Complex Clocking Situations Using Primetime

Paul Zimmer
(pzimmer@cisco.com)

Telco chips have:

- Multiplexed clocks
- Falling-, Rising-, and Both-edge clocks
- Divide-by clocks with unusual phase relationships
- Combinations of the above
- Many clocks (one chip had over 200!)

Topics covered in paper

- Clock Muxes / Managing Large Numbers of Clocks
- I/O Interfaces with Outgoing Clocks
- Describing complex clocking relationships using -edges
- Phantom Delays on input and output ports
- Parsing command output for use in a script

Topics covered in paper

- **Clock Muxes / Managing Large Numbers of Clocks**
- I/O Interfaces with Outgoing Clocks
- Describing complex clocking relationships using -edges
- Phantom Delays on input and output ports
- Parsing command output for use in a script

Clock Muxes / Managing large numbers of clocks

- Use check_timing to make sure that *all* clock muxes are controlled
- Use array to deal with false paths between unrelated clocks.
- For details and examples, see paper.

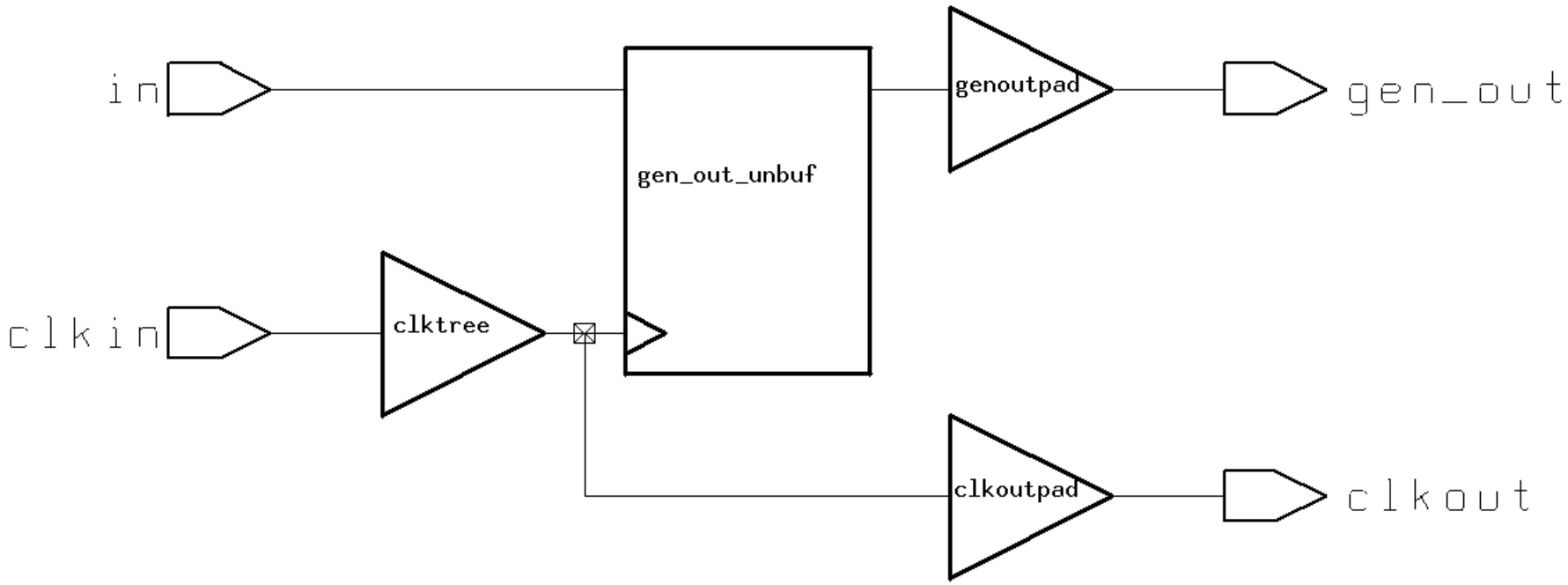
Topics covered in paper

- Clock Muxes / Managing Large Numbers of Clocks
- **I/O Interfaces with Outgoing Clocks**
- Describing complex clocking relationships using -edges
- Phantom Delays on input and output ports
- Parsing command output for use in a script

I/O interfaces with outgoing clocks

- Becoming very common.
- Allows for higher speeds, since clock cycle is limited by skew rather than absolute delay.
- Challenge is constraining them in PrimeTime such that timing problem appear as VIOLATIONS.

Basic Outgoing Clock



Outgoing clock code using divide_by 1

```
create_clock -period 10.0 [get_ports clkin]
set_propagated_clock clkin
create_generated_clock \
    -name clkout \
    -source [get_ports clkin] \
    -divide_by 1 \
    [get_ports clkout]
;
set _gen2src(clkout) clkin

set_output_delay -clock clkout 1.0 gen_out

# Now create a difference in output timing:

set_load 0.5 [get_ports gen_out]
set_load 0.25 [get_ports clkout]
set_annotated_delay -cell -from clkoutpad/A -to clkoutpad/Z -rise 0.3
set_annotated_delay -cell -from clkoutpad/A -to clkoutpad/Z -fall 0.1
```

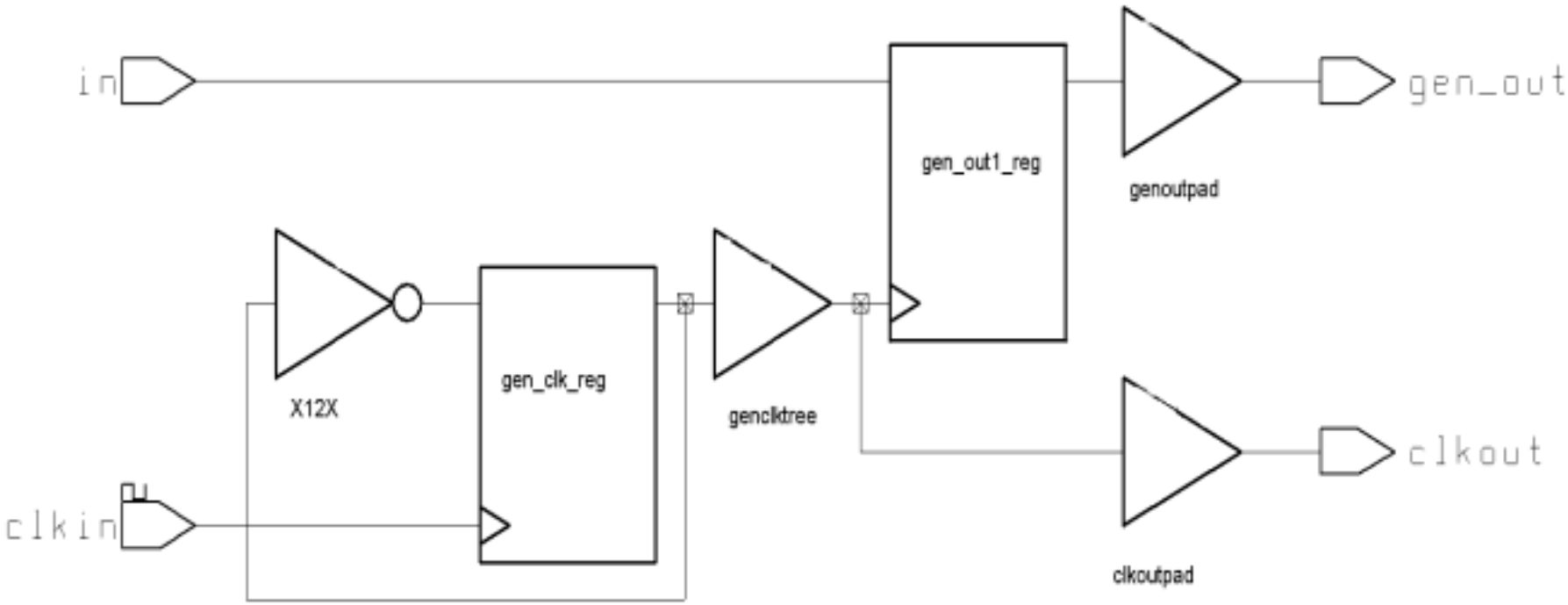
Resulting Timing Report

```
report_timing -input_pins -path_type full_clock -to gen_out

Startpoint: gen_out_unbuf_reg
            (rising edge-triggered flip-flop clocked by clkin)
Endpoint: gen_out (output port clocked by clkout)
Path Group: clkout
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
clock clkin (rise edge)                        0.00       0.00
clock source latency                           0.00       0.00
clkin (in)                                     0.00       0.00 r
clktree/A (BUFC)                               0.00       0.00 r
clktree/Z (BUFC)                               0.11       0.11 r
gen_out_unbuf_reg/CP (FD1QA)                   0.00       0.11 r
gen_out_unbuf_reg/CP (FD1QA)                   0.00       0.11 r
gen_out_unbuf_reg/Q (FD1QA)                    0.33       0.44 r
genoutpad/A (BUFC)                             0.00       0.44 r
genoutpad/Z (BUFC)                             0.42       0.86 r
gen_out (out)                                  0.00       0.86 r
data arrival time                                         0.86

clock clkout (rise edge)                      10.00      10.00
clock network delay (ideal)                    0.41      10.41
output external delay                         -1.00       9.41
data required time                                       9.41
-----------------------------------------------------------------
data required time                                       9.41
data arrival time                                       -0.86
-----------------------------------------------------------------
slack (MET)                                              8.55
```

