

Cell Library and Simulation

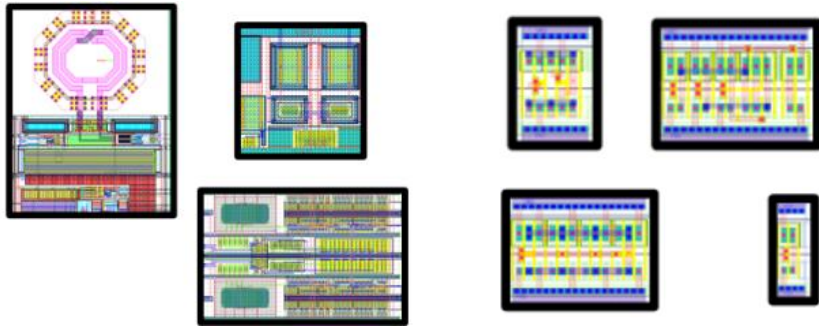
Jaeha Kim, Sung-Jun Lee, and Eunseo Kim
Mixed-Signal IC and System Group
Seoul National University
May 19. 2016

Build Mixed-Signal System like LEGO

Cell Library

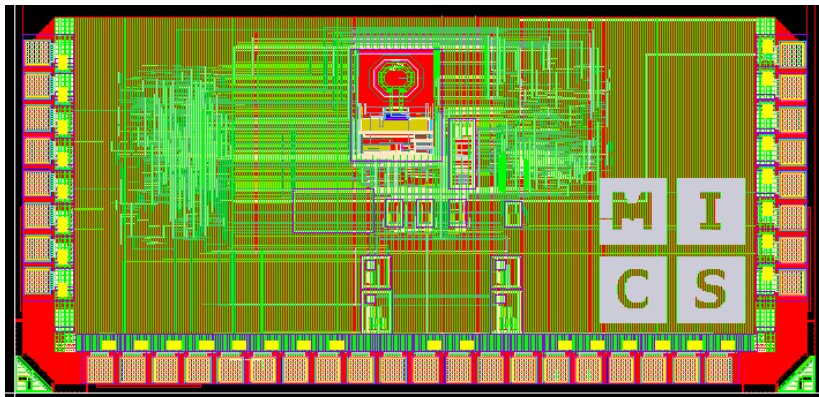
Analog

Digital



- Analog : PFD, DCO, divider, ...
- Digital : INV, DFF, MUX, ...

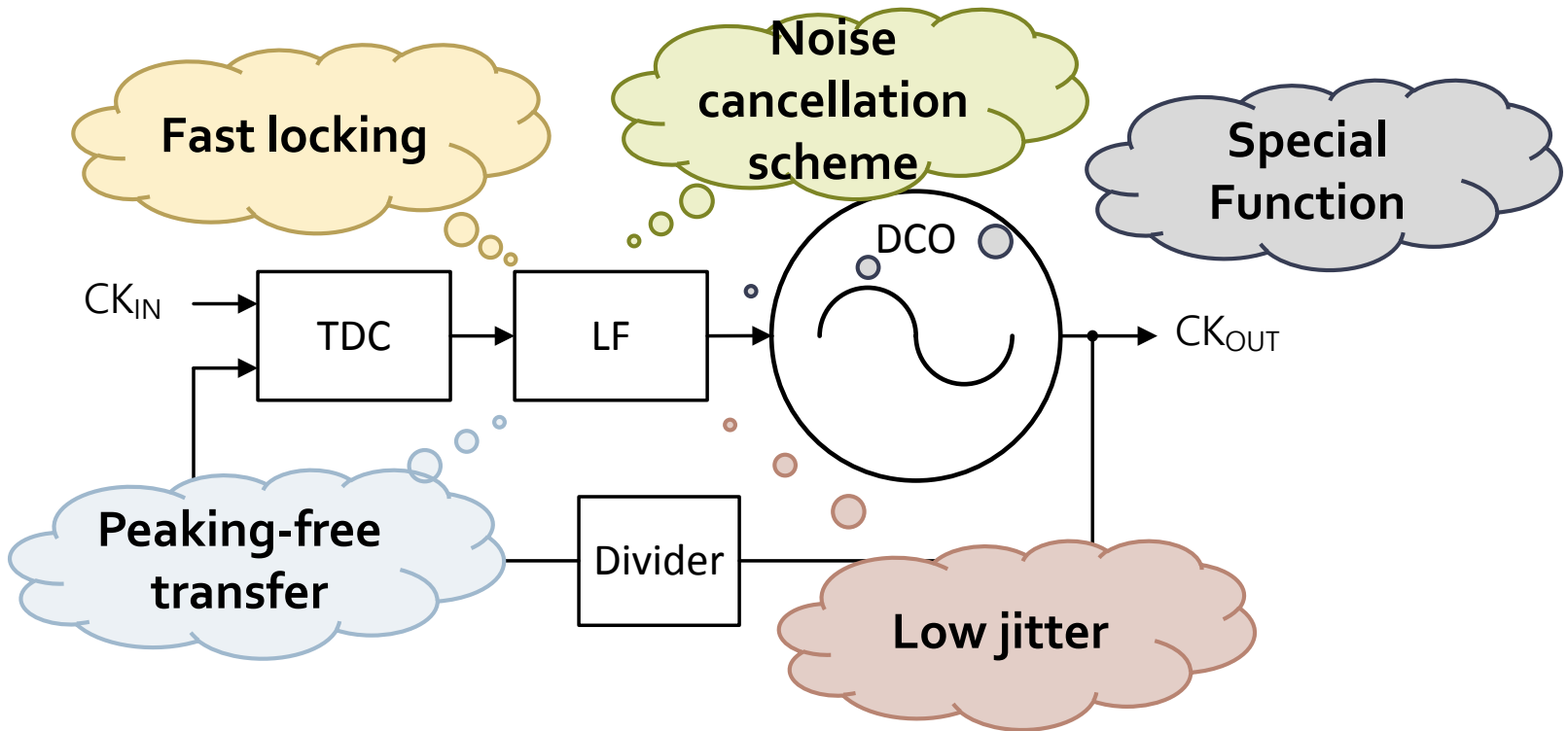
Final Chip



Cell Library Establishment

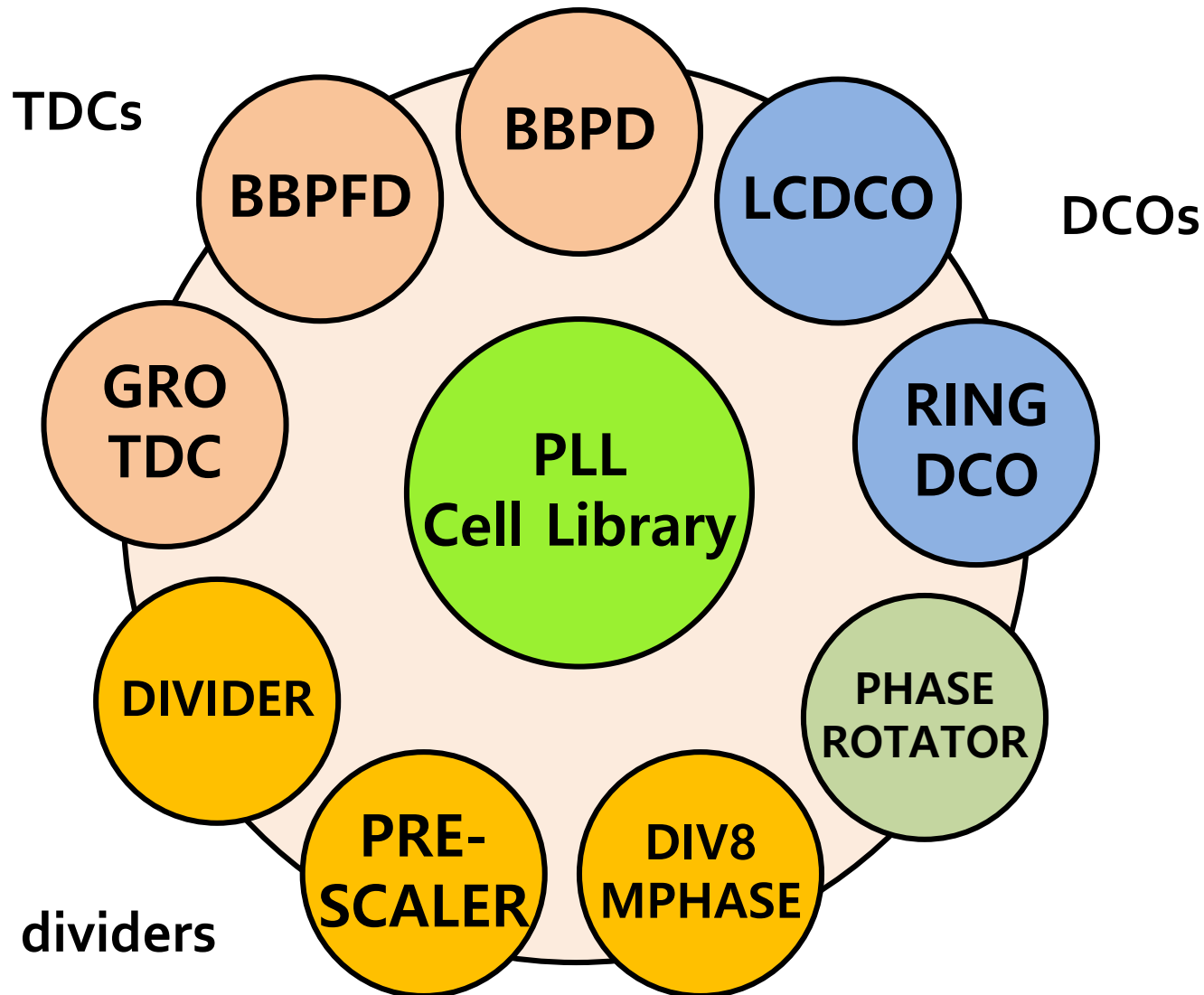
- ▶ We provide PLL cell library to enable quick design of PLL
 - ▶ No manual custom cell design
 - ▶ Automatic P&R
 - ▶ Designing analog and digital circuits at the same level
- ▶ In this course, the provided cell library is in **generic technology**
(lambda-based metric, missing confidential technology specific parameters)
- ▶ Our next plan is *to establish well verified PLL cell library in Samsung 65-nm CMOS process (IDEC MPW)*
 - ▶ Everyone can use it for free

The Cell Library is for...

