode@#1{%
  \@ifnextchar\bgroup
    {\tikztiming@handlecode}%
    {%
      \PackageWarning{tikz-timing}{Missing braces after '!' character. Ignoring
      this character}{}{}{}%
      \tikztiming@
    }%
}
```

\tikztiming@defcodeB

```
\tikztiming@defcode{B}{%
  \pgfmathparse{\tikztiming@back+\tikztiming@num}%
  \let\tikztiming@back\pgfmathresult
  \tikztiming@
}
```

\tikztiming@defcodeb

```
\tikztiming@defcode{b}{%
  \pgfmathparse{\tikztiming@back+\tikztiming@num/2}%
  \let\tikztiming@back\pgfmathresult
  \tikztiming@
}
```

\tikztiming@defcodeF

```
\tikztiming@defcode{F}{%
  \pgfmathparse{\tikztiming@back-\tikztiming@num}%
  \let\tikztiming@back\pgfmathresult
  \tikztiming@
}
```

\tikztiming@defcodef

```
\tikztiming@defcode{f}{%
  \pgfmathparse{\tikztiming@back-\tikztiming@num/2}%
  \let\tikztiming@back\pgfmathresult
  \tikztiming@
}
```

```
⟨/package⟩
⟨*lib – overlays⟩
```

\tikztiming@defcodeO

```
\tikztiming@defcode{O}{%
  \@ifnextchar\relax
    {\tikztiming@\relax}%
    {\tikztiming@overlay}%
}
```

\tikztiming@overlays

```
\def\tikztiming@overlay#1#2{%
  \ifx\relax#1\relax
    \tikztiming@output@flush
    \def\next{\tikztiming@\relax}%
  \else
    \ifx\relax#2\relax
```

```
      \def\next{\tikztiming@#1\relax}%
    \else
      \tikztiming@output@flush
      \let\tikztiming@overlay@lastchar\lastchar
      \tikztiming@addtostr{ node [timing/save] (timing@overlay@start) {} }%
      \tikztiming@#1\relax
      \tikztiming@output@flush
      \let\lastchar\tikztiming@overlay@lastchar
      \tikztiming@addtostr{ (timing@overlay@start) }%
      \def\next{\tikztiming@#2}%
    \fi
  \fi
  \next
}
```

⟨/lib − overlays⟩
⟨*package⟩

\tikztiming@handlecode

```
\def\tikztiming@handlecode#1{%
  \tikztiming@output@flush
  \tikztiming@addtostr{ #1 }%
  \tikztiming@
}
```

\tikztiming@defcode []

```
\tikztiming@defcode[#1]{%
  \tikztiming@addtostr{ [#1] }%
  \tikztiming@
}
```

\tikztiming@testfornum

```
\def\tikztiming@testfornum{%
  \let\tikztiming@numchars\empty
  \tikztiming@numfrac0\relax
  \afterassignment
  \tikztiming@testfornum@
  \tikztiming@numint0%
}
```

\tikztiming@testfornumfrac

```
\def\tikztiming@testfornumfrac{%
  \afterassignment
  \tikztiming@testfornum@@@
  \tikztiming@numfrac1%
}
```

\tikztiming@numloop

```
\def\tikztiming@numloop{%
  \ifnum\tikztiming@numint>0%
    \toks@\expandafter{\tikztiming@numchars}%
    \xdef\tikztiming@numchars{%
      \the\toks@
      \the\@temptokena
    }%
    \advance\tikztiming@numint by -1\relax
    \expandafter\tikztiming@numloop
  \fi
}
```

\tikztiming@testfornum@

```
\def\tikztiming@testfornum@{%
  \ifnum0<\tikztiming@numint
    \let\tikztiming@next\tikztiming@testfornum@@
  \else
    \def\tikztiming@next{%
      \@ifnextchar{.}%
        {\expandafter\tikztiming@testfornumfrac\@gobble}%
        {%
          \tikztiming@numint1\relax
          \tikztiming@numfrac0\relax
          \def\tikztiming@num{1.0}%
          \if@tikztiming@metachar
            {\def\@tempa{\expandafter\expandafter\expandafter
            \tikztiming@testfornum@@@@
            \expandafter\expandafter\expandafter{%
            \csname tikztiming@metachar@\meaning\@let@token\endcsname}}%
            \expandafter\@tempa\@gobble
            }%
            {\tikztiming@testforcode}%
        }%
    }%
  \fi
  \tikztiming@next
}
```

\tikztiming@testfornum@@

```
\def\tikztiming@testfornum@@{%
  \@ifnextchar{.}%
    {\expandafter\tikztiming@testfornumfrac\@gobble}%
    {\tikztiming@testfornum@@@}%
}
```

\tikztiming@testfornum@@@

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

```
  \xdef\tikztiming@num{\the\tikztiming@numint.\expandafter\@gobble\the\tikztiming@numfrac}%
  \@ifnextchar\bgroup
    {%
      \expandafter\tikztiming@numfrac\expandafter0\expandafter
      \@gobble\the\tikztiming@numfrac\relax
      \ifnum0=\tikztiming@numfrac\else
        \pgfmathparse{round(\tikztiming@num)}%
        \PackageWarning{tikz-timing}%
          {Can not repeat group by a non-integer factor!^^J%
           Rounding '\tikztiming@num' to '\pgfmathresult'.}{}{}{}%
        \let\tikztiming@num\pgfmathresult
      \fi
      \tikztiming@testfornum@@@@
    }%
    {%
      \if@tikztiming@metachar
        {\def\@tempa{\expandafter\expandafter\expandafter
        \tikztiming@testfornum@@@@
        \expandafter\expandafter\expandafter{%
        \csname tikztiming@metachar@\meaning\@let@token\endcsname}}%
        \expandafter\@tempa\@gobble
        }%
        {\tikztiming@testforcode}%
    }%
}
```

\tikztimig@metachar