Where did the “clock network delay” come from?

```
 report_timing -delay max_rise -to [get_ports clkout]

Startpoint: clkin (clock source 'clkin')
Endpoint: clkout (output port)
Path Group: (none)
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
clkin (in)                                     0.00       0.00 r
clktree/Z (BUFC)                               0.11       0.11 r
clkoutpad/Z (BUFB)                             0.30 *     0.41 r
clkout (out)                                   0.00       0.41 r
data arrival time                                         0.41
```

Divided Outgoing Clock



Divided Outgoing Clock Code

```
create_clock -period 10.0 [get_ports clkin]
set_propagated_clock clkin

# create the div2 clock on the port
create_generated_clock \
    -name clkout \
    -source [get_ports clkin] \
    -divide_by 2 \
    [get_ports clkout]
;
# create the div2 clock on the Q pin (so that it gets used for gen_out flop)
create_generated_clock \
    -name clkout_at_q \
    -source [get_ports clkin] \
    -divide_by 2 \
    [get_pins gen_clk_reg/Q]
;
set_propagated_clock clkout_at_q

set_output_delay -clock clkout 1.0 gen_out

# To make the timing report easier to follow, we'll again add some
# loads and delays:

set_load 0.5 [get_ports gen_out]
set_load 0.25 [get_ports clkout]
```

Resulting Timing Report

```
report_timing -to [get_ports gen_out] -input_pins -path_type full_clock

 Startpoint: gen_out1_reg
             (rising edge-triggered flip-flop clocked by clkout_at_q)
 Endpoint: gen_out (output port clocked by clkout)
 Path Group: clkout
 Path Type: max

 Point                                          Incr       Path
 -----------------------------------------------------------------
 clock clkout_at_q (rise edge)                  0.00       0.00
 clock source latency                           0.34       0.34
 gen_clk_reg/Q (FD1QA)                          0.00       0.34 r
 genclktree/A (BUFC)                            0.00       0.34 r
 genclktree/Z (BUFC)                            0.14       0.48 r
 gen_out1_reg/CP (FD1QA)                        0.00       0.48 r
 gen_out1_reg/CP (FD1QA)                        0.00       0.48 r
 gen_out1_reg/Q (FD1QA)                         0.33       0.82 r
 genoutpad/A (BUFC)                             0.00       0.82 r
 genoutpad/Z (BUFC)                             0.42       1.24 r
 gen_out (out)                                  0.00       1.24 r
 data arrival time                                         1.24

 clock clkout (rise edge)                      20.00      20.00
 clock network delay (ideal)                    0.81      20.81
 output external delay                         -1.00      19.81
 data required time                                       19.81
 -----------------------------------------------------------------
 data required time                                       19.81
 data arrival time                                        -1.24
 -----------------------------------------------------------------
 slack (MET)                                              18.57
```

Where do the 0.34 and 0.81 come from?

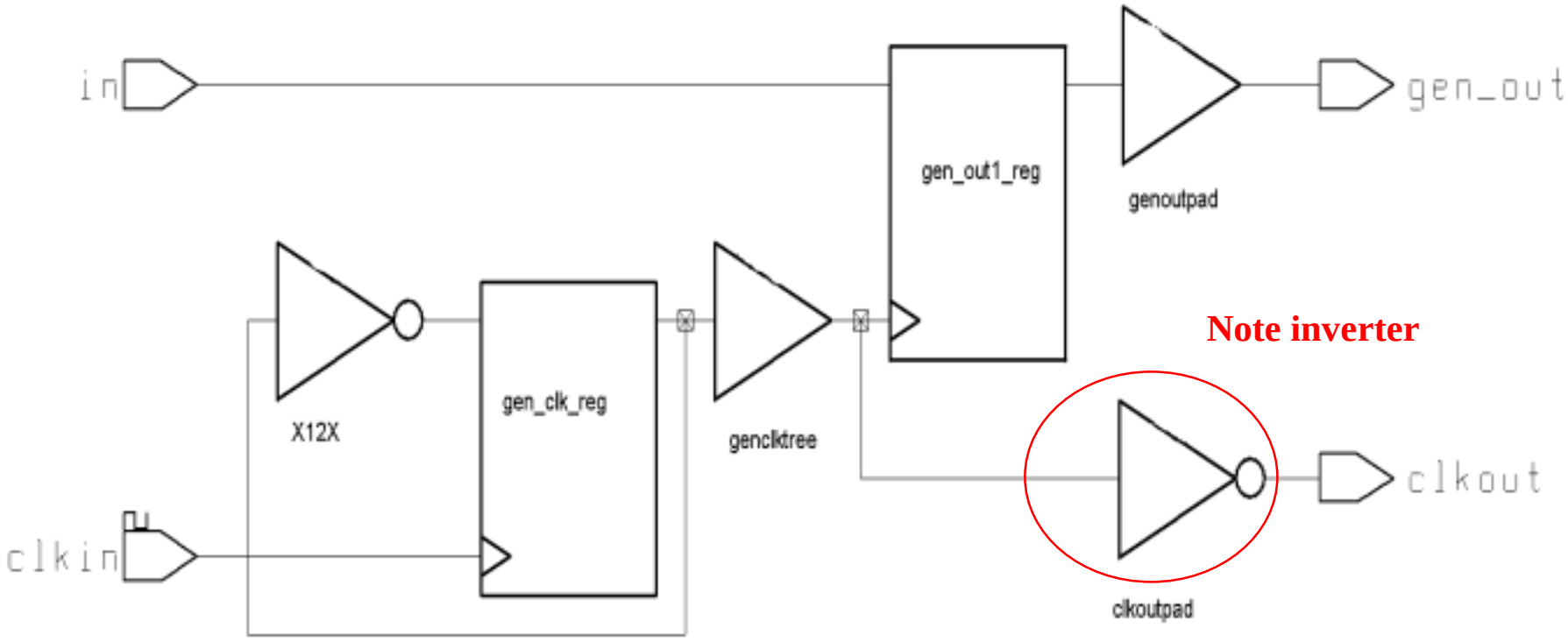
First, here's where the "clock source latency" of clkout_at_q comes from:

```
  report_timing -from gen_clk_reg/CP -to gen_clk_reg/Q

Startpoint: gen_clk_reg
            (rising edge-triggered flip-flop clocked by clkin)
Endpoint: gen_clk_reg/Q (internal pin)
Path Group: (none)
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
gen_clk_reg/CP (FD1QA)                         0.00       0.00 r
gen_clk_reg/Q (FD1QA)                          0.34       0.34 r
data arrival time                                         0.34
```

The "clock network delay (ideal)" of clkout is this value plus the delay through the genclktree and the clkout pad:

```
report_timing -to [get_ports clkout]

Startpoint: gen_clk_reg/Q
            (clock source 'clkout_at_q')
Endpoint: clkout (output port)
Path Group: (none)
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
clock source latency                           0.34       0.34
gen_clk_reg/Q (FD1QA)                          0.00       0.34 r
genclktree/Z (BUFC)                            0.14       0.48 r
clkoutpad/Z (BUFB)                             0.33       0.81 r
clkout (out)                                   0.00       0.81 r
data arrival time                                         0.81
```