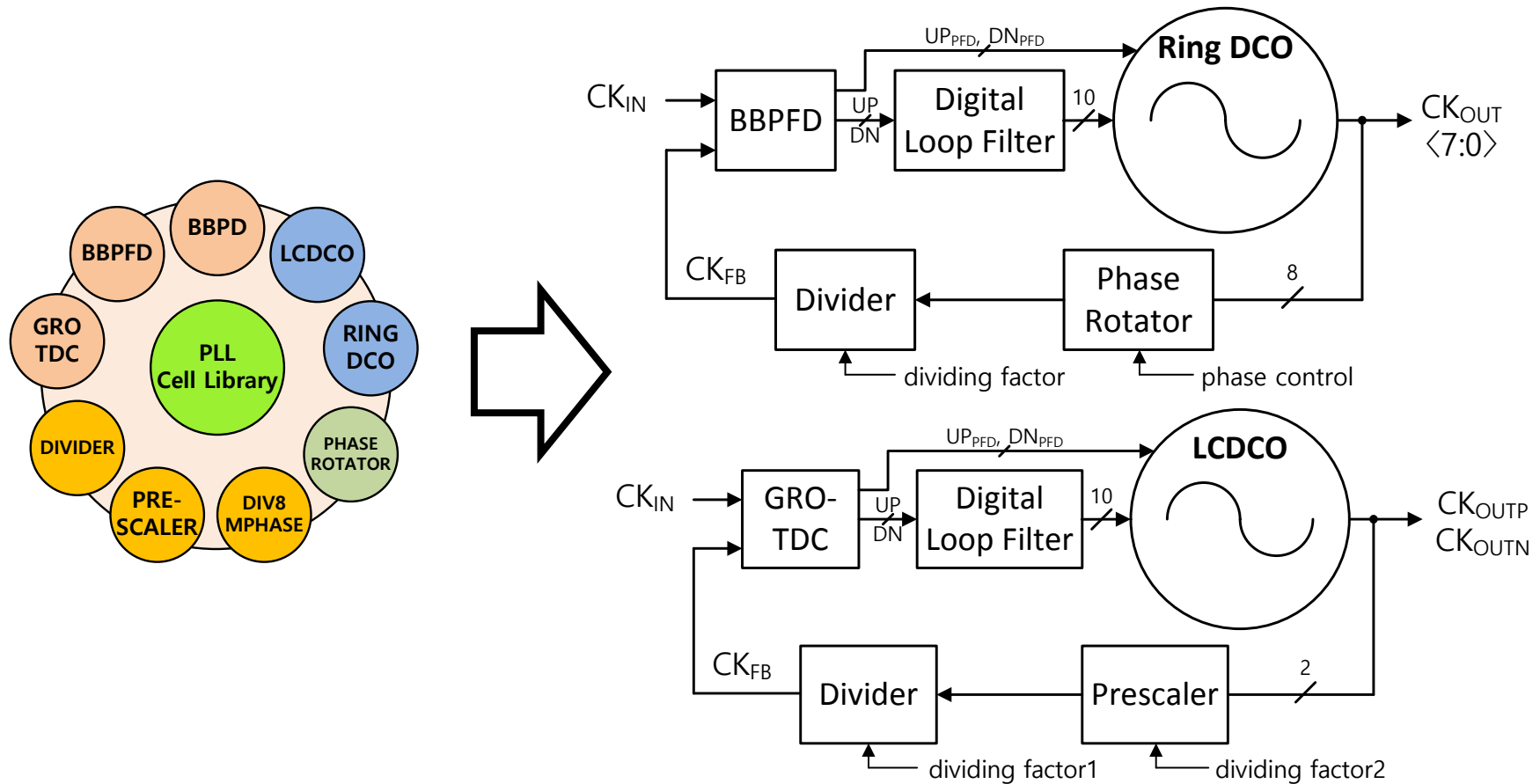


- ▶ A person who may not aware of detail (analog) circuit topologies can make PLLs without difficulty
- ▶ **Quick realization** of novel idea in loop filter

PLL Cell Library Composition



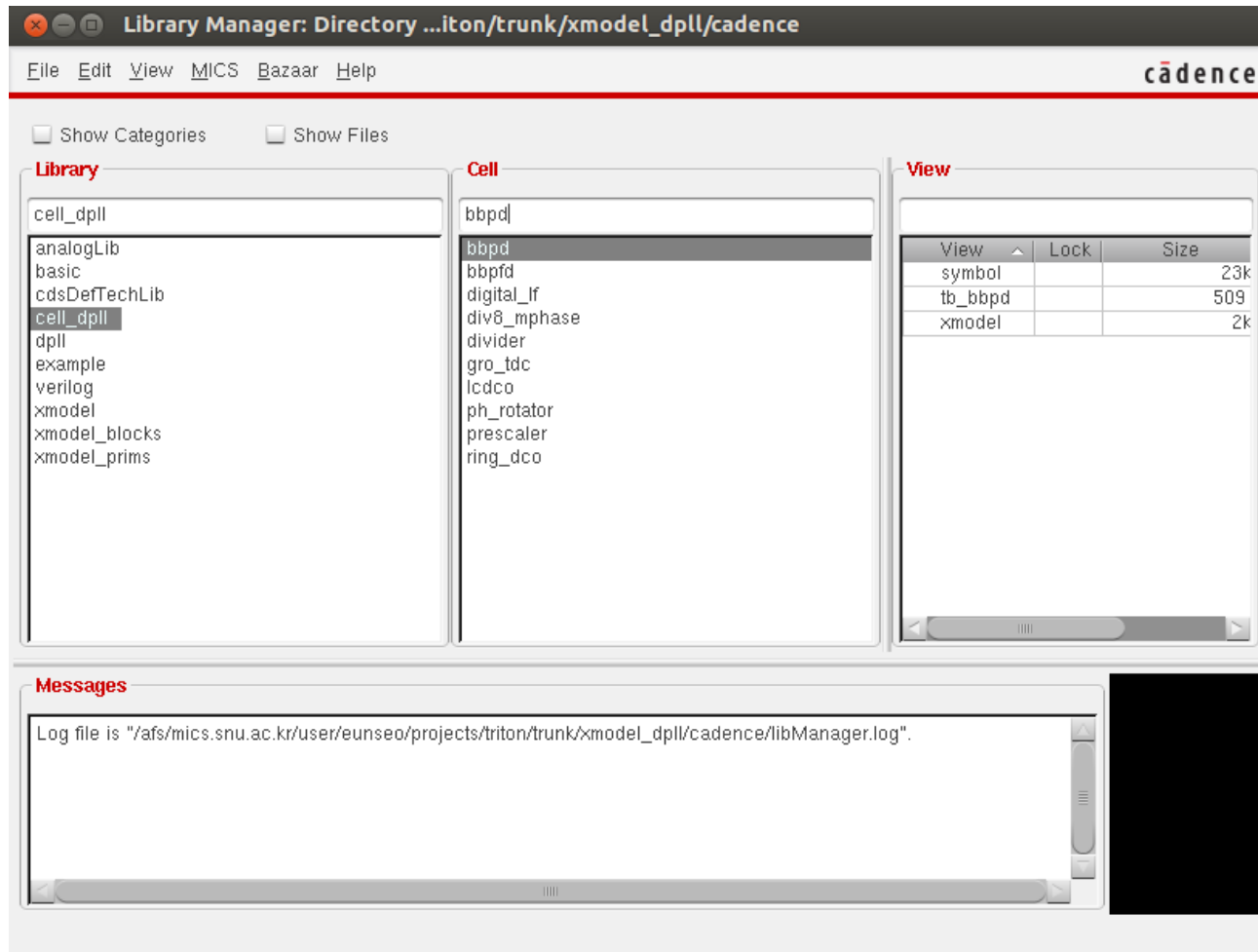
Various PLLs are Realizable



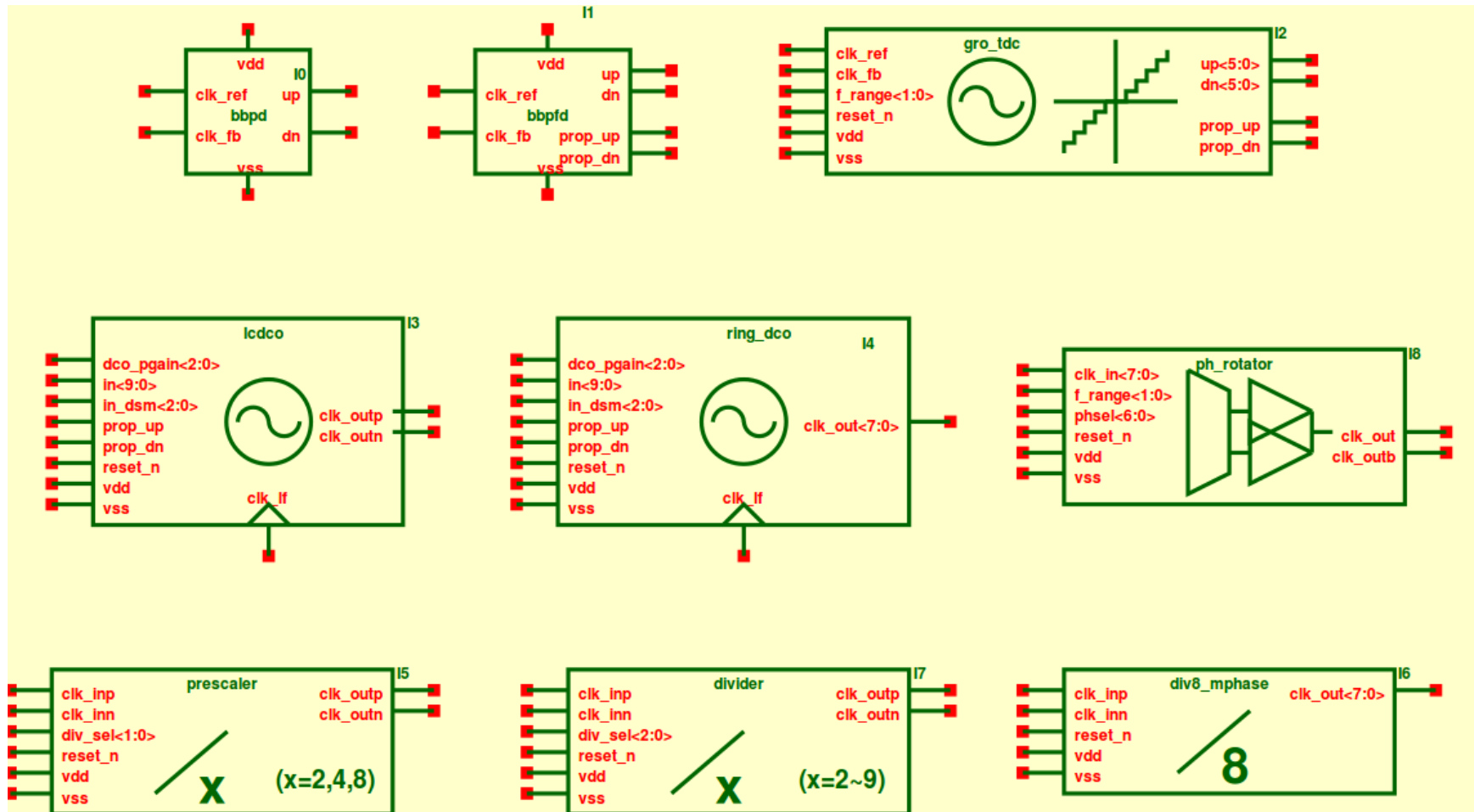
- ▶ Various PLL topologies are realizable just by blocking together the analog cells