```
\def\tikztimingmetachar#1#2{%
  \ifx\relax#2\relax
    \expandafter\let\csname tikztiming@metachar@\meaning#1\endcsname\@undefined
  \else
    \@namedef{tikztiming@metachar@\meaning#1}{#2}%
  \fi
}
```

\if@tikztimig@metachar

```
\def\if@tikztiming@metachar#1#2{%
  \@ifundefined{tikztiming@metachar@\meaning\@let@token}{#2}{#1}%
}
```

\tikztiming@testfornum@@@@

```
\def\tikztiming@testfornum@@@@#1{%
  \begingroup
```

```
    \@temptokena{#1}%
    \tikztiming@numloop%
  \endgroup
  \tikztiming@numint1\relax
  \tikztiming@numfrac0\relax
  \expandafter\tikztiming@\tikztiming@numchars
}
```

6.4 Table environment

```
%\usetikzlibrary{backgrounds}
\newcounter{tikztimingrows}
```

\tikztiming@extracode

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

tikztimingtable

```
\newenvironment{tikztimingtable}[1][]{%
  \begingroup
  \setcounter{tikztiming@nrows}{0}%
  \def\tikztiming@maxwidth{0.0}%
  \let\extracode\tikztiming@extracode
  \let\tablegrid\tikztiming@tablegrid
  \let\fulltablegrid\tikztiming@fulltablegrid
  \let\horlines\tikztiming@horlines
  \let\vertlines\tikztiming@vertlines
  \let\tablerules\tikztiming@tablerules
  \def\rowdist{\tikztiming@rowdist}%
  \def\coldist{\tikztiming@coldist}%
  \def\nrows{\the\c@tikztiming@nrows}%
  \def\twidth{\tikztiming@maxwidth}%
  \tikzpicture[timing/table,#1]%
    \coordinate (last row)  at (0,\rowdist);
    \coordinate (label pos) at (-1*\coldist,0);
    \coordinate (timing/table/top left) at (0,1);
    \coordinate (timing/table/bottom right) at (0,0);
  \collect@body\tikztiming@table
}{%
}
```

\tikztiming@table

```
\def\tikztiming@table#1{%
  \tikztimingtable@row#1\endtikztimingtable@
  \endtikzpicture
  \endgroup
}
```

\endtikztimingtable@

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

\tikztimingextracode

```
\long\def\tikztimingextracode#1#2\endtikztimingtable@{%
  \path let
    \p1 = (timing/table/top left),
    \p2 = (timing/table/bottom right)
   in
    coordinate (timing/table/bottom left) at (\x1,\y2)
    coordinate (timing/table/top right) at (\x2,\y1)
    coordinate (timing/table/size) at (\x2-\x1,\y1-\y2)
   ;
  #2%
}
```

\tikztiming@emptycell

Just used as marker. Needs unique definition.

```
\def\tikztiming@emptycell{%
  \@gobble{tikztiming@emptycell}%
}
```

\tikztimingtable@row

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

\tikztimingtable@row@

```
\def\tikztimingtable@row@#1&#2&#3\\{%
  \ifx\\#3\\\else
    \begingroup
      \def\@tempa{\tikztiming@emptycell&}%
      \def\@tempb{#3}%
      \ifx\@tempa\@tempb\else
        \PackageWarning{tikz-timing}{%
          To many columns in tikztimingtable row! Only two are allowed%
        }{}{}{}%
      \fi
    \endgroup
  \fi
  \ifx\tikztiming@emptycell#2%
    \def\next{\tikztimingtable@row@@{#1}{}}%
  \else
    \def\next{\tikztimingtable@row@@{#1}{#2}}%
  \fi
  \next
}%
```

\tikztimingtable@row@@

```
\def\tikztimingtable@row@@#1#2{%
  \addtocounter{tikztiming@nrows}{1}%
  \path ($ (last row) + (0,-1*\rowdist) $) coordinate (last row);
  \path ($ (last row) + (-1*\coldist,0) $) node [anchor=base east,timing/name]
    {\tikztiminglabel{#1}};
  \@ifnextchar{[}%
    {\tikztiming@tabletiming}%
    {\tikztiming@tabletiming[]}%
  #2\relax
  \path let \p1 = (timing/table/bottom right), \p2 = (timing/end base) in
    coordinate (timing/table/bottom right) at ({max(\x1,\x2)},\y2);
  %
  \pgfmathparse{max(\tikztiming@maxwidth,\tikztimingwidth)}%
  \let\tikztiming@maxwidth\pgfmathresult
  \@ifnextchar\extracode
    {%
      \let\extracode\relax
      \tikztimingextracode
    }%
    {%
      \@ifnextchar\endtikztimingtable@
        {\@gobble}{\tikztimingtable@row}%
    }%
}
```

\tikztiming@tabletiming

```
\def\tikztiming@tabletiming[#1]#2\relax{%
  \let\lastchar\empty
  \let\settings\empty
  \ifx\relax#1\relax\else
    \tikztiming@testoptarg#1\relax\relax
  \fi
  \edef\@tempa{\noexpand\timing[name=row\the\c@tikztiming@nrows,\settings]
    at (last row)}%
  \expandafter\@tempa\expandafter{\expandafter[\lastchar]#2};
}
```

\tikztiming@fulltablegrid