Outgoing *Inverted* Clock



Outgoing Inverted Clock code

```
create_clock -period 10.0 [get_ports clkin]
set_propagated_clock clkin

# create the div2 clock on the port
create_generated_clock \
    -name clkout \
    -source [get_ports clkin] \
    -divide_by 2 \
    -invert \
    [get_ports clkout]
;

# create the div2 clock on the Q pin (so that it gets used for gen_out flop)
create_generated_clock \
    -name clkout_at_q \
    -source [get_ports clkin] \
    -divide_by 2 \
    [get_pins gen_clk_reg/Q]
;
set_propagated_clock clkout_at_q

set_output_delay -clock clkout 1.0 gen_out

# create a difference in output timing
set_load 0.5 [get_ports gen_out]
set_load 0.25 [get_ports clkout]
```

Outgoing Inverted Clock timing report

```
  report_timing -to [get_ports gen_out] -input_pins -path_type full_clock

Startpoint: gen_out1_reg
            (rising edge-triggered flip-flop clocked by clkout_at_q)
Endpoint: gen_out (output port clocked by clkout)
Path Group: clkout
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
clock clkout_at_q (rise edge)                  0.00       0.00
clock source latency                           0.34       0.34
gen_clk_reg/Q (FD1QA)                          0.00       0.34 r
genclktree/A (BUFC)                            0.00       0.34 r
genclktree/Z (BUFC)                            0.15       0.49 r
gen_out1_reg/CP (FD1QA)                        0.00       0.49 r
gen_out1_reg/CP (FD1QA)                        0.00       0.49 r
gen_out1_reg/Q (FD1QA)                         0.34       0.83 r
genoutpad/A (BUFC)                             0.00       0.83 r
genoutpad/Z (BUFC)                             0.42       1.25 r
gen_out (out)                                  0.00       1.25 r
data arrival time                                         1.25

clock clkout (rise edge)                      10.00      10.00
clock network delay (ideal)                    0.77      10.77
output external delay                         -1.00       9.77
data required time                                        9.77
-----------------------------------------------------------------
data required time                                        9.77
data arrival time                                        -1.25
-----------------------------------------------------------------
slack (MET)                                               8.52
```

Outgoing Inverted Clock timing report - hold

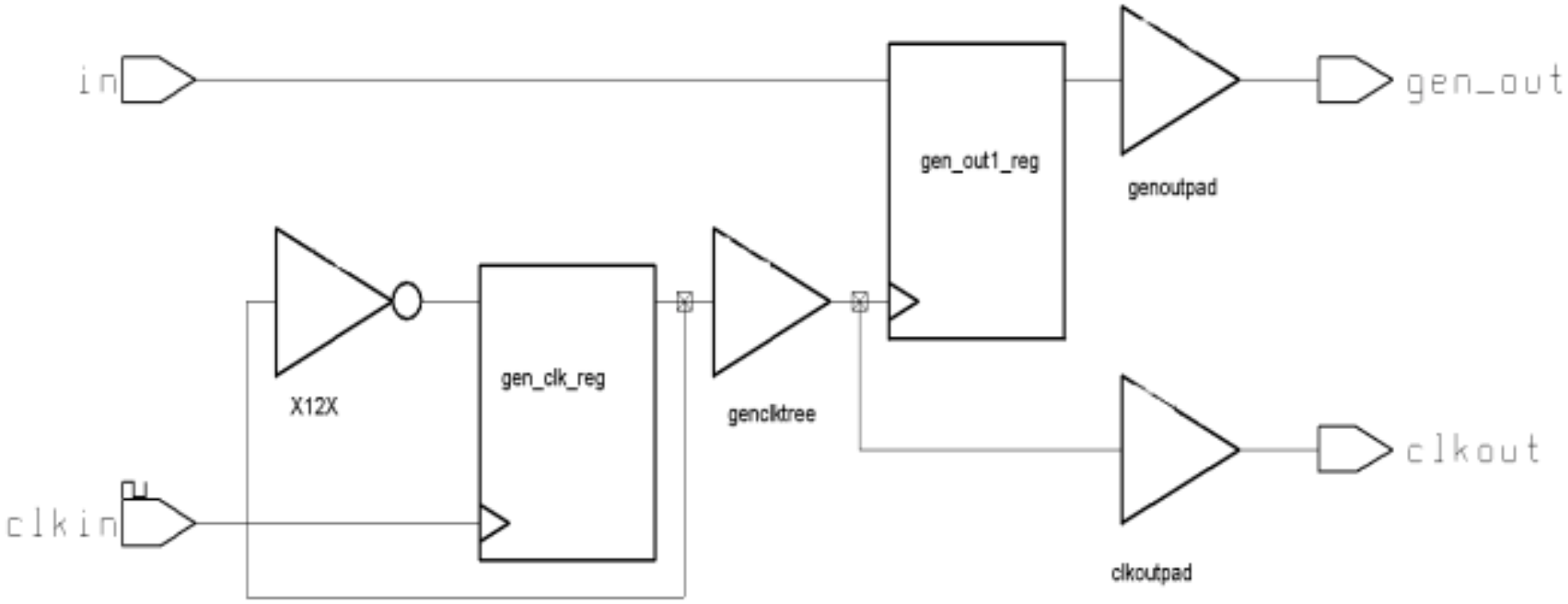
```
  report_timing -to [get_ports gen_out] -input_pins -path_type full_clock -
delay min

  Startpoint: gen_out1_reg
              (rising edge-triggered flip-flop clocked by clkout_at_q)
  Endpoint: gen_out (output port clocked by clkout)
  Path Group: clkout
  Path Type: min

  Point                                          Incr       Path
  -----------------------------------------------------------------
  clock clkout_at_q (rise edge)                 20.00      20.00
  clock source latency                           0.34      20.34
  gen_clk_reg/Q (FD1QA)                          0.00      20.34 r
  genclktree/A (BUFC)                            0.00      20.34 r
  genclktree/Z (BUFC)                            0.15      20.49 r
  gen_out1_reg/CP (FD1QA)                        0.00      20.49 r
  gen_out1_reg/CP (FD1QA)                        0.00      20.49 r
  gen_out1_reg/Q (FD1QA)                         0.34      20.83 f
  genoutpad/A (BUFC)                             0.00      20.83 f
  genoutpad/Z (BUFC)                             0.36      21.19 f
  gen_out (out)                                  0.00      21.19 f
  data arrival time                                        21.19

  clock clkout (rise edge)                      10.00      10.00
  clock network delay (ideal)                    0.77      10.77
  output external delay                         -1.00       9.77
  data required time                                        9.77
  -----------------------------------------------------------------
  data required time                                        9.77
  data arrival time                                       -21.19
  -----------------------------------------------------------------
  slack (MET)                                              11.42
```

Where does the 0.77ns come from?

```
report_timing -to [get_ports clkout] -delay max_rise

Startpoint: gen_clk_reg/Q
            (clock source 'clkout_at_q')
Endpoint: clkout (output port)
Path Group: (none)
Path Type: max

Point                                              Incr       Path
-----------------------------------------------------------------------
clock source latency                               0.34       0.34
gen_clk_reg/Q (FD1QA)                              0.00       0.34 f
genclktree/Z (BUFC)                                0.17       0.50 f
clkoutpad/Z (N1B)                                  0.27       0.77 r
clkout (out)                                       0.00       0.77 r
data arrival time                                             0.77
```

What if the external device uses falling edges?

- Back to original circuit:



What if the external device uses falling edges?

- Code now has “-clock_fall”:

```
create_clock -period 10.0 [get_ports clkin]
set_propagated_clock clkin

# create the div2 clock on the port
create_generated_clock \
    -name clkout \
    -source [get_ports clkin] \
    -divide_by 2 \
    [get_ports clkout]
;
# create the div2 clock on the Q pin (so that it gets used for gen_out flop)
create_generated_clock \
    -name clkout_at_q \
    -source [get_ports clkin] \
    -divide_by 2 \
    [get_pins gen_clk_reg/Q]
;
set_propagated_clock clkout_at_q

set_output_delay -clock clkout 1.0 gen_out -clock_fall

# To make the timing report easier to follow, we'll again add some
# loads and delays:

set_load 0.5 [get_ports gen_out]
set_load 0.25 [get_ports clkout]
```

