Characterizing Minimum Period and Minimum Pulse Width Using Liberate MX

Product Version 14.1
December 16, 2014

Copyright Statement

Contents

Purpose

The characterization of minimum period and minimum pulse width arcs are amongst the most complicated arcs in memory characterization. Each is made up of multiple components that must all be characterized with the maximum value being stored in the library file.

Audience

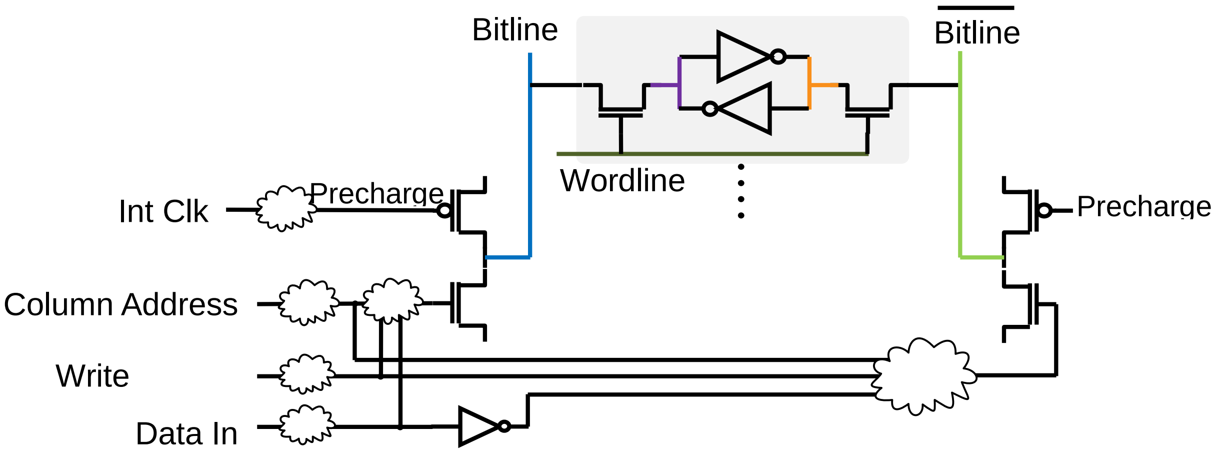
This document is intended for Application Engineers and customers characterizing memory instances using Liberate MX. It is expected that these users have a basic familiarity with Liberate MX, its commands and flow. It is also expected that the user has a basic familiarity of the internal operation of the memory instance.

Defining Minimum Period and Minimum Pulse Width

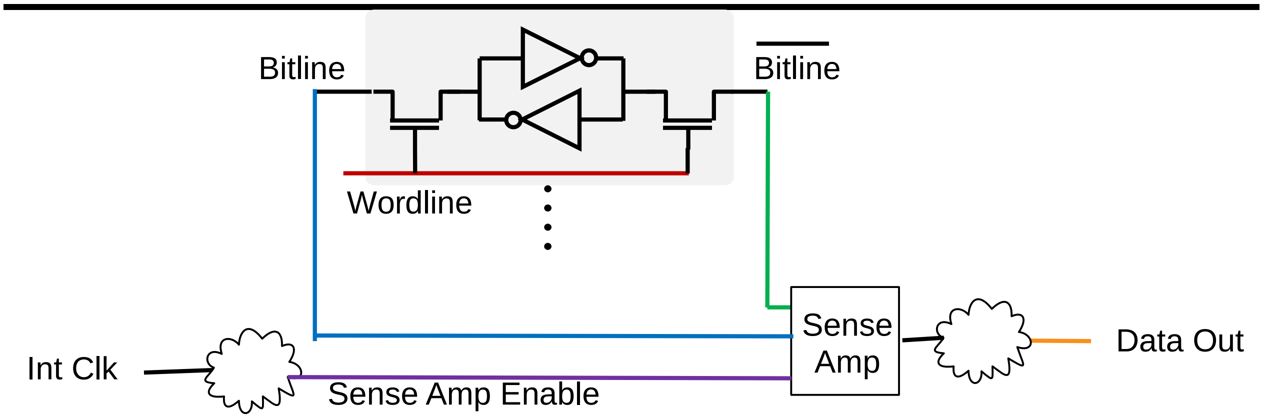
The minimum period and minimum pulse width arcs are critical arcs that are usually associated with the clock of a memory instance. The minimum period defines the total duration of the clock period from one edge to the next similar edge. For example, a minimum period arc can define the duration between a rising edge of the clock and the subsequent rising edge. The minimum pulse width defines the minimum time between an edge and the subsequent edge in opposite direction. In this application note, it is assumed that the clock rising edge starts the cycle.