```
\newcommand*\tikztiming@fulltablegrid[1][]{%
```

```
  \begin{pgfonlayer}{background}
    \scope[xstep={\timingwidth/2.},ystep={\timingheight/2.},
    shift={(timing/table/bottom left)},timing/table/grid,#1]
      \draw (0,0) grid
        ($ (timing/table/top right) - (timing/table/bottom left) $);
   \endscope
  \end{pgfonlayer}
}
```

\tikztiming@tablegrid

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

\tikztiming@tablerules

```
\newcommand*\tikztiming@tablerules[1][]{%
    \draw [timing/table/rules,#1] let
      \p1 = (current bounding box.north west),
      \p2 = (current bounding box.south east),
      \p3 = (last row)
    in
      (\x1-\tabcolsep,\rowdist) -- (\x2+\tabcolsep,\rowdist)
      ($ (\x1-\tabcolsep,\y3) - (0,\rowdist-1) $) --
      ($ (\x2+\tabcolsep,\y3) - (0,\rowdist-1) $)
    ;
}
```

\tikztiming@horlines

```
\newcommand*\tikztiming@horlines[2][]{%
  \begingroup
    \def\list{#2}%
    \ifx\list\empty
      \def\list{1,2,...,\nrows}%
    \fi
    \foreach \row in \list%
      \draw [timing/table/lines,#1] let
```

```
        \p1 = (timing/table/bottom right)
      in
        (0,\rowdist-\row*\rowdist) -- +(\x1,0);
  \endgroup
}
```

\tikztiming@vertlines

```
\newcommand*\tikztiming@vertlines[2][]{%
  \begingroup
    \def\list{#2}%
    \ifx\list\empty
      \def\list{0,1,...,\twidth}%
    \fi
    \draw [timing/table/lines,#1] let
        \p1 = ($ (timing/table/bottom right) - (0,2) $)
      in
        \foreach \clk in \list {
          (\clk,+1.5) -- +(0,\y1)
        }
    ;
  \endgroup
}
```

6.5 Shape

```
\pgfdeclareshape{tikztiming@shape}{%
  \savedanchor\northeast{\pgfpointxy{.5*\tikztimingwidth}{.5}}%
  \savedanchor\northeastborder{\pgfpointxy{.5*\tikztimingwidth+0.1}{.6}}%
  \savedanchor\startpoint{%
    \pgfpointanchor{timing/start}{center}%
    \@tempdima=\pgf@x
    \@tempdimb=\pgf@y
    \pgfpointxy{-.5*\tikztimingwidth}{-.5}%
    \advance\pgf@x by \@tempdima
    \advance\pgf@y by \@tempdimb
  }%
  \savedanchor\endpoint{%
    \pgfpointanchor{timing/end}{center}%
    \@tempdima=\pgf@x
    \@tempdimb=\pgf@y
    \pgfpointxy{-.5*\tikztimingwidth}{-.5}%
    \advance\pgf@x by \@tempdima
    \advance\pgf@y by \@tempdimb
  }%
  \anchor{center}{\pgfpointorigin}%
  \anchor{start}{\startpoint}%
  \anchor{end}{\endpoint}%
  %
  \anchor{origin}{\northeast \pgf@x=-\pgf@x \pgf@y=-\pgf@y }%
  \anchor{east}{\northeast \pgf@y=0pt }%
  \anchor{west}{\northeast \pgf@y=0pt \pgf@x=-\pgf@x }%
  \anchor{north}{\northeast \pgf@x=0pt }%
  \anchor{north west}{\northeast \pgf@x=-\pgf@x }%
  \anchor{north east}{\northeast}
  \anchor{high mid}{\northeast \pgf@x=0pt }%
  \anchor{high start}{\northeast \pgf@x=-\pgf@x }%
  \anchor{high end}{\northeast}
  \anchor{south}{\northeast \pgf@x=0pt \pgf@y=-\pgf@y }%
  \anchor{south west}{\northeast \pgf@x=-\pgf@x \pgf@y=-\pgf@y }%
  \anchor{south east}{\northeast \pgf@y=-\pgf@y }%
  \anchor{low mid}{\northeast \pgf@x=0pt \pgf@y=-\pgf@y }%
  \anchor{low start}{\northeast \pgf@x=-\pgf@x \pgf@y=-\pgf@y }%
  \anchor{low end}{\northeast \pgf@y=-\pgf@y }%
  \anchor{mid}{\pgfpointorigin}%
  \anchor{mid east}{\northeast \pgf@y=0pt }%
  \anchor{mid west}{\northeast \pgf@y=0pt \pgf@x=-\pgf@x }%
  \anchor{mid end}{\northeast \pgf@y=0pt }%
  \anchor{mid start}{\northeast \pgf@y=0pt \pgf@x=-\pgf@x }%
  \anchor{base}{\northeast \pgf@x=0pt \pgf@y=-\pgf@y }%
  \anchor{base west}{\northeast \pgf@x=-\pgf@x \pgf@y=-\pgf@y }%
  \anchor{base east}{\northeast \pgf@y=-\pgf@y }%
  \anchorborder{%
    \@tempdima=\pgf@x
    \@tempdimb=\pgf@y
    \pgfpointborderrectangle{\pgfpoint{\@tempdima}{\@tempdimb}}{\northeastborder}%
  }%
}
```

6.6 Other Macros

\tikztiming@iflower

```
\def\tikztiming@iflower#1{%
  \begingroup
  \edef\@tempa{‘#1}%
  \ifnum\@tempa=\lccode\@tempa
    \endgroup
    \expandafter
    \@firstoftwo
  \else
    \endgroup
    \expandafter
    \@secondoftwo
  \fi
}
```

\timingwidth

\timingheight