External Device uses falling edges - timing report

```
report_timing -to [get_ports gen_out] -input_pins -path_type full_clock


Startpoint: gen_out1_reg
            (rising edge-triggered flip-flop clocked by clkout_at_q)
Endpoint: gen_out (output port clocked by clkout)
Path Group: clkout
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
clock clkout_at_q (rise edge)                  0.00       0.00
clock source latency                           0.34       0.34
gen_clk_reg/Q (FD1QA)                          0.00       0.34 r
genclktree/A (BUFC)                            0.00       0.34 r
genclktree/Z (BUFC)                            0.14       0.48 r
gen_out1_reg/CP (FD1QA)                        0.00       0.48 r
gen_out1_reg/CP (FD1QA)                        0.00       0.48 r
gen_out1_reg/Q (FD1QA)                         0.33       0.82 r
genoutpad/A (BUFC)                             0.00       0.82 r
genoutpad/Z (BUFC)                             0.42       1.24 r
gen_out (out)                                  0.00       1.24 r
data arrival time                                         1.24

clock clkout (fall edge)                      10.00      10.00
clock network delay (ideal)                    0.78      10.78
output external delay                         -1.00       9.78
data required time                                        9.78
-----------------------------------------------------------------
data required time                                        9.78
data arrival time                                        -1.24
-----------------------------------------------------------------
slack (MET)                                               8.54
```

Where does the 0.78 ns come from?


```
report_timing -to [get_ports clkout] -delay max_fall

Startpoint: gen_clk_reg/Q
            (clock source 'clkout_at_q')
Endpoint: clkout (output port)
Path Group: (none)
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
clock source latency                           0.34       0.34
gen_clk_reg/Q (FD1QA)                          0.00       0.34 f
genclktree/Z (BUFC)                            0.16       0.50 f
clkoutpad/Z (BUFB)                             0.28       0.78 f
clkout (out)                                   0.00       0.78 f
data arrival time                                         0.78
```

What happens if you do *both*?

- Setup trace:

```
report_timing -to [get_ports gen_out] -input_pins -path_type full_clock

Startpoint: gen_out1_reg
            (rising edge-triggered flip-flop clocked by clkout_at_q)
Endpoint: gen_out (output port clocked by clkout)
Path Group: clkout
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
clock clkout_at_q (rise edge)                  0.00       0.00
clock source latency                           0.34       0.34
gen_clk_reg/Q (FD1QA)                          0.00       0.34 r
genclktree/A (BUFC)                            0.00       0.34 r
genclktree/Z (BUFC)                            0.15       0.49 r
gen_out1_reg/CP (FD1QA)                        0.00       0.49 r
gen_out1_reg/CP (FD1QA)                        0.00       0.49 r
gen_out1_reg/Q (FD1QA)                         0.34       0.83 r
genoutpad/A (BUFC)                             0.00       0.83 r
genoutpad/Z (BUFC)                             0.42       1.25 r
gen_out (out)                                  0.00       1.25 r
data arrival time                                         1.25

clock clkout (fall edge)                      20.00      20.00
clock network delay (ideal)                    0.68      20.68
output external delay                         -1.00      19.68
data required time                                       19.68
-----------------------------------------------------------------
data required time                                       19.68
data arrival time                                        -1.25
-----------------------------------------------------------------
slack (MET)                                              18.43
```

What happens if you do *both*?

- Hold trace:

```
     report_timing -to [get_ports gen_out] -input_pins -path_type full_clock -
delay min

  Startpoint: gen_out1_reg
              (rising edge-triggered flip-flop clocked by clkout_at_q)
  Endpoint: gen_out (output port clocked by clkout)
  Path Group: clkout
  Path Type: min

  Point                                          Incr       Path
  ---------------------------------------------------------------
  clock clkout_at_q (rise edge)                  0.00       0.00
  clock source latency                           0.34       0.34
  gen_clk_reg/Q (FD1QA)                          0.00       0.34 r
  genclktree/A (BUFC)                            0.00       0.34 r
  genclktree/Z (BUFC)                            0.15       0.49 r
  gen_out1_reg/CP (FD1QA)                        0.00       0.49 r
  gen_out1_reg/CP (FD1QA)                        0.00       0.49 r
  gen_out1_reg/Q (FD1QA)                         0.34       0.83 f
  genoutpad/A (BUFC)                             0.00       0.83 f
  genoutpad/Z (BUFC)                             0.36       1.19 f
  gen_out (out)                                  0.00       1.19 f
  data arrival time                                         1.19

  clock clkout (fall edge)                       0.00       0.00
  clock network delay (ideal)                    0.68       0.68
  output external delay                         -1.00      -0.32
  data required time                                       -0.32
  ---------------------------------------------------------------
  data required time                                       -0.32
  data arrival time                                        -1.19
  ---------------------------------------------------------------
  slack (MET)                                               1.52
```

And where did the 0.68 come from?

- From the *falling* edge of clkout:

```
report_timing -to [get_ports clkout] -delay max_fall

Startpoint: gen_clk_reg/Q
            (clock source 'clkout_at_q')
Endpoint: clkout (output port)
Path Group: (none)
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
clock source latency                           0.34       0.34
gen_clk_reg/Q (FD1QA)                          0.00       0.34 r
genclktree/Z (BUFC)                            0.15       0.49 r
clkoutpad/Z (N1B)                              0.18       0.67 f
clkout (out)                                   0.00       0.68 f
data arrival time                                         0.68
```

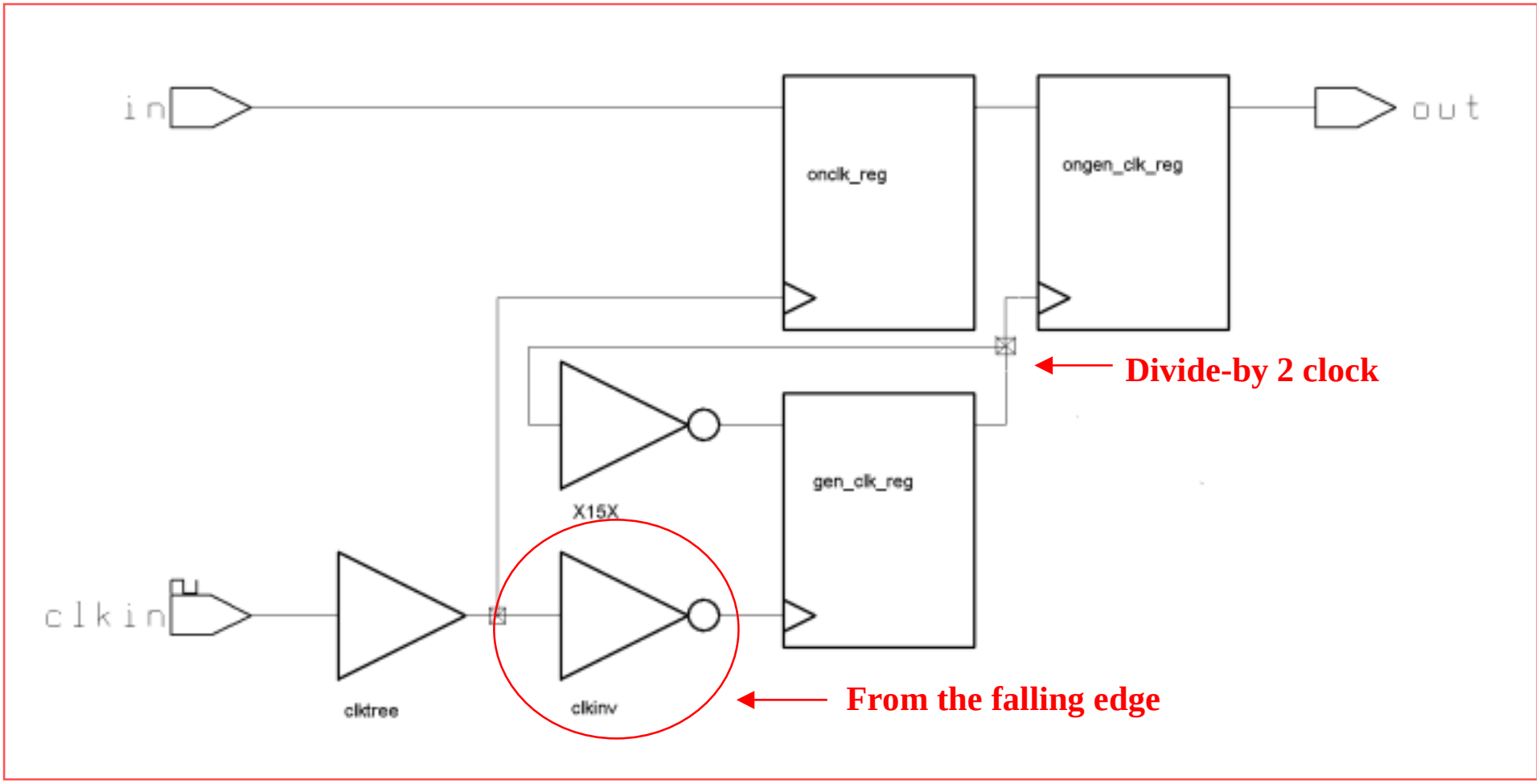
Topics covered in paper

- Clock Muxes / Managing Large Numbers of Clocks
- I/O Interfaces with Outgoing Clocks
- **Describing complex clocking relationships using -edges**
- Phantom Delays on input and output ports
- Parsing command output for use in a script