Cell Library in GLISTER

- ▶ 'cell_dp11' contains PLL cell library models



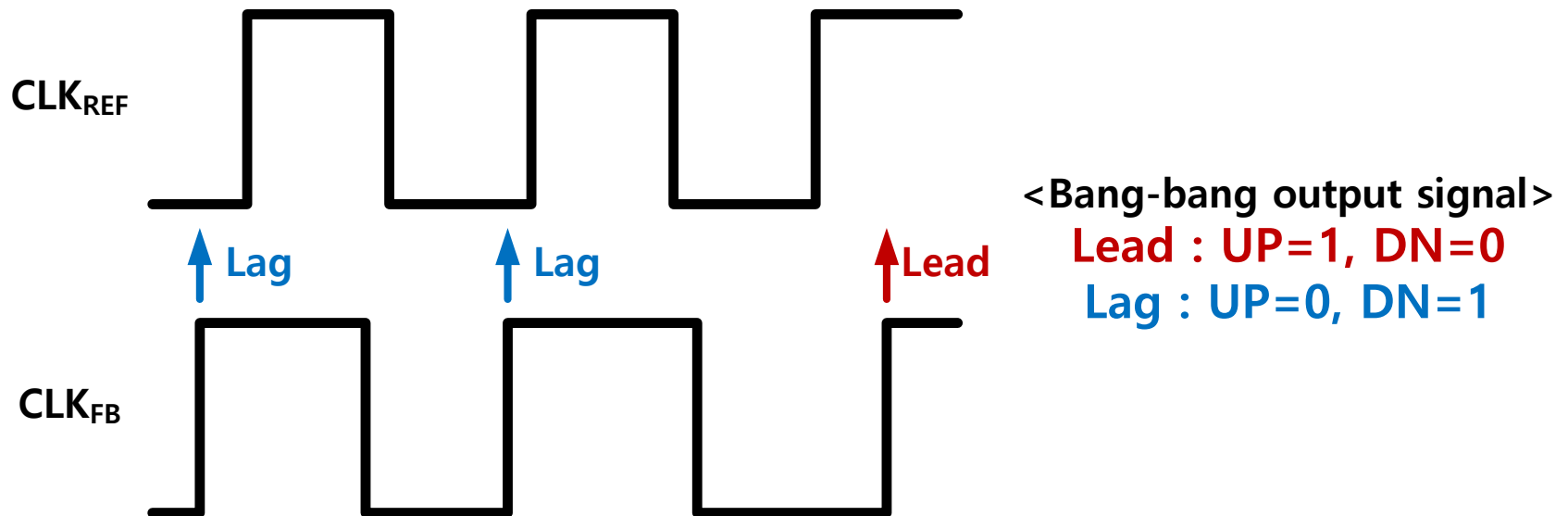
Cell Library in GLISTER (2)



Let's Get Started with PLL Cell Library

Bang-Bang Phase Detector

- ▶ compares phases of two input clocks and expresses the result with bang-bang up and down output signals (digital output)
- ▶ The output is synchronized to the feedback clock



Testbench for BBPD

▶ Testbench for BBPD (cell_dp11/bbpd/tb_bbpd)

```
`include "xmodel.h"
```

```
module tb_bbpd();
```

```
    parameter real freq_ref = 130.0e6;
```

```
    parameter real freq_fb = 125.0e6;
```

Two clocks having
slightly different frequency

```
    xbit clk_ref, clk_fb; reg vdd = 1; reg vss = 0;
```

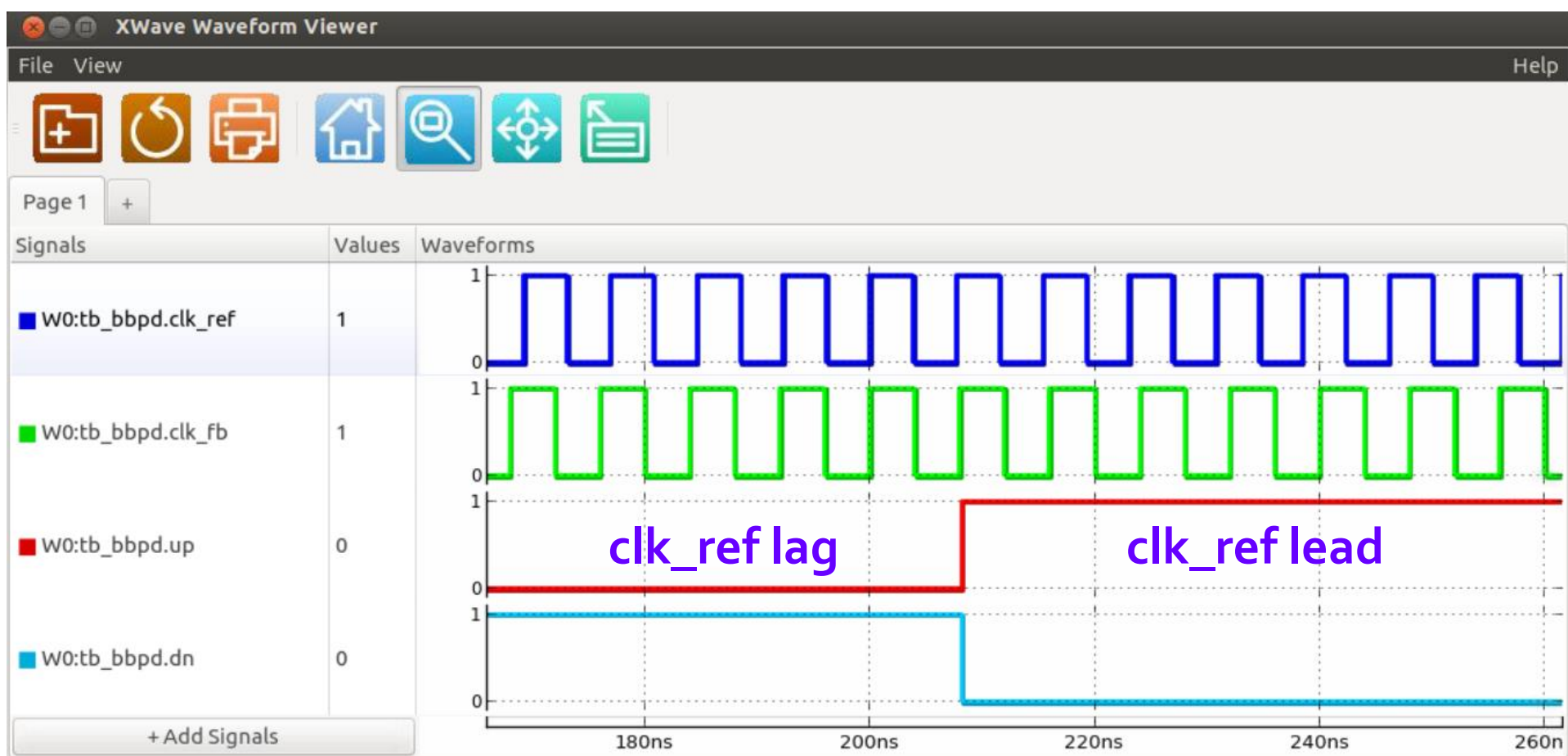
```
    clk_gen #(.freq(freq_ref)) clk_gen_ref(clk_ref);
```

```
    clk_gen #(.freq(freq_fb)) clk_gen_fb(clk_fb);
```