Memory Clocking Schemes

The method of internal clocking of the memory instance has a significant effect on the methods and components of the characterization of these arcs. There are two primary clocking schemes seen in memory instances.

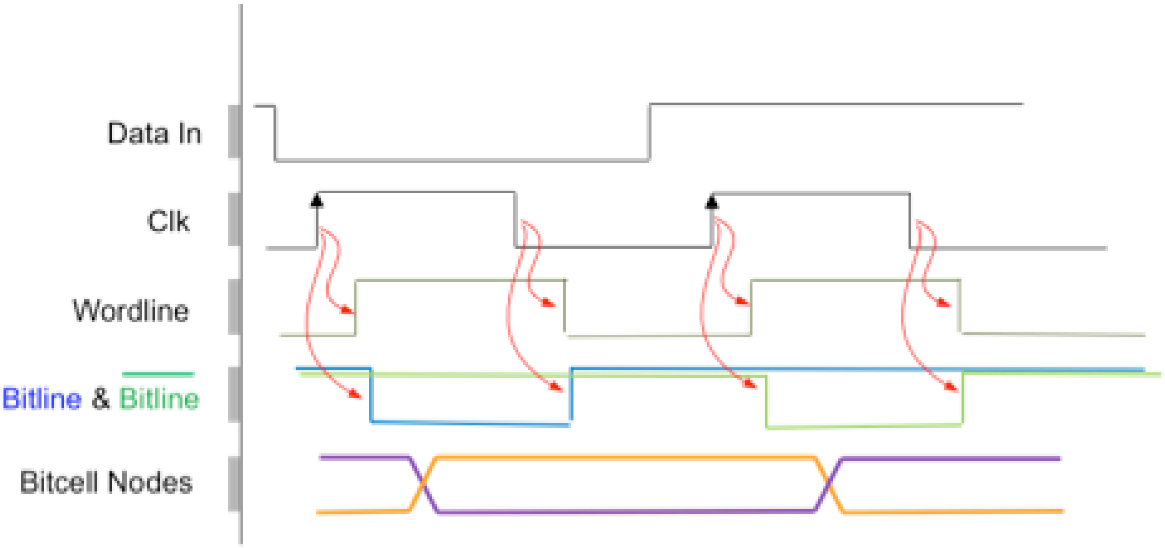


Write Circuit

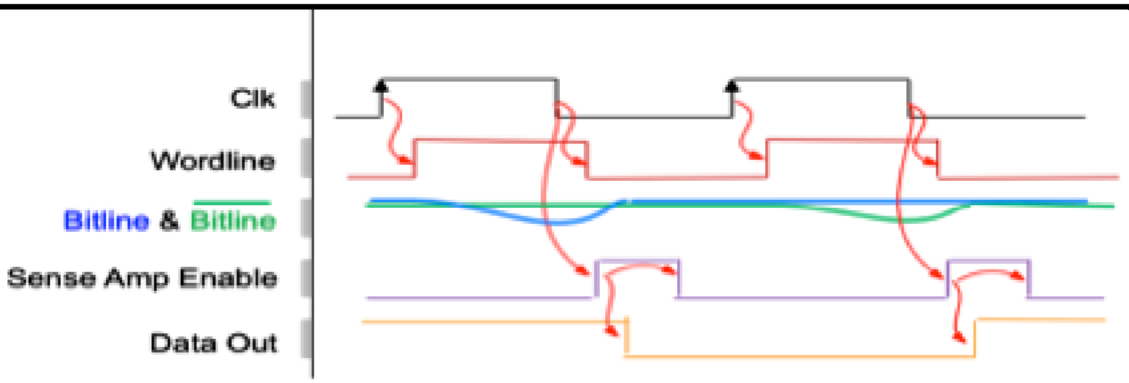


Read Circuit

An externally timed memory, alternatively called as a level triggered memory uses one edge of the clock to start the memory operation and the opposite edge to terminate the operation.