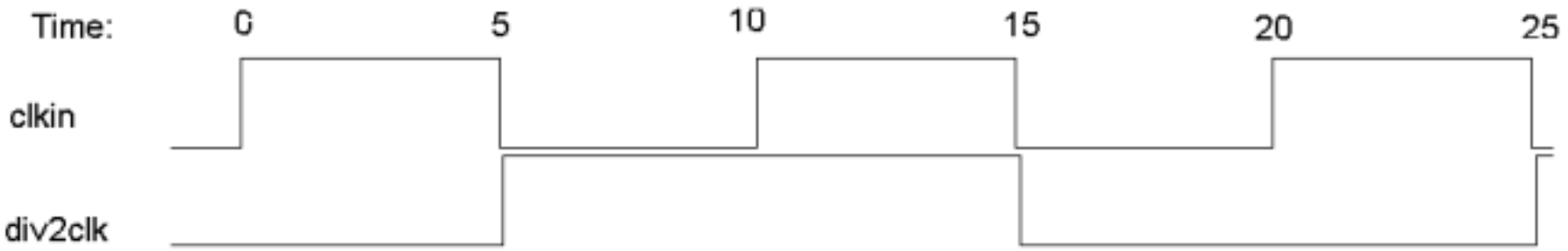
Describing complex clocking relationships using -edges

- Most common divided clocks can be easily described using the “-divide_by” option on create_generated_clocks.
- Sometimes, this doesn’t work - particularly if things are happening on falling edges.

Basic use of -edges



Basic use of -edges - waveform



What would happen with divide_by 2

```
  report_timing -to [get_pins ongen_clk_reg/D] -input_pins -path_type
full_clock

  Startpoint: onclk_reg (rising edge-triggered flip-flop clocked by clkin)
  Endpoint: ongen_clk_reg
            (rising edge-triggered flip-flop clocked by div2clk)
  Path Group: div2clk
  Path Type: max

  Point                                    Incr       Path
  ------------------------------------------------------------
  clock clkin (rise edge)                 10.00      10.00
  clock source latency                     0.00      10.00
  clkin (in)                               0.00      10.00 r
  clktree/A (BUFC)                         0.00      10.00 r
  clktree/Z (BUFC)                         0.12      10.12 r
  onclk_reg/CP (FD1QA)                     0.00      10.12 r
  onclk_reg/CP (FD1QA)                     0.00      10.12 r
  onclk_reg/Q (FD1QA)                      0.34      10.45 f
  ongen_clk_reg/D (FD1QA)                  0.00      10.45 f
  data arrival time                                  10.45

  clock div2clk (rise edge)               20.00      20.00
  clock source latency                     0.55      20.55
  gen_clk_reg/Q (FD1QA)                    0.00      20.55 r
  ongen_clk_reg/CP (FD1QA)                 0.00      20.55 r
  library setup time                      -0.27      20.29
  data required time                                 20.29
  ------------------------------------------------------------
  data required time                                 20.29
  data arrival time                                 -10.45
  ------------------------------------------------------------
  slack (MET)                                         9.84
```

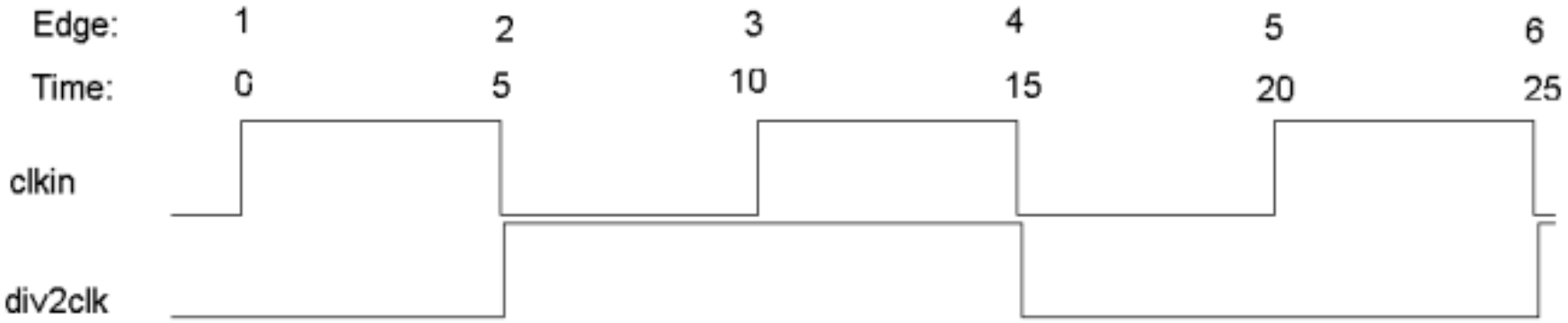
This is wrong

This is right

Half right – the clock path timing is inverted

```
report_timing -delay max_rise -to gen_clk_reg/CP

Startpoint: clkin (clock source 'clkin')
Endpoint: gen_clk_reg/CP (internal pin)
Path Group: (none)
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
clkin (in)                                     0.00       0.00 f
clktree/Z (BUFC)                               0.16       0.16 f
clkinv/Z (NIB)                                 0.04       0.20 r
gen_clk_reg/CP (FD1QA)                         0.00       0.20 r
data arrival time                                         0.20


report_timing -delay max_rise -from gen_clk_reg/CP -to gen_clk_reg/Q

Startpoint: gen_clk_reg
            (rising edge-triggered flip-flop clocked by clkin')
Endpoint: gen_clk_reg/Q (internal pin)
Path Group: (none)
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
gen_clk_reg/CP (FD1QA)                         0.00       0.00 r
gen_clk_reg/Q (FD1QA)                          0.35       0.35 r
data arrival time                                         0.35
```

Waveform with PT edges marked



Script using -edges

```
create_clock -period 10.0 [get_ports clkin]
set_propagated_clock clkin

create_generated_clock \
    -name div2clk \
    -source [get_ports clkin] \
    -edges {2 4 6} \
    [get_pins gen_clk_reg/Q]
;
set_propagated_clock div2clk
```

Timing report using -edges (setup)

```
  report_timing -to [get_pins ongen_clk_reg/D] -input_pins -path_type
full_clock

  Startpoint: onclk_reg (rising edge-triggered flip-flop clocked by clkin)
  Endpoint: ongen_clk_reg
            (rising edge-triggered flip-flop clocked by div2clk)
  Path Group: div2clk
  Path Type: max

  Point                                          Incr       Path
  ---------------------------------------------------------------
  clock clkin (rise edge)                        0.00       0.00
  clock source latency                           0.00       0.00
  clkin (in)                                     0.00       0.00 r
  clktree/A (BUFC)                               0.00       0.00 r
  clktree/Z (BUFC)                               0.12       0.12 r
  onclk_reg/CP (FD1QA)                           0.00       0.12 r
  onclk_reg/CP (FD1QA)                           0.00       0.12 r
  onclk_reg/Q (FD1QA)                            0.34       0.45 f
  ongen_clk_reg/D (FD1QA)                        0.00       0.45 f
  data arrival time                                         0.45

  clock div2clk (rise edge)                      5.00       5.00
  clock source latency                           0.55       5.55
  gen_clk_reg/Q (FD1QA)                          0.00       5.55 r
  ongen_clk_reg/CP (FD1QA)                       0.00       5.55 r
  library setup time                            -0.27       5.29
  data required time                                       5.29
  ---------------------------------------------------------------
  data required time                                       5.29
  data arrival time                                       -0.45
  ---------------------------------------------------------------
  slack (MET)                                              4.84
```

Right!

Right!