```
\def\timingwidth{1}%
\def\timingheight{1}%
```

\tikztiming@internaldefs

```
\def\tikztiming@internaldefs#1{%
  \def\draw{\noexpand\draw}%
  \def\path{\noexpand\path}%
  \def\fill{\noexpand\fill}%
  \def\width{####1*\noexpand\timingwidth}%
  \def\fwidth{\noexpand\timingwidth}%
  \def\height{\noexpand\timingheight}%
  \def\slope{\noexpand\timingslope}%
  \def\zslope{\noexpand\timingzslope}%
  \def\dslope{\noexpand\timingdslope}%
  \def\gslope{0}%
  \lowercase{%
  \def\style{timing/#1}%
  \def\bgstyle{timing/#1/background}%
  }%
  \def\newdraw{\tikztiming@newdraw}%
  \def\newdrawns{\tikztiming@newdraw@nosave}%
  \def\code##1{ [/utils/exec={\unexpanded{##1}}] }%
}
```

\tikztimingsetslope

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

\tikztimingsetdslope

```
\def\tikztimingsetdslope#1{%
  \pgfmathparse{min(1.0,{max(0.0,#1)})}%
  \let\tikztiming@dslope\pgfmathresult
  \edef\timingdslope{\tikztiming@dslope*\noexpand\timingwidth}%
}
```

\tikztimingsetzslope

```
\def\tikztimingsetzslope#1{%
```

```
  \pgfmathparse{min(1.0,{max(0.0,#1)})}%
  \let\tikztiming@zslope\pgfmathresult
  \edef\timingzslope{\tikztiming@zslope*\noexpand\timingwidth}%
}
```

```
\tikztimingsetslope{0.10}%
\tikztimingsetdslope{0.20}%
\tikztimingsetzslope{0.05}%
```

\tikztiminguse

```
\def\tikztiminguse#1{%
  \@ifundefined{\tikztiming@prefix#1@general}%
    {\PackageWarning{Can not use transition macro for '#1'.}{}{}{}}%
    {\@nameuse{\tikztiming@prefix#1@general}}%
}
```

\tikztimingdef

```
\def\tikztimingdef#1{%
  \tikztimingdef@#1\relax%
}
```

\tikztimingdef@

```
\def\tikztimingdef@#1#2\relax#3{%
  \ifx\relax#2\relax
    \tikztiming@nameedef[#1]{#1}{#3}%
  \else
    \tikztiming@nameedef[#2]{#1#2}{#3}%
  \fi
}
```

\tikztiminglet

```
\def\tikztiminglet#1#2{%
  \tikztiminglet@#1\relax#2\relax
}
```

\tikztiminglet@

```
\def\tikztiminglet@#1#2\relax#3#4\relax{%
  \tikztiming@namelet{#1#2}{#3#4}%
  \tikztiming@namelet{#1#2@general}{#3#4@general}%
  \tikztiming@iflower{#1}{}%
    {\tikztiming@iflower{#2}%
      {%
        \lowercase{\tikztiminglet@{#1}{#2}\relax{#3}{#4}}\relax
      }%
      {%
        \uppercase{\lowercase{%
        \uppercase{\lowercase{\tikztiminglet@{#1}}{#2}}\relax{#3}}{#4}}\relax
        \lowercase{\uppercase{%
        \lowercase{\uppercase{\tikztiminglet@{#1}}{#2}}\relax{#3}}{#4}}\relax
      }%
    }%
}
```

\tikztiming@chars

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

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

\tikztiming@ifcharexists

```
\def\tikztiming@ifcharexists#1{%
  \def\tikztiming@ifcharexists@##1,#1,##2\relax{%
    \ifx\relax##2\relax%
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
  }%
  \expandafter\tikztiming@ifcharexists@
  \expandafter,\tikztiming@chars,#1,\relax%
}
```

\tikztiming@addchar

```
\def\tikztiming@addchar#1{%
  \tikztiming@ifcharexists{#1}{%
    \edef\tikztiming@chars{\tikztiming@chars,#1}%
  }{}%
}
```

\tikztimingchar

```
\def\tikztimingchar#1{%
  \uppercase{%
  \tikztiming@addchar{#1}%
  \tikztimingchar@{#1}}%
}
```

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

\tikztimingchar@

```
\def\tikztimingchar@#1#2#3{%
  \tikztiming@nameedef[#1]{#1@start}{#2 coordinate (timing/start) }%
  \tikztiming@nameedef[#1]{#1}{#2 coordinate (timing/start) #3}%
  \tikztimingdef{#1#1}{#3}%
}
```

\tikztimingalias

```
\def\tikztimingalias#1#2{%
  \uppercase{\tikztimingalias@{#1}{#2}}%
}
```

\tikztimingalias@

```
\def\tikztimingalias@#1#2{%
  \tikztiming@namelet{#1}{#2}%
  \tikztiming@namelet{#1@start}{#2@start}%
  \lowercase{%
  \tikztiming@namelet{#1}{#2}%
  \tikztiming@namelet{#1@start}{#2@start}%
  }%
  \tikztiminglet{#1#1}{#2#2}%
  \@for\@tempa:=\tikztiming@chars\do{%
    \expandafter\tikztiminglet@@
    \expandafter{\@tempa}{#1}{#2}%
  }%
}
```

\tikztimingecopy

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

\tikztimingecopy@