Clock generation

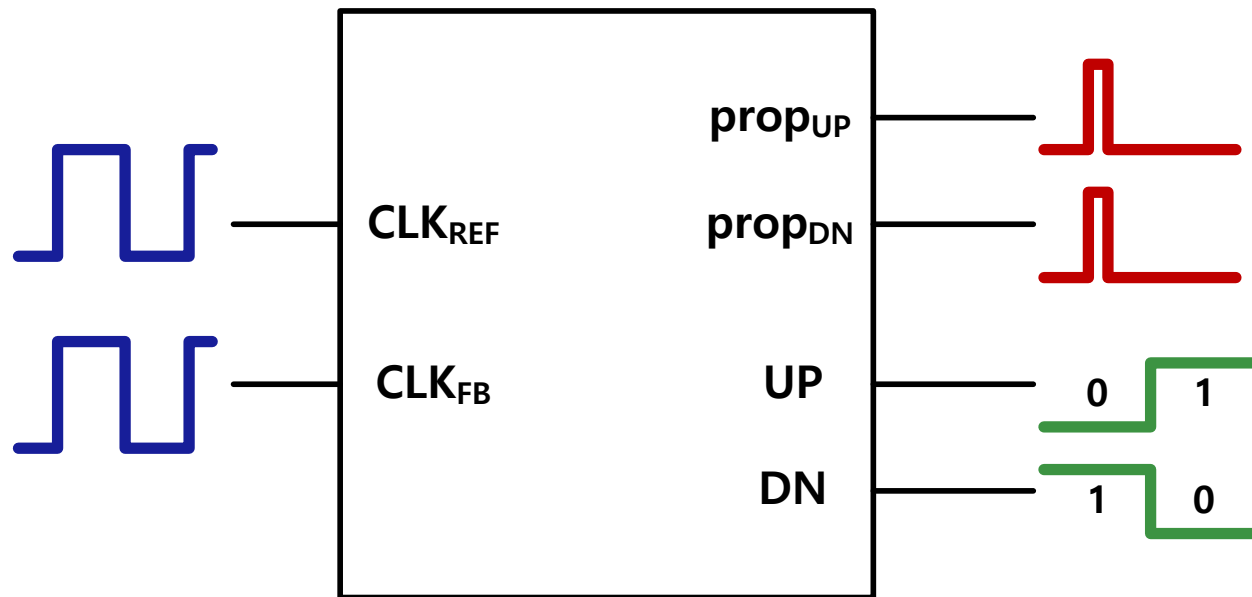
Exercise: BBPD

► cell_dpll/bbpd/tb_bbpd

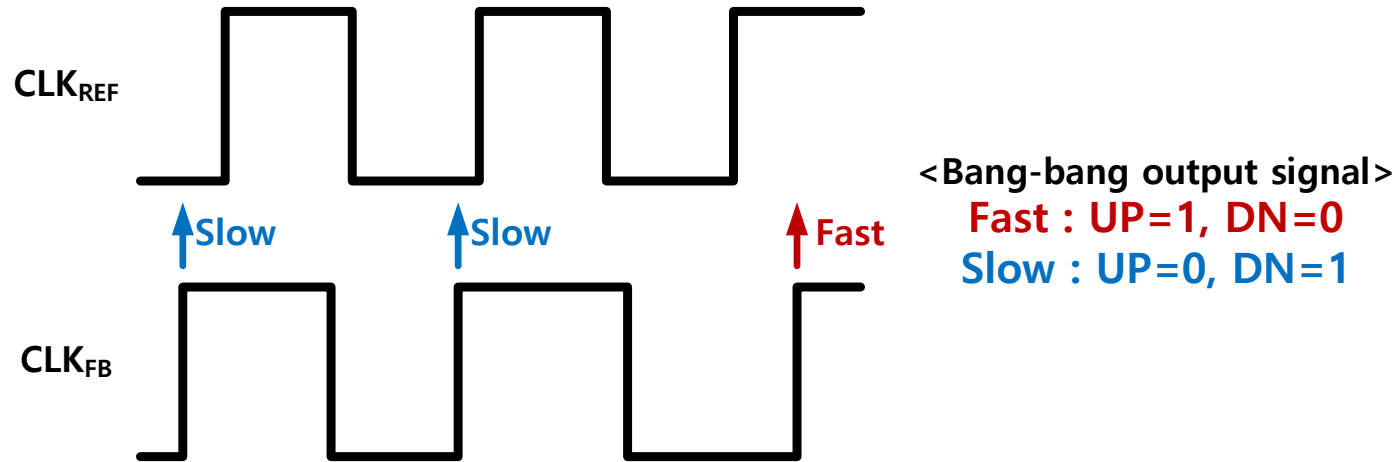


Bang-Bang PFD

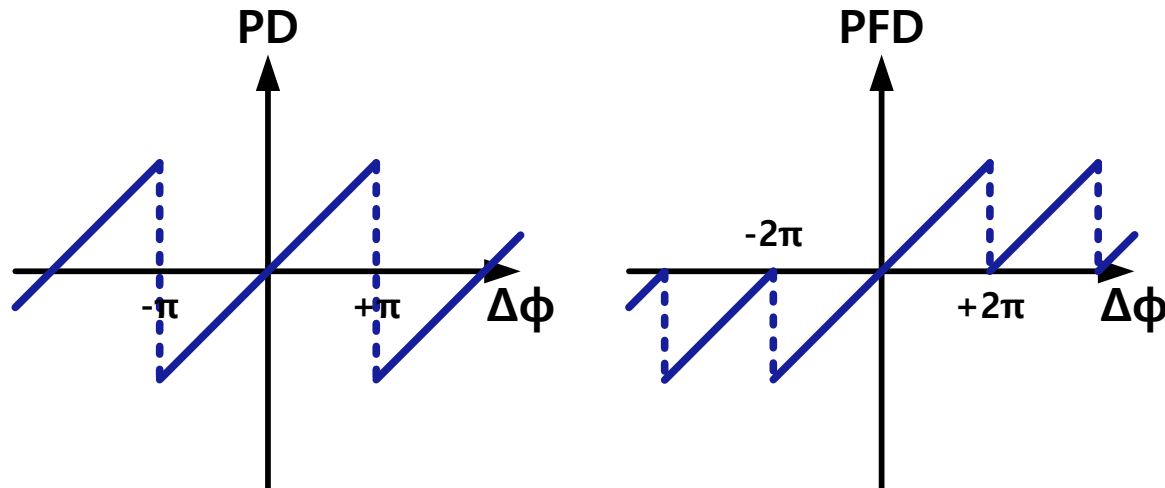
- ▶ compares phase and frequency of two input clocks and expresses the result with bang-bang up and down output signals (digital output)
- ▶ It also has PFD output (pulse width) **Why?**