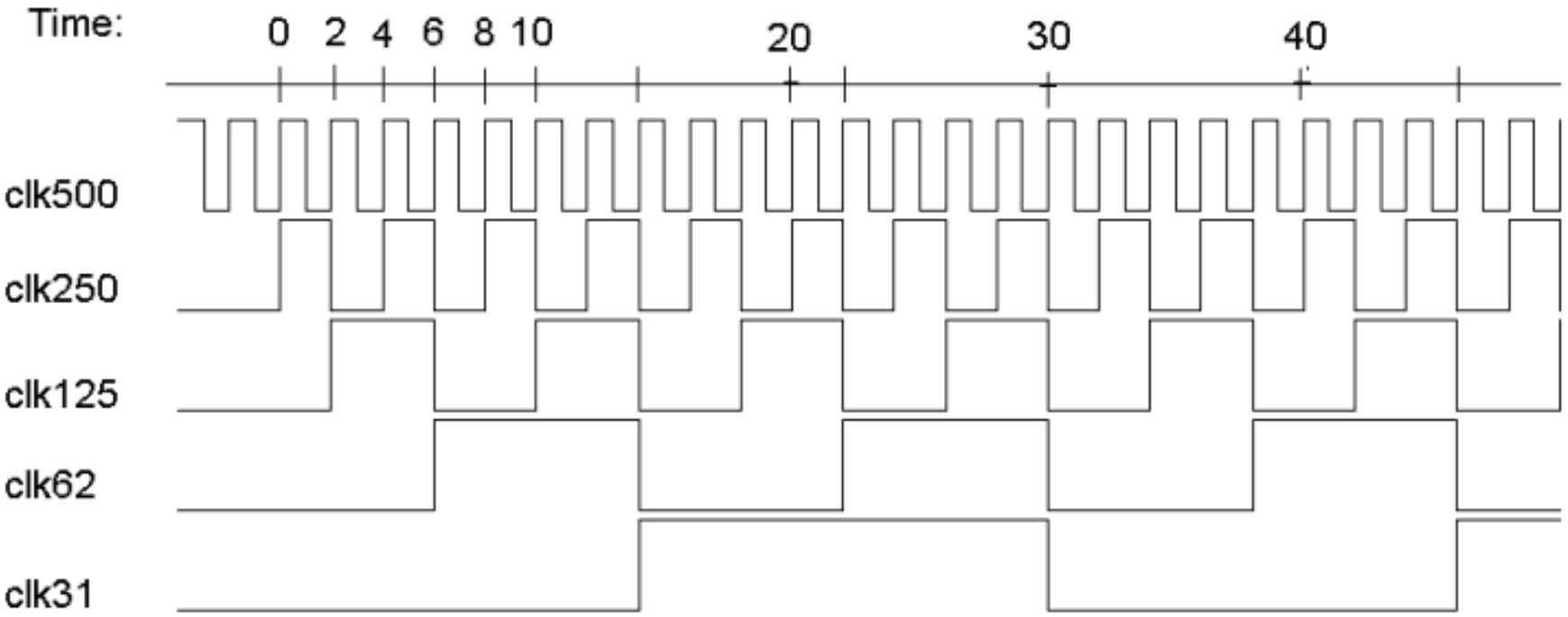
Timing report using -edges (hold)

```
    report_timing -to [get_pins ongen_clk_reg/D] -input_pins -path_type
full_clock -delay min

  Startpoint: onclk_reg (rising edge-triggered flip-flop clocked by clkin)
  Endpoint: ongen_clk_reg
            (rising edge-triggered flip-flop clocked by div2clk)
  Path Group: div2clk
  Path Type: min

  Point                                          Incr       Path
  ---------------------------------------------------------------
  clock clkin (rise edge)                       10.00      10.00
  clock source latency                           0.00      10.00
  clkin (in)                                     0.00      10.00 r
  clktree/A (BUFC)                               0.00      10.00 r
  clktree/Z (BUFC)                               0.12      10.12 r
  onclk_reg/CP (FD1QA)                           0.00      10.12 r
  onclk_reg/CP (FD1QA)                           0.00      10.12 r
  onclk_reg/Q (FD1QA)                            0.34      10.45 f
  ongen_clk_reg/D (FD1QA)                        0.00      10.45 f
  data arrival time                                        10.45

  clock div2clk (rise edge)                      5.00       5.00
  clock source latency                           0.55       5.55
  gen_clk_reg/Q (FD1QA)                          0.00       5.55 r
  ongen_clk_reg/CP (FD1QA)                       0.00       5.55 r
  ongen_clk_reg/CP (FD1QA)                                  5.55 r
  library hold time                              0.16       5.72
  data required time                                        5.72
  ---------------------------------------------------------------
  data required time                                        5.72
  data arrival time                                       -10.45
  ---------------------------------------------------------------
  slack (MET)                                               4.74
```

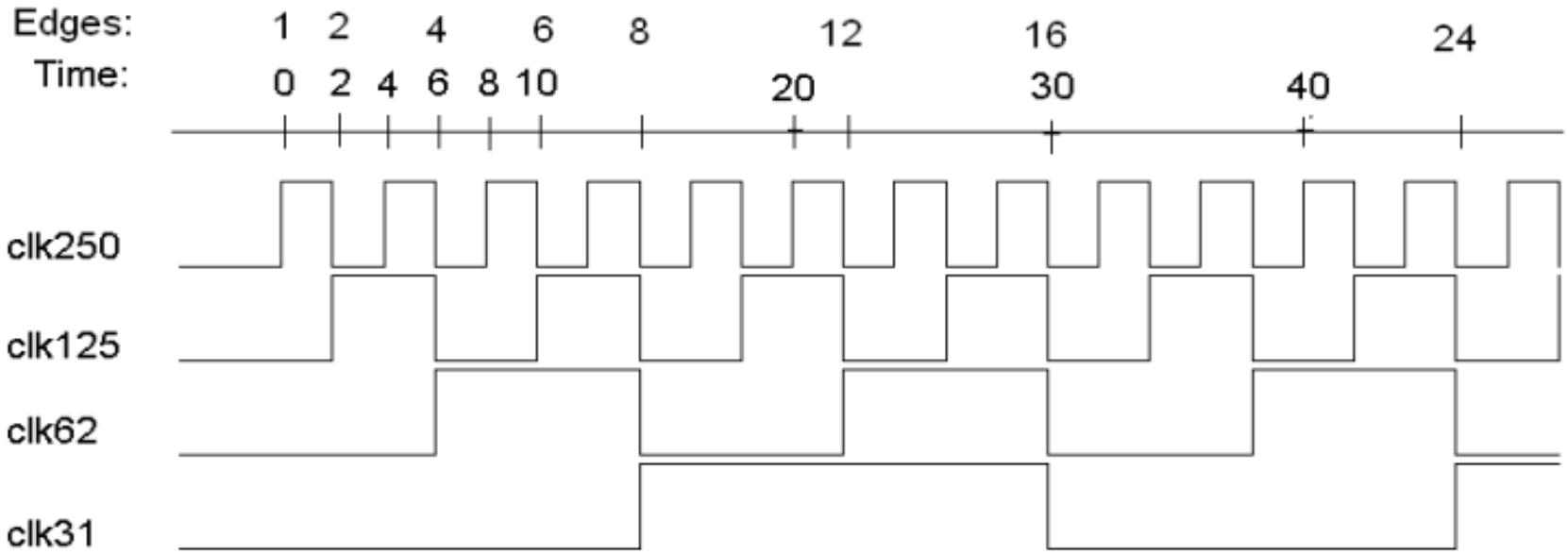
More complex use of -edges

```
always @(posedge clk500 or negedge resetL)
begin
  if (!resetL)
    clk250 <= 0;
  else
    clk250 <= !clk250;
end

always @(negedge clk250 or negedge resetL)  // Note the use of negedge!
begin
  if (!resetL)
    {clk31,clk62,clk125} <= 0;
  else
    {clk31,clk62,clk125} <= {clk31,clk62,clk125} + 1;
end
```