```
\def\tikztimingecopy@#1#2{%
  \tikztimingchar{#1}{}{}%
  \tikztimingdef{#1}{\tikztiminguse{#2}{##1}}%
  \tikztiming@nameedef[#1]{#1@start}{\tikztiminguse{#2@start}{##1}}%
  \lowercase{%
    \@ifundefined{\tikztiming@prefix#2}{}{%
      \tikztimingdef{#1}{\tikztiminguse{#2}{##1}}%
      \tikztiming@nameedef[#1]{#1@start}{\tikztiminguse{#2@start}{##1}}%
    }%
  }%
  \tikztimingdef{#1#1}{\tikztiminguse{#2#2}{##1}}%
  \@for\@tempa:=\tikztiming@chars\do{%
    \expandafter\tikztimingdef@@
    \expandafter{\@tempa}{#1}{#2}%
```

Handle lowercase macros:

```
    \expandafter\lowercase\expandafter{\expandafter\def\expandafter\@tempb
    \expandafter{\@tempa}}%
    \@ifundefined{\tikztiming@prefix#2\@tempb}{}{%
      \expandafter\tikztimingdef@@
      \expandafter{\@tempb}{#1}{#2}%
    }%
  }%
}
```

\tikztiminglet@@

```
\def\tikztiminglet@@#1#2#3{%
  \tikztiminglet@@@#1#2#3%
  % Should stay, cause no harm:
  \lowercase{\tikztiminglet@@@#1}#2#3%
  \lowercase{\tikztiminglet@@@#1#2#3}%
  \lowercase{\uppercase{\tikztiminglet@@@#1}#2#3}%
}
```

\tikztiminglet@@@

```
\def\tikztiminglet@@@#1#2#3{%
  \tikztiminglet{#1#2}{#1#3}%
  \tikztiminglet{#2#1}{#3#1}%
}
```

\tikztimingdef@@

```
\def\tikztimingdef@@#1#2#3{%
  \tikztimingdef{#1#2}{\tikztiminguse{#1#3}{##1}}%
  \tikztimingdef{#2#1}{\tikztiminguse{#3#1}{##1}}%
}
```

\tikztiming@savesettings

```
\def\tikztiming@savesettings{%
  \xdef\tikztiming@saved@settings{%
    {\tikztiming@slope}%
    {\tikztiming@dslope}%
    {\tikztiming@zslope}%
    {\the\pgflinewidth}%
  }%
}
```

\tikztiming@restoresettings

```
\def\tikztiming@restoresettings{%
  \expandafter\tikztiming@restoresettings@
  \tikztiming@saved@settings\relax
}
```

\tikztiming@restoresettings@

```
\def\tikztiming@restoresettings@#1#2#3#4\relax{%
  \tikztimingsetslope{#1}%
  \tikztimingsetdslope{#2}%
  \tikztimingsetzslope{#3}%
  \pgfsetlinewidth{#4}%
}
```

\tikztiming@newdraw

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

\tikztiming@newdraw

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

6.7 Definition of Timing Characters

```
\tikztimingchar{H}{++(0,\height)}{-- ++(#1,0)}

\tikztimingchar{L}{++(0,0)}{-- ++(#1,0)}

\tikztimingchar{Z}{++(0,\height/2.)}{%
  \newdraw [\style]
  -- ++(#1,0)
}

\tikztimingchar{X}{}{}%
\tikztimingchar{D}{}{}%
\tikztimingchar{U}{}{}%
%\tikztimingchar{O}{}{}%
\tikztimingchar{M}{}{}%

\tikztimingchar{G}{++(0,0)}{-- ++(\gslope,\height) -- ++(\gslope,-\height)}
\tikztimingchar{S}{++(0,0)}{++(#1,0)}

\tikztimingdef{DD}{
  node [timing/save] (timing@save) {}; \path [\bgstyle] (timing@save) ++(0,0)
     +(0.5*\dslope,0.5*\height) -- +(\dslope,0)
  -- +(#1,0)
  -- +($ (#1,0) + 0.5*(\dslope,\height) $)
  -- +(#1,\height)
  -- +(\dslope,\height) -- cycle;
  \draw [timing/restore,\style] (timing@save) ++(0,0)
  node [timing/save] (timing@dstart) at +(\dslope/2.,\height/2.) {}
  --  +(\dslope,+\height) --  +(#1,+\height) ++(0,+\height)
  --  +(\dslope,-\height) -- ++(#1,-\height)
}
\tikztiming@namelet{D@edge@D}{DD}
\tikztiming@namelet{D@edge@D@general}{DD@general}

\tikztimingchar{D}{++(0,0)}{
  node [timing/save] (timing@save) {}; \path [\bgstyle] (timing@save) ++(0,0)
  -- +(#1,0)
  -- +($ (#1,0) + 0.5*(\dslope,\height) $)
  -- +(#1,\height)
  -- +(0,\height)
  -- cycle;
  \draw [timing/restore,\style] (timing@save) ++(0,0)
  node [timing/save] (timing@dstart) at +(-\dslope/2.,\height/2.0) {}
  --  +(#1,0) ++(0,+\height)
  -- ++(#1,0) ++(0,-\height)
}

\tikztimingdef{DD}{
  node [timing/save] (timing@save) {}; \path [\bgstyle] (timing@save) ++(0,0)
  -- +(#1,0)
  -- +($ (#1,0) + 0.5*(\dslope,\height) $)
  -- +(#1,\height)
  -- +(0,\height)
  -- cycle;
  \newdraw [\style]
  --  +(#1,0) ++(0,+\height)
  -- ++(#1,0) ++(0,-\height)
}

\tikztiming@namelet{D@edge@@start}{D@start}
\tikztiming@namelet{d@edge@@start}{d@start}

```

\tikztiming@trans@D@fill