BBPFD Operation

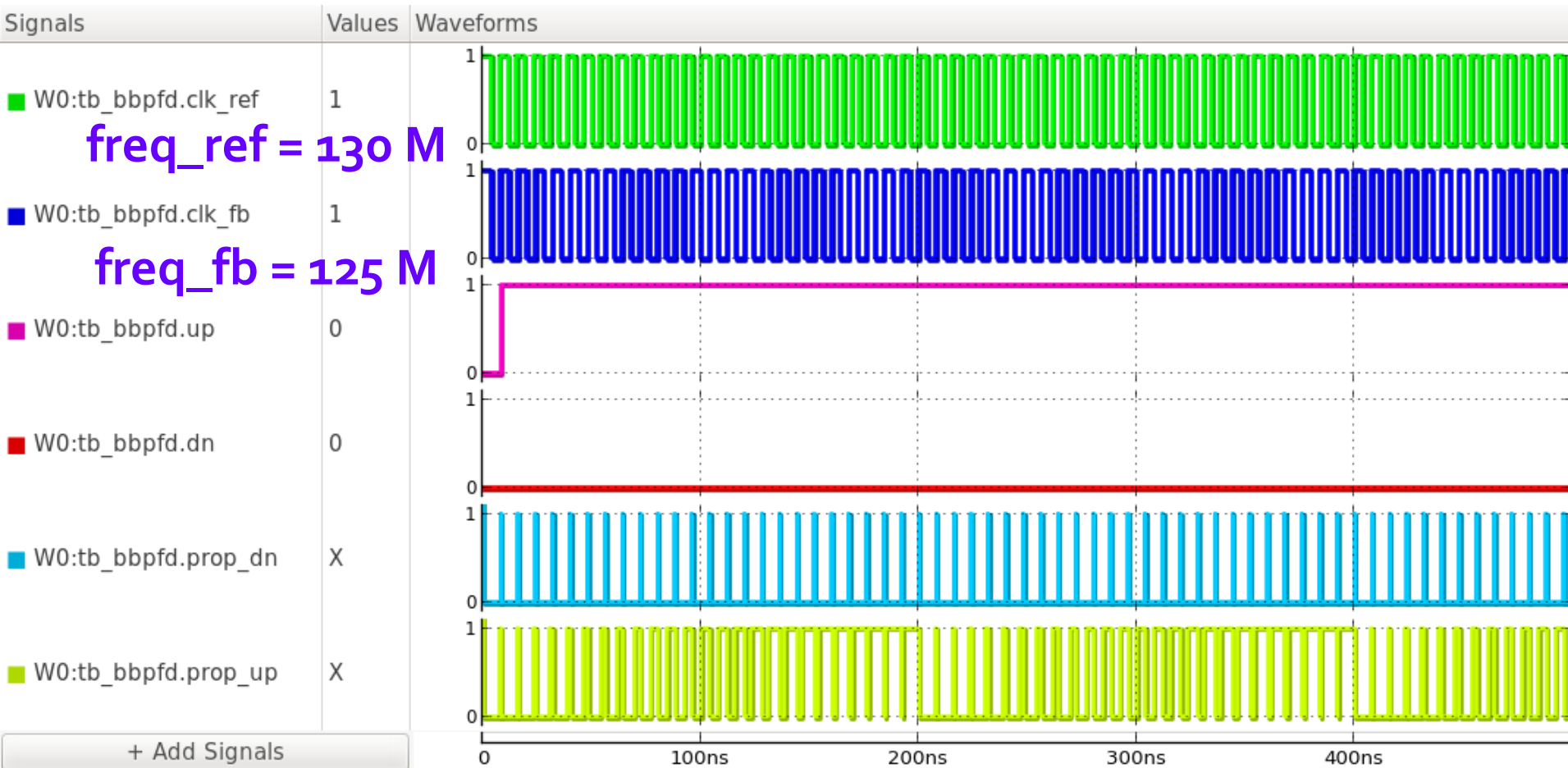


► Difference between PD and PFD



Exercise: BBPFD

► cell_dpll/bbpfd/tb_bbpfd

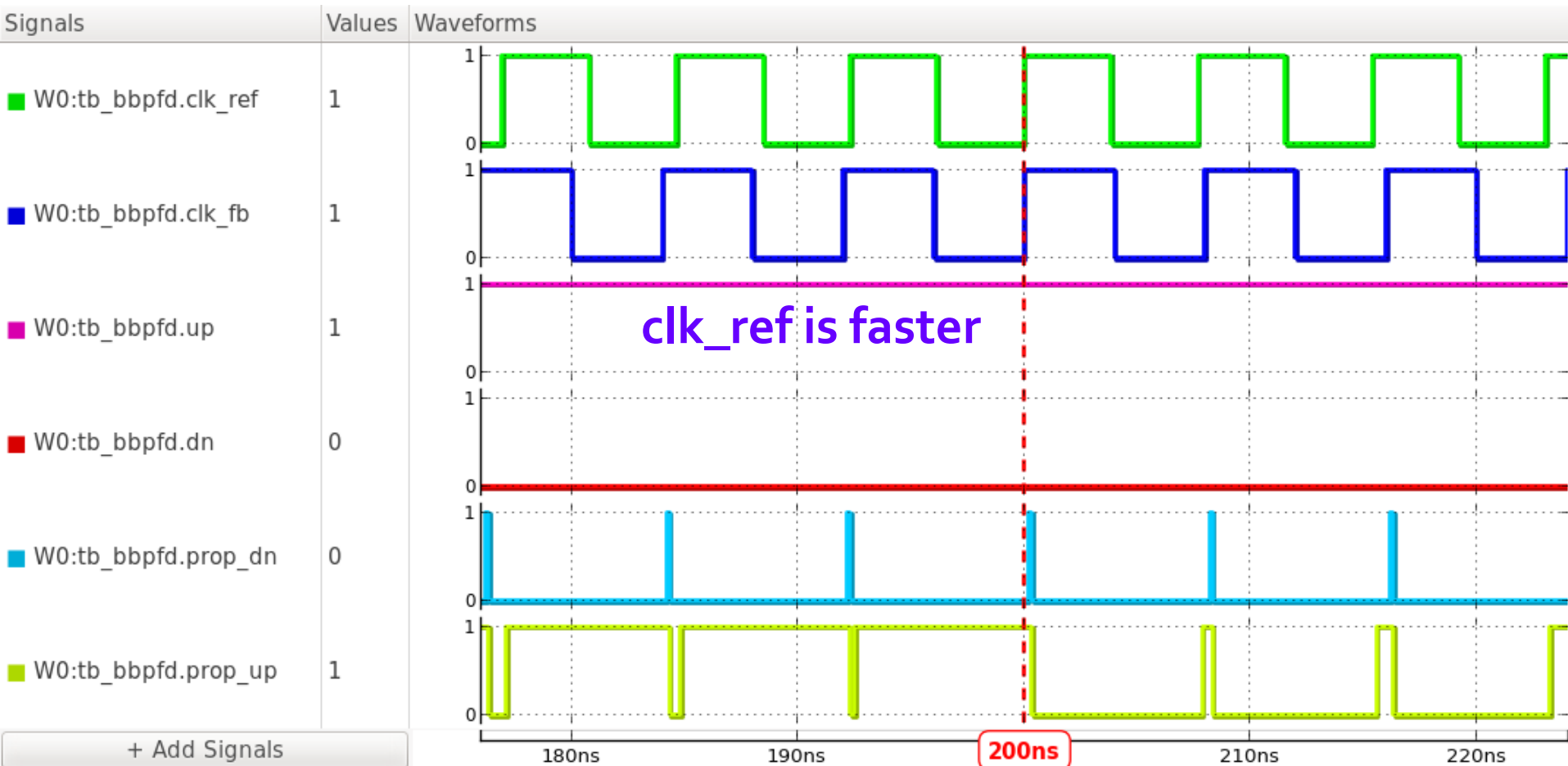


freq_ref = 130 M

freq_fb = 125 M

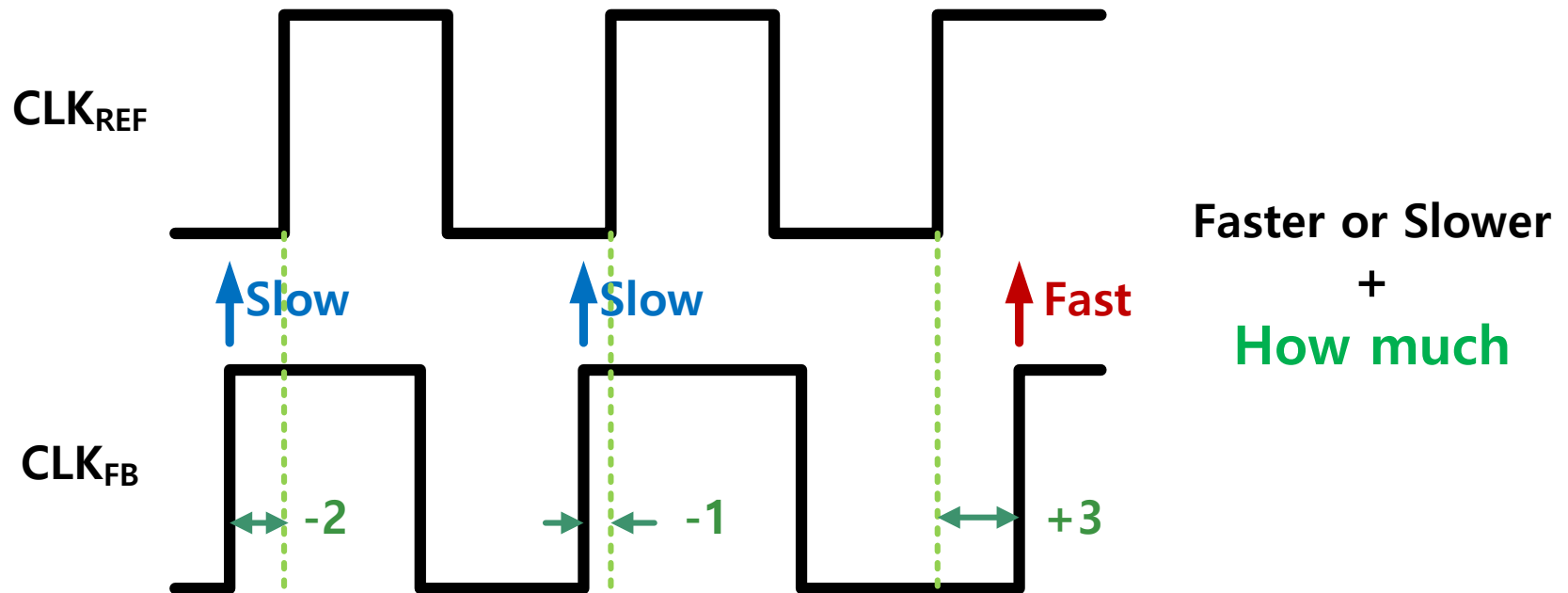
Exercise: BBPFD

- ▶ Once the reference clock is faster than the feedback clock, it maintains up=1 (dn=0) output



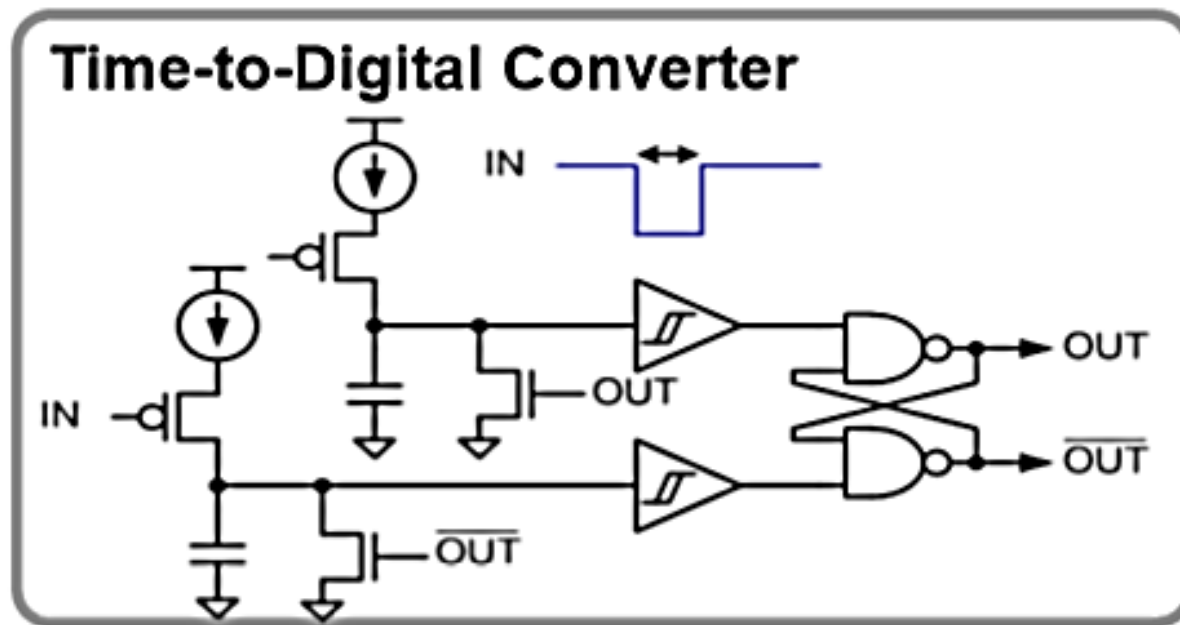
Gated (Relaxation) Oscillator TDC

- measures phase and frequency difference between two input clocks and expresses the timing error with digital code output