Complex use of -edges - waveform



Complex use of -edges - first try



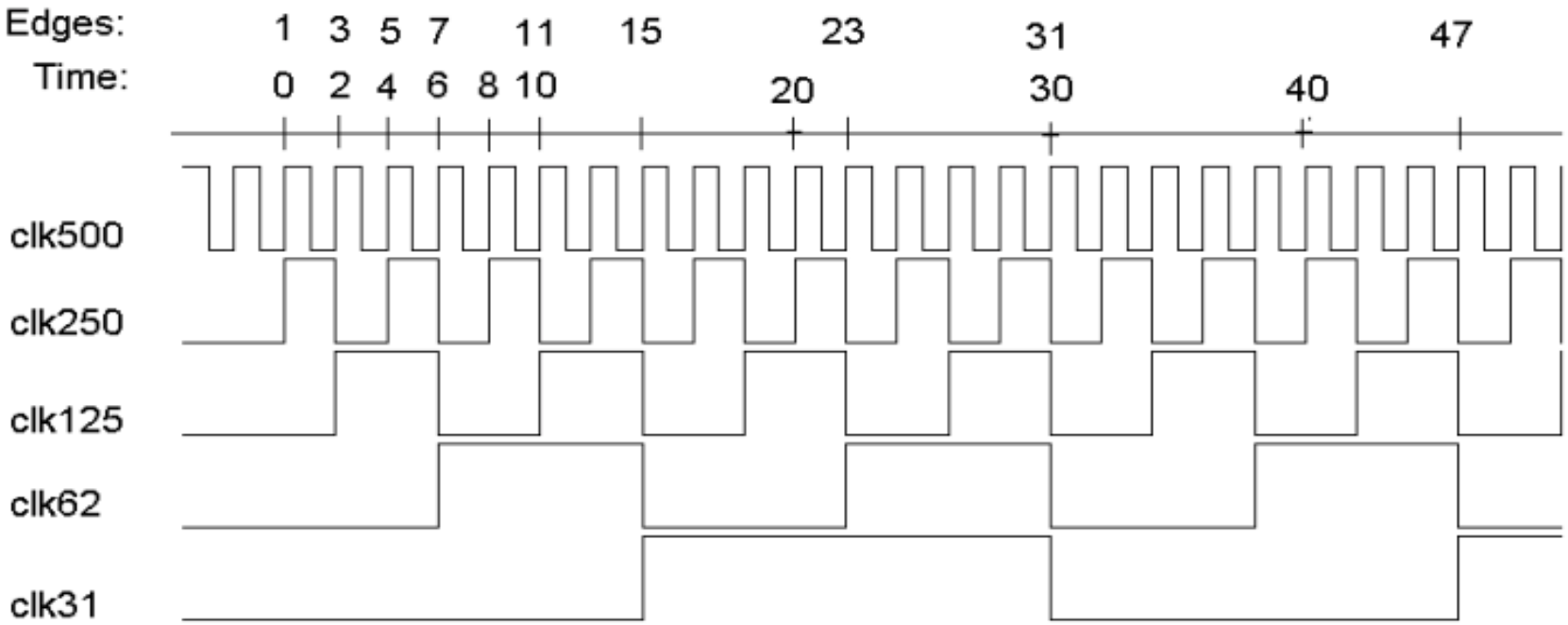
- Results in:

```
clk125   {2  4  6}
clk62    {6  8  12}
clk31    {8  16  24}
```

This is wrong!

- Clk250 is *itself* a generated clock.
- Generated clocks derived from other generated clocks should always be traced back to their non-generated source.

Complex use of -edges - second try



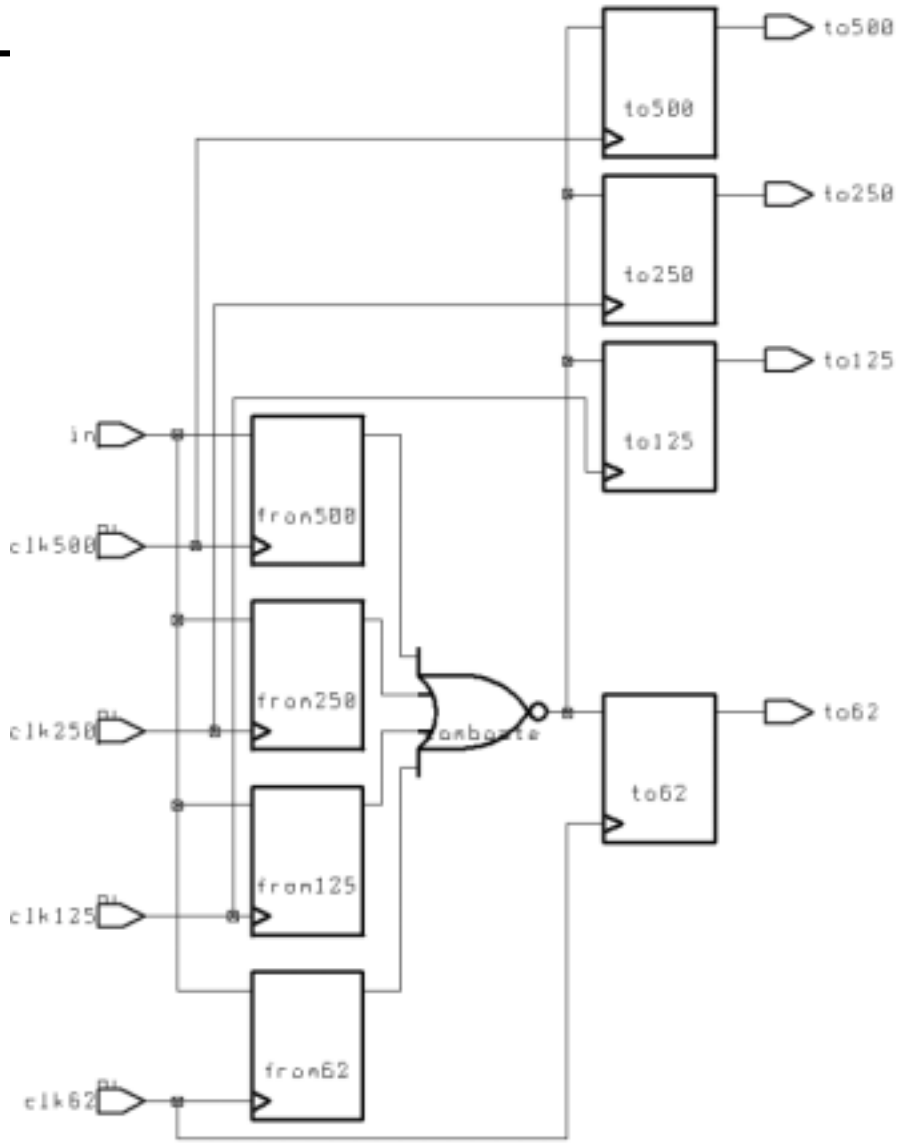
- Results in:

```
clk125   {3 7 11}
clk62    {7 15 23}
clk31    {15 31 47}
```

This is correct!

```
create_clock -period 2.0 [get_ports clk500]
set_propagated_clock clk500

# create the clk250 on the Q pin using -divide_by
create_generated_clock \
    -name clk250 \
    -source [get_ports clk500] \
    -divide_by 2 \
    [get_pins edges/clk250_reg/Q]
;
set_propagated_clock clk250

# create the divided clocks on the Q pins using -edges
create_generated_clock \
    -name clk125 \
    -source [get_ports clk500] \
    -edges {3 7 11} \
    [get_pins edges/clk125_reg/Q]
;
set_propagated_clock clk125
create_generated_clock \
    -name clk62 \
    -source [get_ports clk500] \
    -edges {7 15 23} \
    [get_pins edges/clk62_reg/Q]
;
set_propagated_clock clk62
create_generated_clock \
    -name clk31 \
    -source [get_ports clk500] \
    -edges {15 31 47} \
    [get_pins edges/clk31_reg/Q]
;
set_propagated_clock clk31
```

Complex -edges : test circuit



Timing report - clk250 to clk250

```
report_timing -from sampler/from250_reg/Q -to sampler/to250_reg/D

Startpoint: sampler/from250_reg
            (rising edge-triggered flip-flop clocked by clk250)
Endpoint: sampler/to250_reg
          (rising edge-triggered flip-flop clocked by clk250)
Path Group: clk250
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
clock clk250 (rise edge)                       0.00       0.00
clock network delay (propagated)               0.44       0.44
sampler/from250_reg/CP (FD1QA)                 0.00       0.44 r
sampler/from250_reg/Q (FD1QA) <-               0.37       0.81 f
sampler/combgate/Z (NR4A)                      0.44       1.25 r
sampler/to250_reg/D (FD1QA)                    0.00       1.25 r
data arrival time                                         1.25

clock clk250 (rise edge)                       4.00       4.00
clock network delay (propagated)               0.44       4.44
sampler/to250_reg/CP (FD1QA)                              4.44 r
library setup time                            -0.28       4.16
data required time                                        4.16
-----------------------------------------------------------------
data required time                                        4.16
data arrival time                                        -1.25
-----------------------------------------------------------------
slack (MET)                                               2.91
```