```
\def\tikztiming@trans@D@fill#1#2{%
  node [timing/save] (timing@save) {}; \path [\bgstyle] (timing@save) ++(0,0)
  -- +(0.5*\dslope,-0.5*\height)
  -- ++($ (#1,-0.5*\height) - (#2,0) $)
  -- +(0.5*\dslope,0.5*\height)
  -- +(0,\height)
  -- ++($ (#2,\height) - (#1,0) + (0.5*\dslope,0) $)
  -- cycle;
  \draw [timing/restore,\style] (timing@save) ++(0,0)
  node [timing/save] (timing@dstart) {}
}
\tikztimingdef{HH}{-- ++(#1,0)}
\tikztimingdef{LL}{-- ++(#1,0)}
\tikztimingdef{HL}{-- ++(\slope,-\height) \tikztiminguse{HH}{#1-\slope}}
\tikztimingdef{LH}{-- ++(\slope, \height) \tikztiminguse{LL}{#1-\slope}}

\tikztimingdef{HG}{-- ++(\gslope,-\height) -- ++(\gslope,+\height)}
\tikztimingdef{LG}{-- ++(\gslope,+\height) -- ++(\gslope,-\height)}
\tikztimingdef{ZG}{
  -- ++(\gslope,-\height/2.0)
  -- ++(\gslope,+\height)
  -- ++(\gslope,-\height/2.0)
}
\tikztiminglet{DG}{LG}

```

```
\tikztiminglet{HS}{S}
\tikztiminglet{LS}{S}
\tikztiminglet{ZS}{S}
\tikztiminglet{DS}{S}
\tikztiminglet{TS}{S}

\tikztimingdef{LZ}{
  \newdraw [\style]
  -- ++(\zslope,+\height/2.) -- ++($ (#1,0) - (\zslope,0) $)
}
\tikztimingdef{HZ}{%
  \newdraw [\style]
  -- ++(\zslope,-\height/2.) -- ++($ (#1,0) - (\zslope,0) $)
}
\tikztimingdef{ZH}{
    \newdraw
  -- ++(\zslope,+\height/2.) -- ++($ (#1,0) - (\zslope,0) $)
}
\tikztimingdef{ZL}{%
  \newdraw
  -- ++(\zslope,-\height/2.) -- ++($ (#1,0) - (\zslope,0) $)
}

\tikztimingdef{DZ}{
  -- ++( \dslope/2.,+\height/2.)
     ++(-\dslope/2.,+\height/2.)
  -- ++( \dslope/2.,-\height/2.)
  \newdraw [\style]
  -- ++($ (#1,0) - (\dslope/2.,0) $)
}

\tikztimingdef{ZD}{
  \tikztiming@trans@D@fill{#1}{0}%
  -- ++(\dslope/2.,\height/2.)  -- ++($ (#1,0) - (\dslope/2.,0) $)
     ++($ -1*(#1,0) + (0,-\height/2.) $)
  -- ++(\dslope/2.,-\height/2.) -- ++($ (#1,0) - (\dslope/2.,0) $)
}

\tikztimingdef{LD}{
  -- ++(0.5*\dslope,0.5*\height)
  \tikztiming@trans@D@fill{#1}{0.5*\dslope}%
  -- ++(0.5*\dslope,0.5*\height)
  -- ++($ (#1,0) - (\dslope,0) $)
     ++($ -1*(#1,0) + (0,-\height) $)          ++(\dslope/2.,+\height/2.)
  -- ++(\dslope/2.,-\height/2.) -- ++($ (#1,0) - (\dslope,0) $)
}

\tikztimingdef{DL}{
  -- ++( \dslope/2.,+\height/2.)
     ++(-\dslope/2.,+\height/2.)
  -- ++(\dslope/2.,-\height/2.)
  \newdraw [\style]
  -- ++(\dslope/2.,-\height/2.)
  -- ++($ (#1,0) - (\dslope,0) $)
}

\tikztimingdef{HD}{
  -- ++(0.5*\dslope,-0.5*\height)
  \tikztiming@trans@D@fill{#1}{0.5*\dslope}%
  -- ++(0.5*\dslope,-0.5*\height)
  -- ++($ (#1,0) - (\dslope,0) $)
     ++($ -1*(#1,0) + (0,+\height) $)          ++(\dslope/2.,-\height/2.)
  -- ++(\dslope/2.,+\height/2.) -- ++($ (#1,0) - (\dslope,0) $)
     ++(0,-\height)
}

\tikztimingdef{DH}{
     ++(0,+\height)
  -- ++(+\dslope/2.,-\height/2.)
     ++(-\dslope/2.,-\height/2.)
  -- ++(\dslope/2.,+\height/2.)
  \newdraw [\style]
  -- ++(\dslope/2.,+\height/2.)
  -- ++($ (#1,0) - (\dslope,0) $)
}


\tikztimingalias{M}{Z}
\tikztimingchar{M}{++(0,\height/2.)}{
  \newdraw [\style/decoration,\style]
  -- ++(#1,0)
}
\tikztimingdef{MG}{
  \newdraw [timing/m]
  -- ++(\gslope,-\height/2.0)
  -- ++(\gslope,+\height)
  -- ++(\gslope,-\height/2.0)
}

\tikztimingdef{MZ}{
  \newdraw [\style]
  \tikztiminguse{ZZ}{#1}
}

\tikztimingdef{ZM}{
  \newdraw [\style]
  \tikztiminguse{MM}{#1}
}

\tikztimingdef{LM}{
  \newdraw [\style]
  -- ++($ (1/8,0) + (0,\height/2) $)
  \newdraw [\style/decoration,\style]
  -- ++($ (-1/8,0) + (#1,0) $)
}