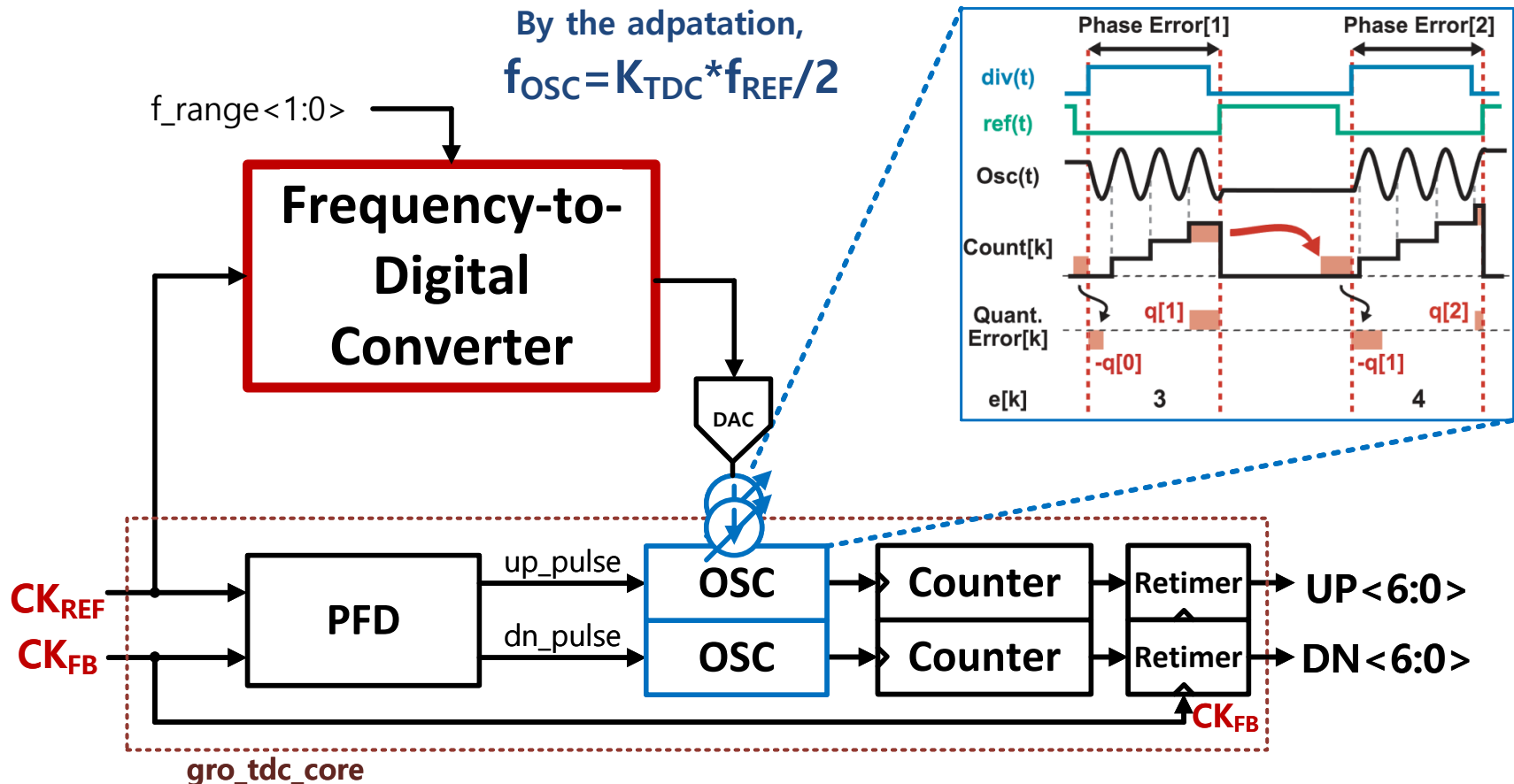


What is Relaxation Oscillator?

- ▶ Integrate and fire (I&F) structure
 - ▶ The capacitor integrates the current
 - ▶ If the voltage level reaches to a certain threshold voltage, it fires to trigger integration of the other side capacitor



GRO_TDC with Constant K_{TDC}



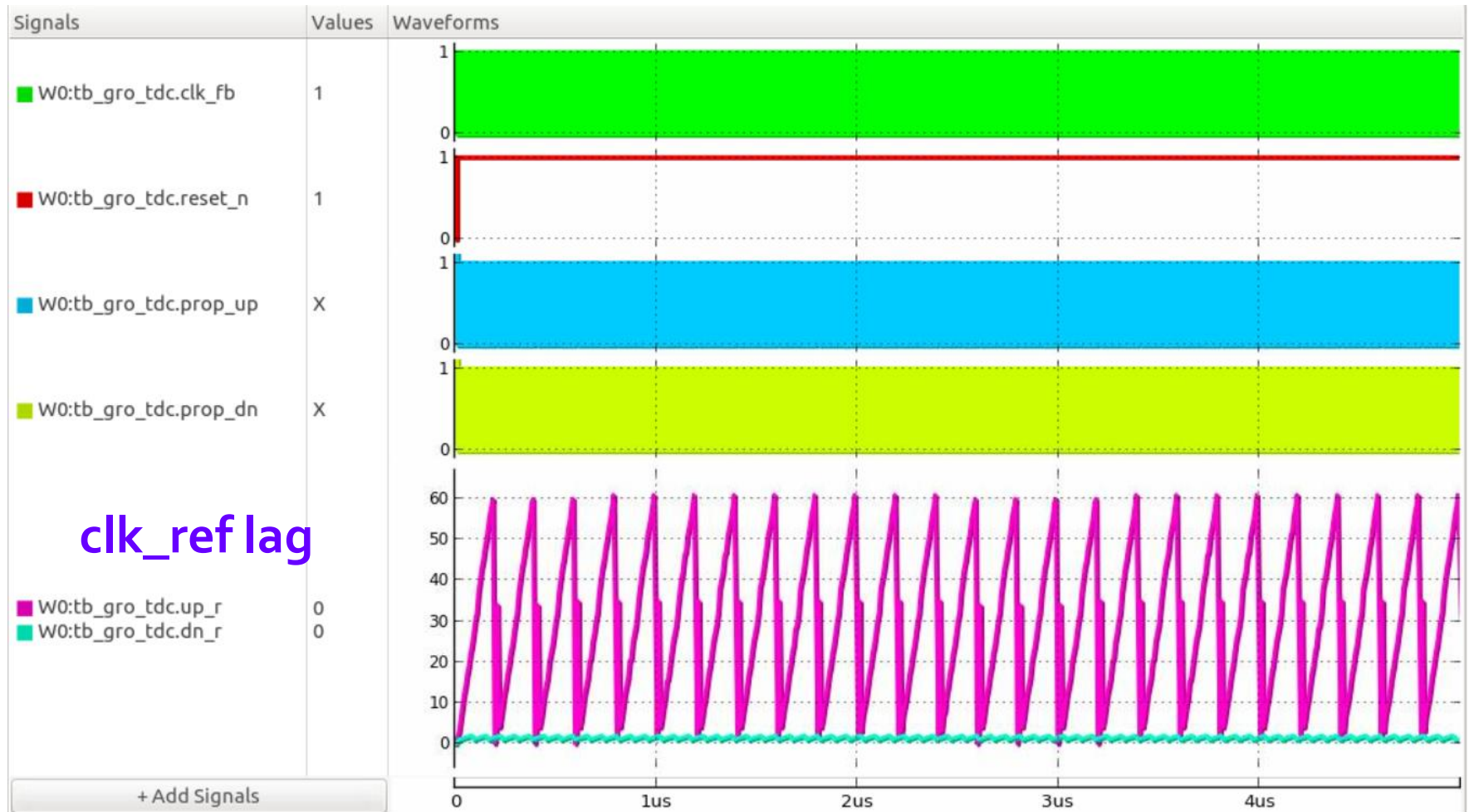
- ▶ The frequency adaptation loop empowers the TDC to have constant K_{TDC} over wide operation range