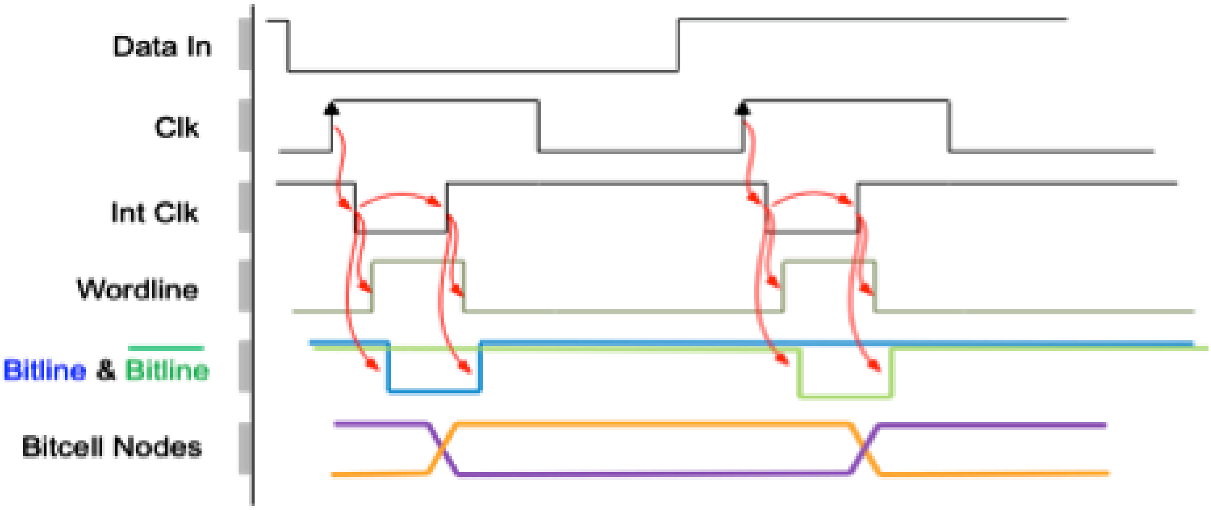


Write Waveforms for the Externally Clocked Design

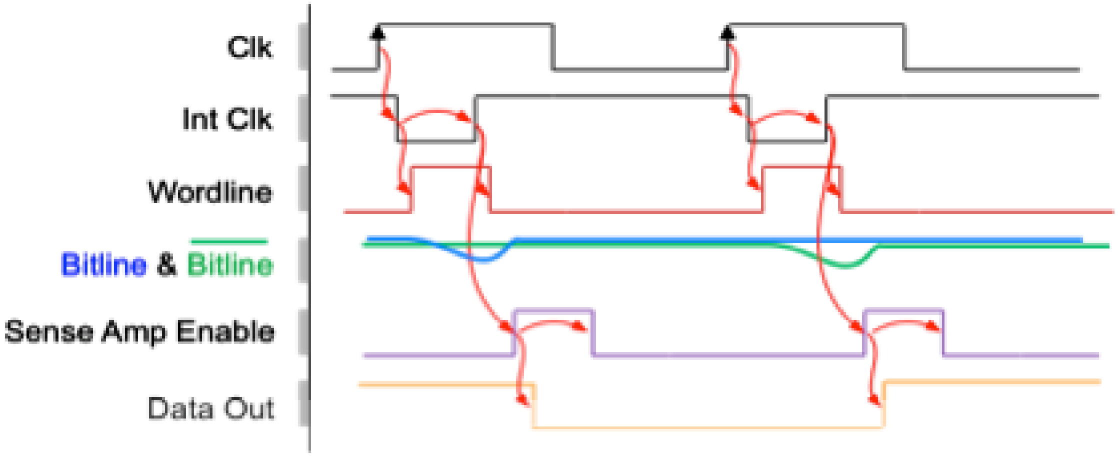


Read Waveforms for the Externally Clocked Design with Falling Edge Sense

A self-timed memory, alternatively called as an edge triggered memory uses a single active clock edge to start the cycle and an internal delay path to end the cycle. The opposite clock edge is only used to provide a reset to the clock generation circuit.