\tikztimingdef{HM}{
  \newdraw [\style]
  -- ++($ (1/8,0) + (0,-1*\height/2) $)
  \newdraw [\style/decoration,\style]
  -- ++($ (-1/8,0) + (#1,0) $)
}

\tikztimingdef{DM}{
  --  +($  (1/8,0) + (0,\height*.50) $)
      +(0,\height)
  -- ++($  (1/8,0) + (0,\height*.50) $)
  \newdraw [\style/decoration,\style]
  -- ++($ (-1/8,0) + (#1,0) $)
}

\newcounter{tikztimingtrans}
\newcounter{tikztimingtranspos}

\tikztimingchar{T}{++(0,0)}{
  -- ++(#1,0)
}

\tikztimingdef{HT}{%
  {[\style]
  \code{\setcounter{tikztimingtrans}{-1}}
  -- ++(\slope,\value{tikztimingtrans}*\height) -- ++($ (#1,0) - (\slope,0) $)
  }
}

\tikztimingdef{LT}{%
  {[\style]
  \code{\setcounter{tikztimingtrans}{+1}}
  -- ++(\slope,\value{tikztimingtrans}*\height) -- ++($ (#1,0) - (\slope,0) $)
  }
}

\tikztimingdef{TL}{%
  \code{\setcounter{tikztimingtranspos}{\value{tikztimingtrans}}%
  \addtocounter{tikztimingtranspos}{+1}}
  -- ++(\slope, -0.5*\value{tikztimingtranspos}*\height) -- ++($ (#1,0) - (\slope,0) $)
}

\tikztimingdef{TH}{%
  \code{\setcounter{tikztimingtranspos}{\value{tikztimingtrans}}%
  \addtocounter{tikztimingtranspos}{-1}}
  -- ++(\slope, -0.5*\value{tikztimingtranspos}*\height) -- ++($ (#1,0) - (\slope,0) $)
}

\tikztimingdef{TZ}{%
  \newdraw [\style]
  \code{\setcounter{tikztimingtrans}{-\value{tikztimingtrans}}}
  -- ++(\slope,\value{tikztimingtrans}*\height/2.)
  -- ++($ (#1,0) - (\slope,0) $)
}

\tikztimingdef{TG}{%
  -- +(\gslope,-1*\value{tikztimingtrans}*\height)
  -- +(\gslope,0)
}

\tikztimingdef{ZT}{%
  \newdraw
  {[\style]
  \code{\setcounter{tikztimingtrans}{-\value{tikztimingtrans}}}
  -- ++(\slope,\value{tikztimingtrans}*\height/2.)
  -- ++($ (#1,0) - (\slope,0) $)
  }
}

\tikztimingdef{TT}{%
  {[\style]
  \code{\setcounter{tikztimingtrans}{-\value{tikztimingtrans}}}
  -- ++(\slope,\value{tikztimingtrans}*\height) -- ++($ (#1,0) - (\slope,0) $)
  }
}

\tikztimingdef{TD}{
  \code{\setcounter{tikztimingtrans}{-\value{tikztimingtrans}}}
  \code{\setcounter{tikztimingtranspos}{\value{tikztimingtrans}}%
  \addtocounter{tikztimingtranspos}{-1}}
  -- ++(0.5*\dslope,+0.5*\value{tikztimingtrans} * \height)
  \tikztiming@trans@D@fill{#1}{0.5*\dslope}%
  -- ++(0.5*\dslope,+0.5*\value{tikztimingtrans} * \height)
  -- ++($ (#1,0) - (\dslope,0) $)
     ++($ -1*(#1,\value{tikztimingtrans}*\height) $)
     ++(\dslope/2.,+1*\value{tikztimingtrans}*\height/2.)
  -- ++(\dslope/2.,-1*\value{tikztimingtrans}*\height/2.)
  -- ++($ (#1,0) - (\dslope,0) $)
     ++(0,\value{tikztimingtranspos}*\height/2.)
}

\tikztimingdef{DT}{
  \code{\setcounter{tikztimingtrans}{-1}}
  \tikztiminguse{DL}{#1}%
```

```
}

\tikztimingdef{MT}{%
  \newdraw
  {[\style]
  -- ++(\slope,\value{tikztimingtrans}*\height/2.) -- ++($ (#1,0) - (\slope,0) $)
  }
}

\tikztimingdef{TM}{%
  \newdraw [\style]
  \code{\setcounter{tikztimingtrans}{-\value{tikztimingtrans}}}
  -- ++($ (1/8,0) + (0,\value{tikztimingtrans}*\height/2) $)
  \newdraw [\style/decoration,\style]
  -- ++($ (-1/8,0) + (#1,0) $)
}

\let\tikztiming@chars@default\tikztiming@chars

```

⟨/package⟩

⟨*lib – either⟩

```

\tikztimingecopy{E}{D}
\tikztimingchar{E}{
  ++(0,0)
  \code{\setcounter{tikztimingtranspos}{0}}%
  \code{\setcounter{tikztimingtrans}{1}}%
}{
  \code{\setcounter{tikztimingtranspos}{0}}%
  \code{\setcounter{tikztimingtrans}{1}}%
  \tikztiminguse{D}{#1}%
}
\tikztimingdef{LE}{%
  -- ++(#1,0)%
  \code{\setcounter{tikztimingtranspos}{0}}%
  \code{\setcounter{tikztimingtrans}{1}}%
     ++($ -1*(#1,0) $)%
  -- ++(\slope,\height)%
  -- ++($ (#1,0) - (\slope,0) $)%
     ++(0,-\height)%
}%
\tikztimingdef{HE}{%
  -- ++(#1,0)%
  \code{\setcounter{tikztimingtranspos}{1}}%
  \code{\setcounter{tikztimingtrans}{-1}}%
     ++($ -1*(#1,0) $)%
  -- ++(\slope,-\height)%
  -- ++($ (#1,0) - (\slope,0) $)%
}%
```