Exercise: GRO_TDC

► cell_dpll/gro_tdc/tb_gro_tdc

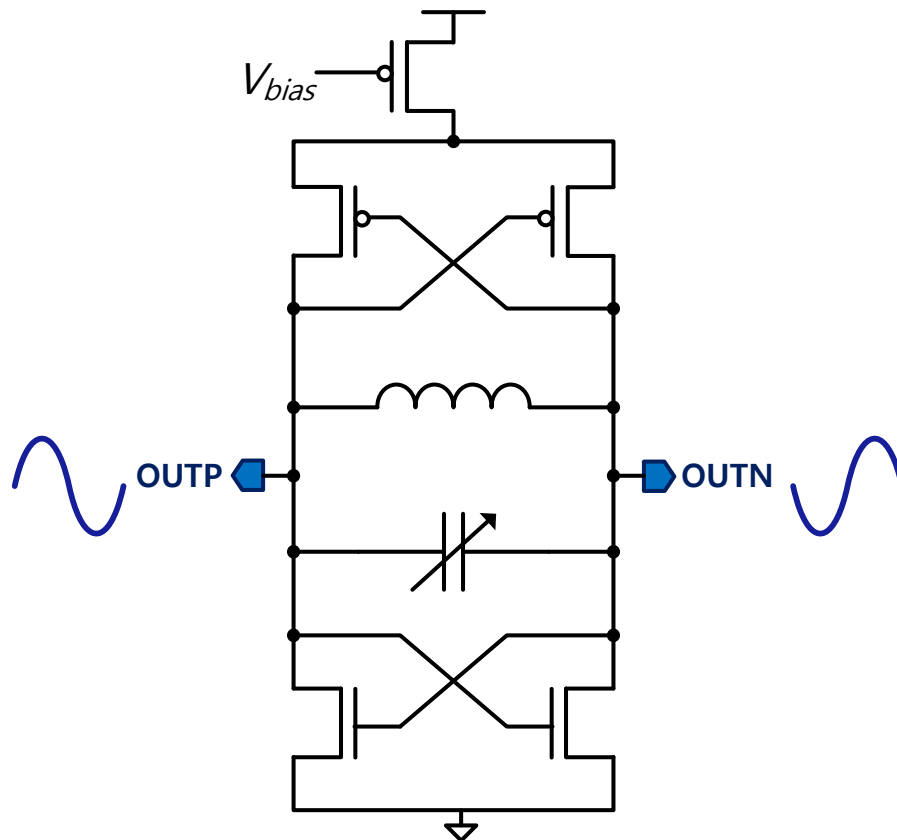
freq_ref = 130 M

freq_fb = 125 M



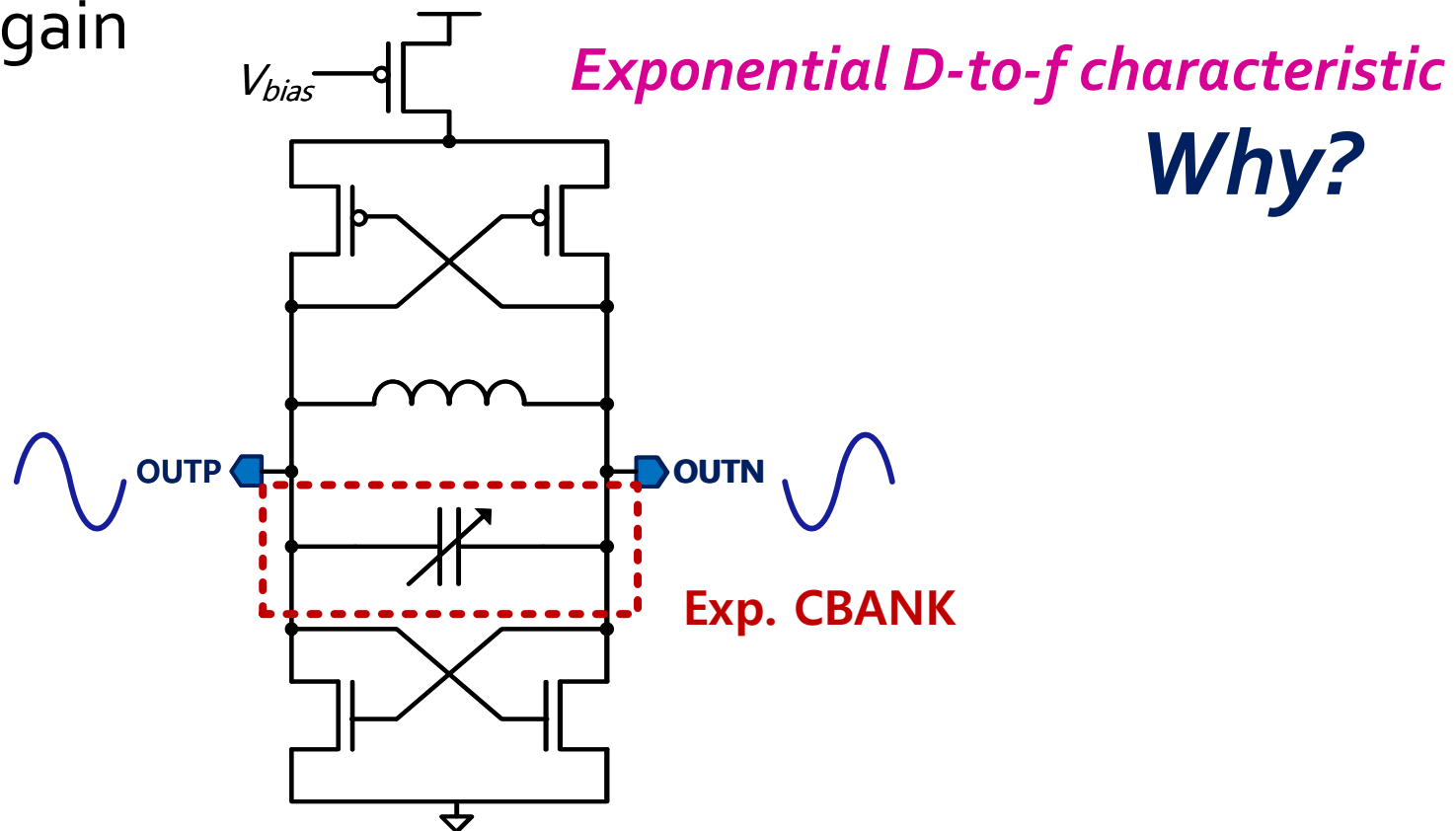
LC Digitally Controlled Oscillator

- Generates a pair of differential clocks whose frequency is corresponding to the input digital code



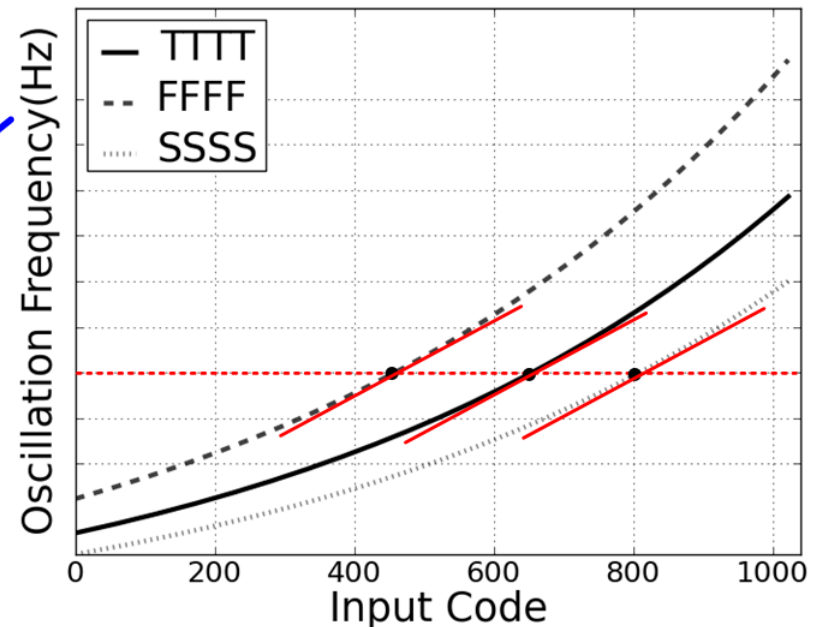
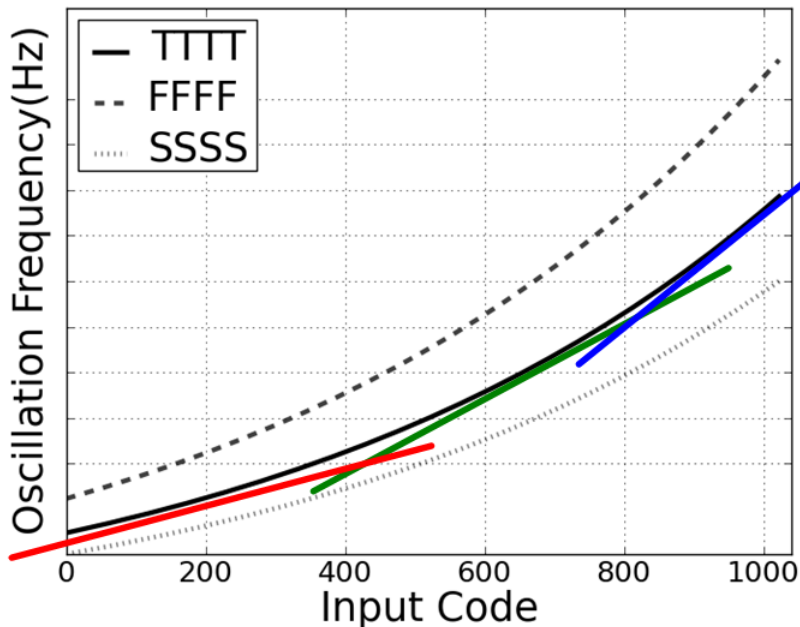
LCDCO Structure

- ▶ LC tank and negative g_m cell constitute resonator
- ▶ Exponentially tuned-capacitor for the constant relative gain



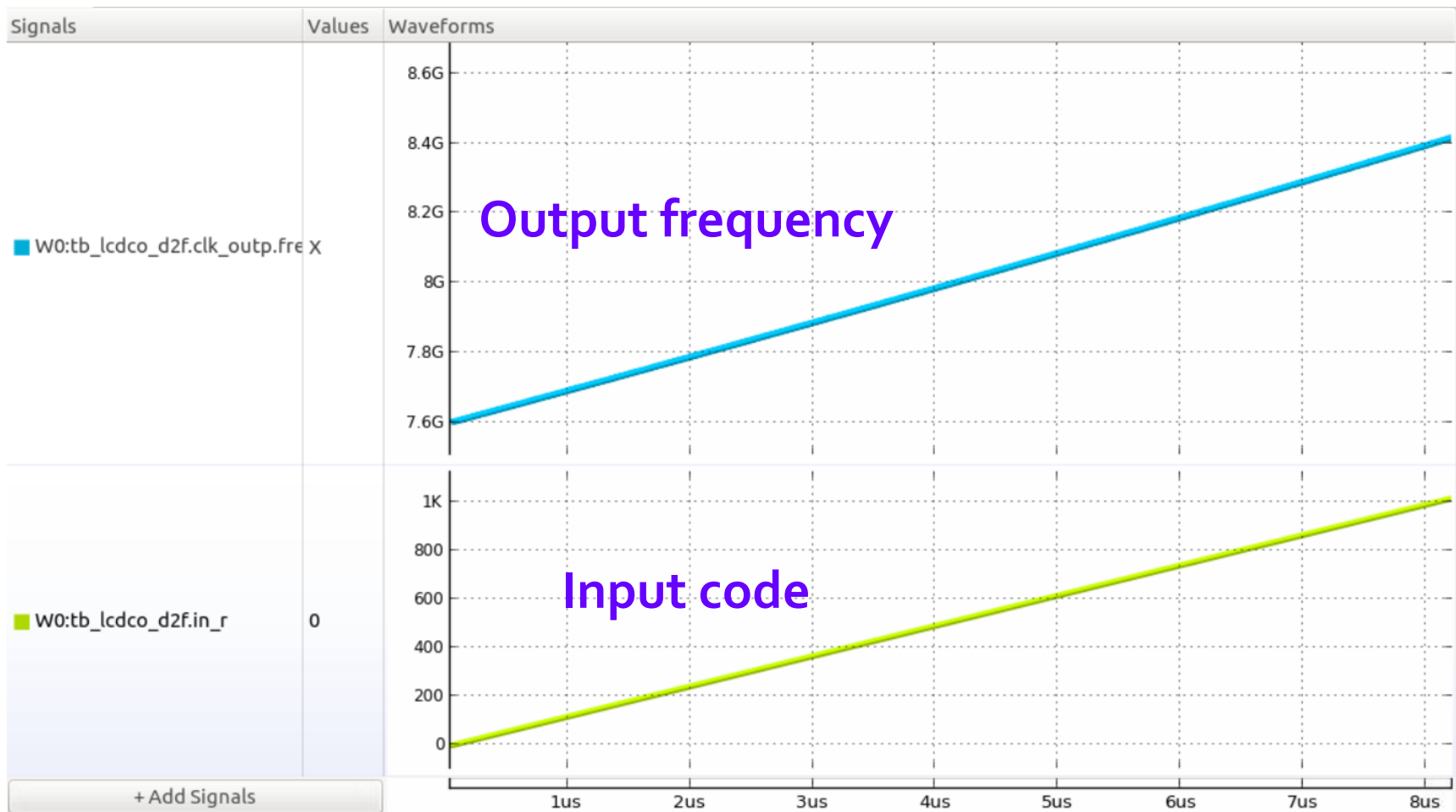
Adaptive-Bandwidth Digital PLLs

- ▶ Equivalent adaptive-BW can be achieved by realizing an ***exponential D - f curve*** for the DCO
 - ▶ K_{DCO} scaling with the frequency (left)
 - ▶ K_{DCO} constant over PVT at a given frequency (right)