Write Waveforms for the Self-Timed Design



Read Waveforms for the Self-Timed Design

Minimum Pulse Width in the Externally Timed Memory

The pulse widths of the external clock are very important in the externally timed design. The clock high time (if the rising edge starts the cycle) starts the wordline and the bitline. Depending on the design, it may also enable the sense amplifier after some delay. The falling edge terminates the wordline, precharge the bitlines, and open the latches. It may also enable the sense amplifier.

The pulse width of the externally timed instance determines how much time is given to each of the internal operations.

The minimum pulse width for clock high needs to satisfy the following criteria:

- The wordline must be high and the bitline must be low enough to write to the bitcell. There is usually a margin requirement to ensure a robust write.
- If the sense amplifier enable results from the rising edge of the clock, the sense amplifier enable should terminate before the bitlines precharge.
- If the sense amplifier enable results from the falling edge of the clock, the bitlines must reach a sufficient signal before the sense amplifier is enabled.
- All signals in the clock propagation should reach full rail transitions.

The minimum pulse width for clock low must meet the following criteria:

- The precharge must be on for long enough to precharge the bitlines fully back to the supply.

- The latches must open early enough before the next rising edge of clock in order to propagate the new values of the latched signals to the next intersection with clock.

- All signals in the clock propagation should reach full rail transitions.

Minimum Period in the Externally Timed Memory

In the externally timed memory, each clock edge has requirements for spacing to the next clock edge. There is no additional rising edge to the subsequent rising edge requirement. As such, the minimum period of the externally timed memory instance should be equal to the sum of the minimum pulse width high and the minimum pulse width low.

Minimum Pulse Width in the Self-Timed Memory

In the self-timed memory, the primary purpose of the clock pulses are to operate the internal clock generator. The clock generator usually works by allowing the rising (active) edge of the clock to generate the internal clock. This is done by having a path from the input clock, through a gating device, and finally to a cross coupled inverter loop that stores the state of the internal clock. The gating device usually stays active for some time after the clock rises in order to reliably flip the internal clock state after which the gating device will turn off. After some time, the internal clock generates a termination signal that returns to the clock generator circuit and terminates the internal clock.

The clock high pulse must be of sufficient duration to reliably generate the internal clock. The clock low pulse must be of sufficient duration to allow the next rising edge of the clock and to have control over the clock generator when it arrives and start the next cycle. The minimum pulse width can be considered as a setup time for one clock edge against the other clock edge.

Minimum pulse width high is measured as follows:

(Clock rising to gating device turns off) – (Clock falling to internal clock generation device turns off)

Minimum pulse width low is measured as follows:

(Clock falling to gating device turns on) – (Clock rising to internal clock generation device turns on)

There are sometimes other locations that need to be considered for minimum pulse width in self-timed designs. These can be found in clock intersections with signals whose latches are not controlled by internal clock such as chip enable.

Minimum Period in the Self-Timed Memory

In the self-timed memory the clock period is an important component because it commands the duration of the internal clock generation and termination. The next clock cycle cannot start until the current cycle has completed and all internal circuitry has reset and is ready for the next cycle.

Since the minimum period is the sum of the internal clock high pulse and the internal clock low pulse, most of the same criteria as was mentioned in the externally timed pulse width section also apply here. However, in the self-timed case, the active pulse width duration is stated by the control circuitry, so minimum period is the sum of the actual active pulse width plus the measured minimum inactive pulse width.

The minimum period needs to be of appropriate duration to satisfy the following:

- The bitlines must be fully pre-charged to the full supply voltage before the next cycle accesses the bitlines.
- The latches must open early enough to allow the new data to propagate through the instance before it intersects with the clock.
- All clock propagation signals should be able to reach either power or ground within the inactive internal cycle.
- The external clock minimum pulse widths should be accounted for.

Using Liberate MX to Characterize Minimum Pulse Width and Minimum Period

There are some components of minimum pulse width and minimum period that can be automatically characterized with Liberate MX and some that need to be coded by the user using the define_measure command. When using define_measure, the user can utilize the mx_match_node command in order to automatically find the names of the nodes that need to be measured. The various components of a single arc can be bundled together using the bundle attribute.

Minimum Pulse Width High in the Externally Timed Memory

The measurements needed to correctly characterize the minimum pulse width of the externally timed memory primarily need to be coded by the user using the define_measure command. This should include the previously mentioned components.