```
\tikztimingdef{EE}{%
    ++(0,      \height * \value{tikztimingtranspos})%
    ++(#1,0)%
  -- ++($ -1*(#1,0) $)%
  -- ++(\slope, \height * \value{tikztimingtrans})%
  -- ++(-\slope,0)%
  -- ++(#1,0)%
    ++(0, -\height + \height * \value{tikztimingtranspos})%
}%
\tikztimingdef{EH}{%
    ++(0,+\height)%
  -- ++(\slope,0)%
    ++(-\slope,-\height)%
  -- ++(\slope/2.,+\height/2.)%
  \newdraw [\style]%
  -- ++(\slope/2.,+\height/2.)%
  -- ++($ (#1,0) - (\slope,0) $)%
}%
\tikztimingdef{EL}{%
  -- ++(\slope,0)%
    ++(-\slope,\height)%
  -- ++(\slope/2.,-\height/2.)%
  \newdraw [\style]%
  -- ++(\slope/2.,-\height/2.)%
  -- ++($ (#1,0) - (\slope,0) $)%
}%
\tikztimingdef{ZE}{%
  \code{\setcounter{tikztimingtranspos}{1}}%
  \code{\setcounter{tikztimingtrans}{-1}}%
  \tikztiminguse{ZD}{#1}%
}
\tikztimingdef{XE}{%
  \code{\setcounter{tikztimingtranspos}{1}}%
  \code{\setcounter{tikztimingtrans}{-1}}%
  \tikztiminguse{XD}{#1}%
}
\tikztimingdef{ME}{%
  \code{\setcounter{tikztimingtranspos}{1}}%
  \code{\setcounter{tikztimingtrans}{-1}}%
  \tikztiminguse{MD}{#1}%
}
\def\tikztiming@nocombine@E{}%%
\def\tikztiming@nocombine@e{}%%
⟨/lib − either⟩
⟨*package⟩

\tikztimingecopy{C}{T}
\def\tikztiming@nocombine@T{}%
\def\tikztiming@nocombine@C{}%
```

```
\def\tikztiming@nocombine@t{}%
\def\tikztiming@nocombine@c{}%
\def\tikztiming@nocombine@M{}%
\def\tikztiming@nocombine@m{}%
\def\tikztiming@nocombine@W{}%
\def\tikztiming@nocombine@w{}%

\tikztimingecopy{U}{D}
\tikztimingdef{UD}{\tikztiminguse{D@edge@D}{#1}}
\tikztimingdef{DU}{\tikztiminguse{D@edge@D}{#1}}

%\tikztimingecopy{O}{D}
\tikztimingecopy{X}{Z}

```

⟨/package⟩
⟨*lib − arrows⟩

```
\tikztimingchar{A}{++(0,0)}{%
  \newdraw [\style] -- ++(0,\height)
  \newdraw ++(#1,-\height)
}
\def\tikztiming@nocombine@A{}%
\def\tikztiming@nocombine@a{}%
\tikztiminglet{AS}{SS}
\tikztiminglet{AH}{H}
\tikztiminglet{AZ}{Z}
\tikztimingdef{ZA}{
  ++(0,-0.5*\height)
  \tikztiminguse{A}{#1}
}
\tikztimingdef{HA}{
  ++(0,-\height)
  \tikztiminguse{A}{#1}
}
\tikztiminglet{XA}{ZA}
\tikztiminglet{MA}{ZA}
\tikztiminglet{AX}{AZ}
\tikztiminglet{AM}{AZ}
\tikztiminglet{LA}{A}
\tikztiminglet{AL}{LL}
\tikztiminglet{AD}{DD}
\tikztiminglet{AE}{LE}
\tikztiminglet{AU}{UU}
\tikztiminglet{SA}{A}
\tikztiminglet{AG}{G}
\tikztiminglet{AT}{LT}
\tikztiminglet{AC}{LC}
\tikztiminglet{DA}{A}
\tikztiminglet{UA}{A}
\tikztiminglet{TA}{A}
```

```
\tikztiminglet{CA}{A}
\tikztiminglet{EA}{A}

\tikztimingecopy{W}{A}
⟨/lib − arrows⟩
⟨*package⟩
```

Change History

v0.3

General: First released version . . 1

v0.4

General: Added output routine which combines successive occurrences of the same character. This improves screen display quality and reduces rendering time and file size. 1

Added parser for rows in {tikztimingtable}. This makes the syntax much more stable. Also replaced row counter with TikZ coordinates which is more user-friendly. . . 1

In-line Nodes, e.g. to mark positions inside the diagram. 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

v0.4a

General: Added \tablerules macro. Changed default style of inline nodes to coordinate. 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 \timing syntax to include an 'at' before the coordinate. Bug fix for use with the 'calc' package. 1

v0.6

General: Added "forward" modifier 'F' as reverse version of the "backward" modifier 'B'. Added support for lower-case modifiers "b','f' and n'. Added new optional characters 'A'/'W' for arrows and 'E' for uncertain low-to-high and high-to-low transitions. 1

v0.6a

General: Added library for overlay modifier 'O'. 1