Timing report - clk125 to clk125

```
report_timing -from sampler/from125_reg/Q -to sampler/to125_reg/D

Startpoint: sampler/from125_reg
            (rising edge-triggered flip-flop clocked by clk125)
Endpoint: sampler/to125_reg
          (rising edge-triggered flip-flop clocked by clk125)
Path Group: clk125
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
clock clk125 (rise edge)                       2.00       2.00
clock network delay (propagated)               1.15       3.15
sampler/from125_reg/CP (FD1QA)                 0.00       3.15 r
sampler/from125_reg/Q (FD1QA) <-               0.38       3.53 f
sampler/combgate/Z (NR4A)                      0.41       3.94 r
sampler/to125_reg/D (FD1QA)                    0.00       3.94 r
data arrival time                                         3.94

clock clk125 (rise edge)                      10.00      10.00
clock network delay (propagated)               1.15      11.15
sampler/to125_reg/CP (FD1QA)                             11.15 r
library setup time                            -0.27      10.88
data required time                                       10.88
-----------------------------------------------------------------
data required time                                       10.88
data arrival time                                        -3.94
-----------------------------------------------------------------
slack (MET)                                               6.94
```

Timing report - clk125 to clk62

```
  report_timing -from sampler/from125_reg/Q -to sampler/to62_reg/D

Startpoint: sampler/from125_reg
            (rising edge-triggered flip-flop clocked by clk125)
Endpoint: sampler/to62_reg
          (rising edge-triggered flip-flop clocked by clk62)
Path Group: clk62
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
clock clk125 (rise edge)                       2.00       2.00
clock network delay (propagated)               1.15       3.15
sampler/from125_reg/CP (FD1QA)                 0.00       3.15 r
sampler/from125_reg/Q (FD1QA) <-               0.38       3.53 f
sampler/combgate/Z (NR4A)                      0.41       3.94 r
sampler/to62_reg/D (FD1QA)                     0.00       3.94 r
data arrival time                                         3.94

clock clk62 (rise edge)                        6.00       6.00
clock network delay (propagated)               1.10       7.10
sampler/to62_reg/CP (FD1QA)                               7.10 r
library setup time                            -0.27       6.83
data required time                                        6.83
-----------------------------------------------------------------
data required time                                        6.83
data arrival time                                        -3.94
-----------------------------------------------------------------
slack (MET)                                               2.88
```

Timing report - clk125 to clk62 (hold)

```
    report_timing -from sampler/from125_reg/Q -to sampler/to62_reg/D -delay
min

  Startpoint: sampler/from125_reg
              (rising edge-triggered flip-flop clocked by clk125)
  Endpoint: sampler/to62_reg
            (rising edge-triggered flip-flop clocked by clk62)
  Path Group: clk62
  Path Type: min

  Point                                          Incr       Path
  -----------------------------------------------------------------
  clock clk125 (rise edge)                      10.00      10.00
  clock network delay (propagated)               1.15      11.15
  sampler/from125_reg/CP (FD1QA)                 0.00      11.15 r
  sampler/from125_reg/Q (FD1QA) <-               0.38      11.53 r
  sampler/combgate/Z (NR4A)                      0.30      11.83 f
  sampler/to62_reg/D (FD1QA)                     0.00      11.83 f
  data arrival time                                        11.83

  clock clk62 (rise edge)                        6.00       6.00
  clock network delay (propagated)               1.10       7.10
  sampler/to62_reg/CP (FD1QA)                               7.10 r
  library hold time                              0.17       7.27
  data required time                                        7.27
  -----------------------------------------------------------------
  data required time                                        7.27
  data arrival time                                       -11.83
  -----------------------------------------------------------------
  slack (MET)                                               4.56
```

Topics covered in paper

- Clock Muxes / Managing Large Numbers of Clocks
- I/O Interfaces with Outgoing Clocks
- Describing complex clocking relationships using -edges
- **Phantom Delays on input and output ports**
- Parsing command output for use in a script

Phantom Delays on Input and Output Ports

- You load up the netlist, read in the sdf, and do report_timing from some input:

```
report_timing -to f1_reg/D

Startpoint: in (input port)
Endpoint: f1_reg (rising edge-triggered flip-flop clocked by clkin)
Path Group: (none)
Path Type: max

Point                                    Incr       Path
-----------------------------------------------------------
input external delay                     0.00       0.00 r
in (in)                                  0.00       0.00 r
inpad/Z (BUFB)                           1.14 *     1.14 r
f1_reg/D (FD1QA)                         0.10 *     1.24 r
data arrival time                                   1.24
```

This should be 0.5!

What's wrong?

- The "*" doesn't mean back-annotated, it means "all or part is back-annotated"!
- Do the report_timing with -input pins:

```
report_timing -to f1_reg/D  -input_pins

Startpoint: in (input port)
Endpoint: f1_reg (rising edge-triggered flip-flop clocked by clkin)
Path Group: (none)
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
input external delay                           0.00       0.00 r
in (in)                                        0.00       0.00 r
inpad/A (BUFB)                                 0.64       0.64 r
inpad/Z (BUFB)                                 0.50 *     1.14 r
f1_reg/D (FD1QA)                               0.10 *     1.24 r
data arrival time                                         1.24
```

There's the 0.5 (with a "*"), but what's that 0.64?

Fixing this using set_resistance

```
set_resistance 0.0 [get_nets in]
```

- Works better than set_annotated_delay because it works through hierarchies (see paper for details).

Timing report after *set_resistance*

```
report_timing -to f1_reg/D -input_pins

Startpoint: in (input port)
Endpoint: f1_reg (rising edge-triggered flip-flop clocked by clkin)
Path Group: (none)
Path Type: max

Point                                          Incr       Path
-----------------------------------------------------------------
input external delay                           0.00       0.00 r
in (in)                                        0.00       0.00 r
pads/in (pads)                                 0.00       0.00 r
pads/inpad/A (BUFB)                            0.00       0.00 r
pads/inpad/Z (BUFB)                            0.50 *     0.50 r
pads/in_buf (pads)                             0.00 *     0.50 r
f1_reg/D (FD1QA)                               0.10 *     0.60 r
data arrival time                                         0.60
```

Interconnect delay now zero.

Topics covered in paper

- Clock Muxes / Managing Large Numbers of Clocks
- I/O Interfaces with Outgoing Clocks
- Describing complex clocking relationships using -edges
- Phantom Delays on input and output ports
- **Parsing command output for use in a script**

Parsing command output for a value

- Ever want to use a report value in your script?
- get_timing_paths works fine for timing values, but what about other report output?
- It can be done!

Parsing command output - basic flow

- Put report output to a file.
- Execute some shell commands on the file to produce a file that looks like PrimeTime commands ("set variable value").
- Source the file.

Parsing command output – tcl code

```
# default the variable to zero
set _unconstrained_endpoints 0

set _tempfile "temp[pid]"
set _pt_tempfile "temp[pid].pt"

set _cmd {#!/bin/sh
  cat <<END | perl -e '

  # Loop over the output looking for lines that match:
  #   Warning: There <some_word> <count> <.*not constrained for maximum delay>
  # The <some_word> handles "is" or "are".
  while (<>) {
     if (/Warning: There \S+ (\S+) .*not constrained for maximum delay/) {
       # Print out the "set" command
       print "set _unconstrained_endpoints $1\n";
     }
   }
   '
}

# Build the executable file
echo $_cmd > $_tempfile
check_timing >> $_tempfile
echo "END" >> $_tempfile
# Make it executable and execute it.
sh chmod +x $_tempfile
sh ./$_tempfile > $_pt_tempfile

# source the resulting output
source $_pt_tempfile
```

temp file result

```
cr$ cat temp18717
#!/bin/sh
    cat <<END | perl -e '

    # Loop over the output looking for lines that match:
    #   Warning: There <some_word> <count> <.*not constrained for maximum delay>
    # The <some_word> handles "is" or "are".
    while (<>) {
       if (/Warning: There \S+ (\S+) .*not constrained for maximum delay/) {
         # Print out the "set" command
         print "set _unconstrained_endpoints $1\n";
       }
     }
     '
```

My perl script

Actual check_timing output

```
Warning: There are 2 endpoints which are not constrained for maximum delay.

0
END
```

Tells shell that input is complete

temp.pt file result

```
cr$ cat temp18717.pt
set _unconstrained_endpoints 2
```

Conclusion

- Complex clocking situations involving muxes, divided clocks, falling edges, outgoing clocks, and combinations of these do occur in real chips, and PrimeTime can be used to time them correctly – *if you know the tricks*!