The wordline and bitline pulses are both the result of the high pulse of the external clock. As such, the external clock high pulse width needs to be appropriate to provide enough time to reliably write to the bitcell. The bitcell write operation is an important part of the memory operation and there should be sufficient margin for the time needed to ensure a reliable write. A margining factor of two could be considered appropriate. Define_measure statements should be coded to create the following equation:

(MAX(CLK rise ➔ bitcell flips) – MAX(CLK rise ➔ Wordline rise, BL falls))*(margining factor) – MIN(CLK fall ➔ Wordline fall, Bitline rise)

```
define_measure \
    -name clk_r_flip_r \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ core_node_0 \
    -targ_type "delay" \
    -targ_dir rise \
    -targ_val 0.9 \
    inst_name

define_measure \
    -name clk_r_flip_f \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ core_node_1 \
    -targ_type "delay" \
    -targ_dir rise \
    -targ_val 0.1 \
    inst_name

define_measure \
    -name clk_r_wl_r \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ word_line \
    -targ_type "delay" \
    -targ_dir rise \
    -targ_val 0.5 \
    inst_name

define_measure \
    -name clk_r_bl_f \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ bit_line \
    -targ_type "delay" \
    -targ_dir fall \
    -targ_val 0.5 \
    inst_name

define_measure \
    -name clk_f_wl_f \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ word_line \
    -targ_type "delay" \
    -targ_dir fall \
    -targ_val 0.5 \
    inst_name

define_measure \
    -name clk_f_bl_r \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ bit_line \
    -targ_type "delay" \
    -targ_dir rise \
    -targ_val 0.5 \
    inst_name

define_measure \
    -name mpw_h_write \
    -duration 0.75 \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ_type "delay" \
    -equations "((MAX(clk_r_flip_r,clk_r_flip_f)-
MAX(clk_r_wl_r,clk_r_bl_f))*2)-MIN(clk_f_wl_f,clk_f_bl_r)" \
    inst_name
```

In the read cycle, there are two options of how the sense amplifier can be enabled. The sense amplifier can be enabled after an internal timer activates it, or when the external clock falls. Each of these options has a different measure for minimum pulse width high.

For the case when the sense amplifier is enabled from an internal timer, the clock must stay high long enough for the sensing to complete. The way this can be measured is by ensuring that the sense amplifier output is fully transitioned before the bitline can start to precharge.

(CLK rise ➔ sense amp out) – (CLK fall ➔ precharger)

```
define_measure \
    -name clk_r_sa_out \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ sense_amp_output \
    -targ_type "delay" \
    -targ_dir rise \
    -targ_val 0.8 \
    inst_name

define_measure \
    -name clk_f_prch_f \
    -trig CLK \
    -trig_dir fall \
    -trig_val 0.5 \
    -targ precharge \
    -targ_type "delay" \
    -targ_dir fall \
    -targ_val 0.8 \
    inst_name

define_measure \
    -name mpw_h_read \
    -duration 0.75 \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ_type "delay" \
    -equations "clk_r_sa_out-clk_f_prch_f" \
    inst_name
```

In the case that the falling external clock enables the sense amplifier, the external clock high pulse width determines the duration of time that the bitline signal requires to develop. A reliable read will usually require 10-20% of supply in bitline differential at the time that the sense amplifier is activated. The define_measure statements are needed to correctly measure this component with the following equation:

(CLK rise ➔ (bitline – bitline bar = 20%supply) – (CLK fall ➔ sense amp enable)

```
define_measure \
    -name clk_r_bl_diff \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ (BL,BLB) \
    -targ_type "delay" \
    -targ_dir rise \
    -targ_val 0.2 \
    inst_name

define_measure \
    -name clk_f_sae \
    -trig CLK \
    -trig_dir fall \
    -trig_val 0.5 \
    -targ sense_amp_enable \
    -targ_type "delay" \
    -targ_dir rise \
    -targ_val 0.5 \
    inst_name

define_measure \
    -name mpw_h_read \
    -duration 0.75 \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ_type "delay" \
    -equations "clk_r_bl_diff-clk_f_sae" \
    inst_name
```

There may be other signals that need to be monitored for specific designs. These measurements need to be determined by the engineer.

Minimum Pulse Width Low in the Externally Timed Memory

There are two primary components of the minimum pulse width low for the externally timed memory. The bitline precharge need to be coded by the user using the define_measure command and the latch opening component can be measured automatically if the vector stimulus is set up properly.

The bitline precharge is enabled when the external clock goes low. The external clock must stay low long enough for the bitline to reach supply before the clock goes high again and turns off the precharge. This component need to be coded using the define_memory command, based on the following equation:

(CLK fall ➔ bitline precharged) – (CLK rise ➔ precharge disabled)

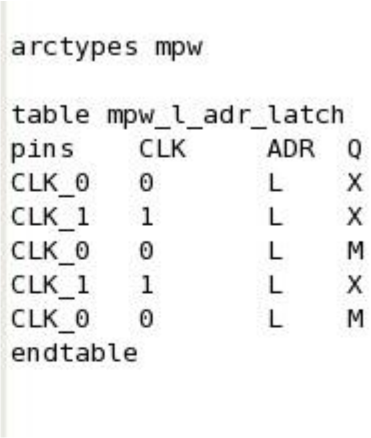
```
define_measure \
    -name clk_f_bl_precharge \
    -trig CLK \
    -trig_dir fall \
    -trig_val 0.5 \
    -targ bit_line \
    -targ_type "delay" \
    -targ_dir rise \
    -targ_val 0.99 \
    inst_name

define_measure \
    -name clk_r_prch_r \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ precharge \
    -targ_type "delay" \
    -targ_dir rise \
    -targ_val 0.2 \
    inst_name

define_measure \
    -name mpw_l_prch \
    -duration 0.75 \
    -trig CLK \
    -trig_dir fall \
    -trig_val 0.5 \
    -targ_type "delay" \
    -equations "clk_f_bl_precharge-clk_r_prch_r" \
    inst_name
```

In the externally timed instance, the latches for signals, such as address, data, write enable, and chip enable, closes with the rising edge of the clock and reopen with the falling edge. This means that the latch must open sufficient amount of time before the next rising edge of the clock. This allows the new values to propagate through the design. In order to stimulate this, the signals must change while the latch is closed. With the proper stimulus, this arc can automatically be probed by Liberate MX without the user providing define_measure.

The following table is an example for address. You can similarly write other tables as well.



```
arctypes mpw

table mpw_l_adr_latch
pins    CLK     ADR  Q
CLK_0   0       L    X
CLK_1   1       L    X
CLK_0   0       L    M
CLK_1   1       L    X
CLK_0   0       L    M
endtable
```

Minimum Period in the Externally Timed Memory

In the externally timed memory, the minimum period is the sum of the minimum pulse width low and the minimum pulse width high. If the minimum pulse widths are properly characterized, then the minimum period is automatically loaded as the sum without any further user input.

Minimum Pulse Width in the Internally Timed Memory

In the internally timed memory, the minimum pulse width requirements are determined at the internal clock generator circuit. These measurements are fully characterized by Liberate MX without any further user input.

Minimum Period in the Internally Timed Memory

The minimum period in the internally timed memory has several components. These components are automatically characterized by Liberate MX. The components include the latch component and the precharging of the bitlines before the next clock cycle.

In addition to the automatically characterized components, there are several design specific components that the user needs to use to describe define_measure.

The pulses of all clock propagation signals must make full transitions to supply and ground. These signals can vary for different designs, but usually contain signals such as wordlines or timer signals. The user needs to code these using the define_measure command.

Following is an example of measuring the time required for full transitions of the wordline. You can similarly write other measures as well.

```
define_measure \
    -name clk_r_wl_r_05 \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ word_line \
    -targ_type "delay" \
    -targ_dir rise \
    -targ_val 0.05 \
    inst_name

define_measure \
    -name clk_r_wl_f_05 \
    -trig CLK \
    -trig_dir rise \
    -trig_val 0.5 \
    -targ word_line \
    -targ_type "delay" \
    -targ_dir fall \
    -targ_val 0.05 \
    inst_name

define_measure \
    -name minp_wl_pulse \
    -duration 0.25 \
    -trig CLK \
    -trig_dir fall \
    -trig_val 0.5 \
    -targ_type "delay" \
    -equations "clk_r_wl_f_05-clk_r_wl_r_05" \
    inst_name
```

The minimum period of the internally timed instance also need to consider the values of the minimum pulse width high and minimum pulse width low. This component is included automatically by Liberate MX.

Summary

This document instructs the user on how to characterize minimum pulse width and minimum period for various memory designs. The document also describes how to perform various measurements using Liberate MX.