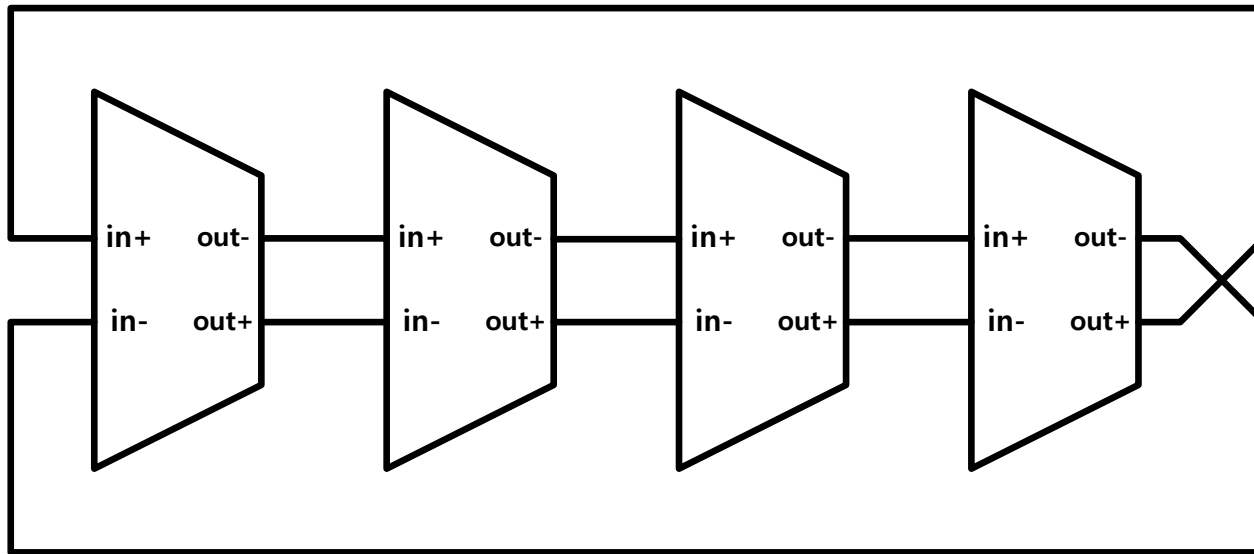
Exercise: LCDCO

► cell_dpll/lcdco/tb_lcdco



Ring Digitally Controlled Oscillator

- ▶ generates 8 uniformly-spaced multiphase clocks with an exponential D-to-f characteristic
- ▶ Comparing to LCDCO, it consumes much smaller power and chip area, but having poor noise performance



Frequency Dividers

▶ Prescaler (PRESCALER)

- ▶ Generates frequency-divided clock output whose frequency is slow enough to operate the other divider cells
- ▶ Use it for LCDCO whose output frequency is over 7 GHz

▶ General Divider (DIVIDER)

- ▶ Generates frequency-divided clock output with controllable integer dividing factor (**2, 3, 4, ..., 9**)

▶ Multiphase output divider (DIV8_MPHASE)

- ▶ Generates 8 uniformly-spaced multiphase clock output
- ▶ Dividing factor = 8

Testbench for DIVIDER

► cell_dpll/divider/tb_divider

```
clk_gen #(.freq(1.0e9) clk_ino(clk_inp);  
clk_gen #(.freq(1.0e9) clk_in1(clk_inn);
```

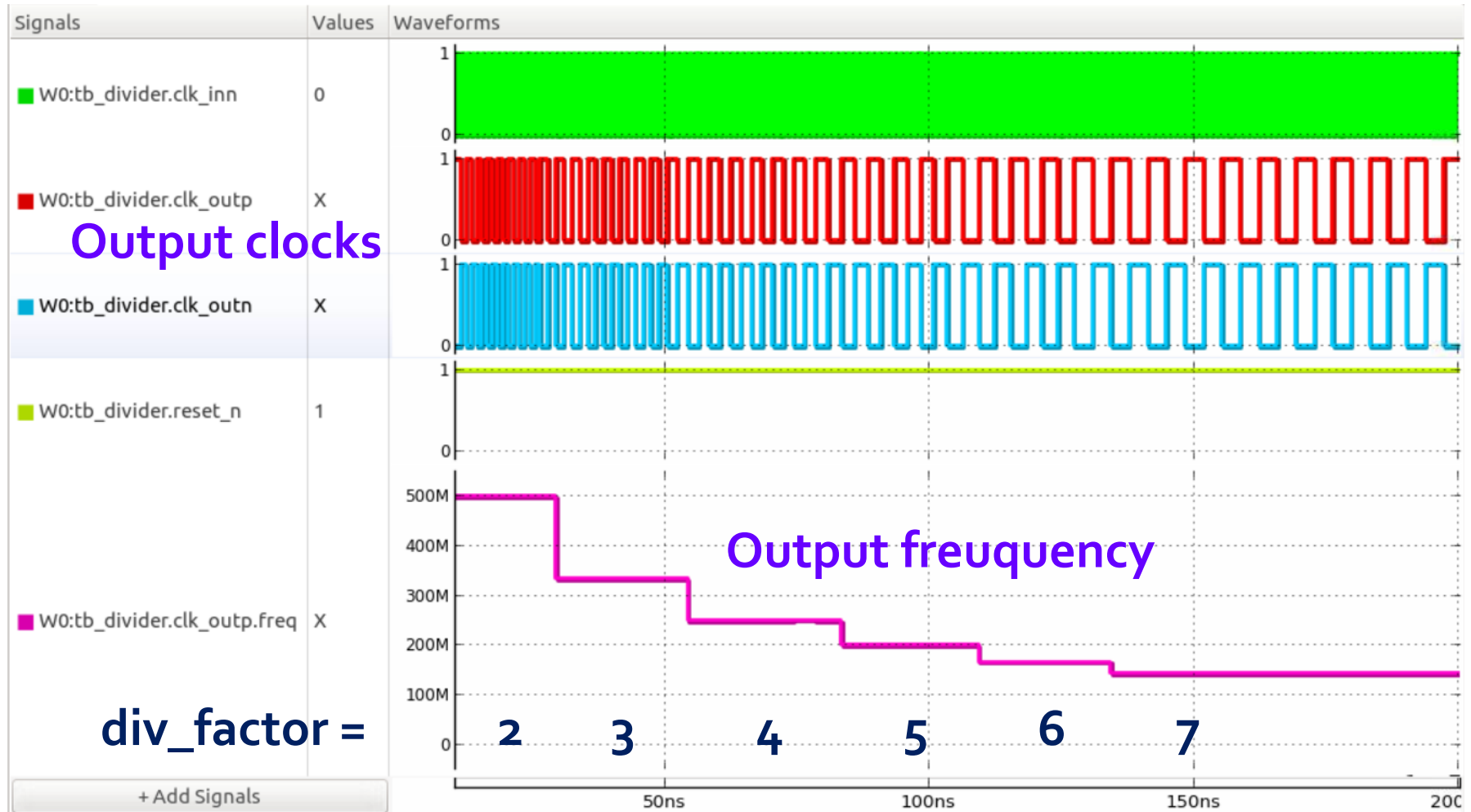
```
initial begin
```

```
    div_sel = 3'bo;           // div_factor = 2  
    #(25 ns) div_sel = 3'b001; // div_factor = 3  
    #(25 ns) div_sel = 3'b010; // div_factor = 4  
    #(25 ns) div_sel = 3'b011; // div_factor = 5  
    #(25 ns) div_sel = 3'b100; // div_factor = 6  
    #(25 ns) div_sel = 3'b101; // div_factor = 7
```

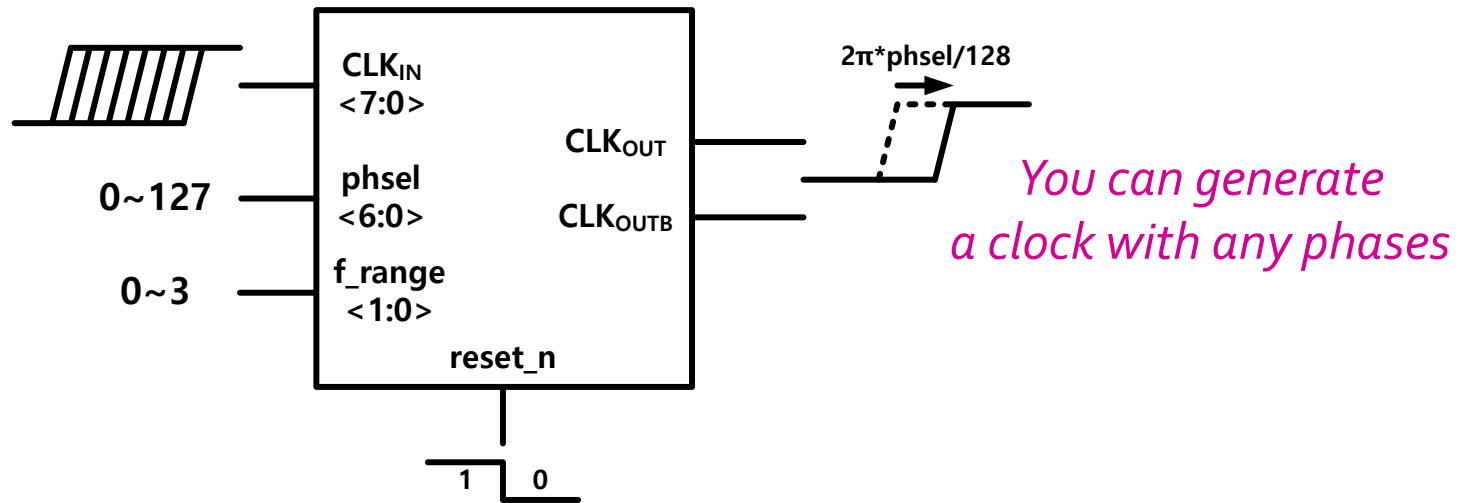
```
end
```

Exercise: DIVIDER

► cell_dpll/divider/tb_divider



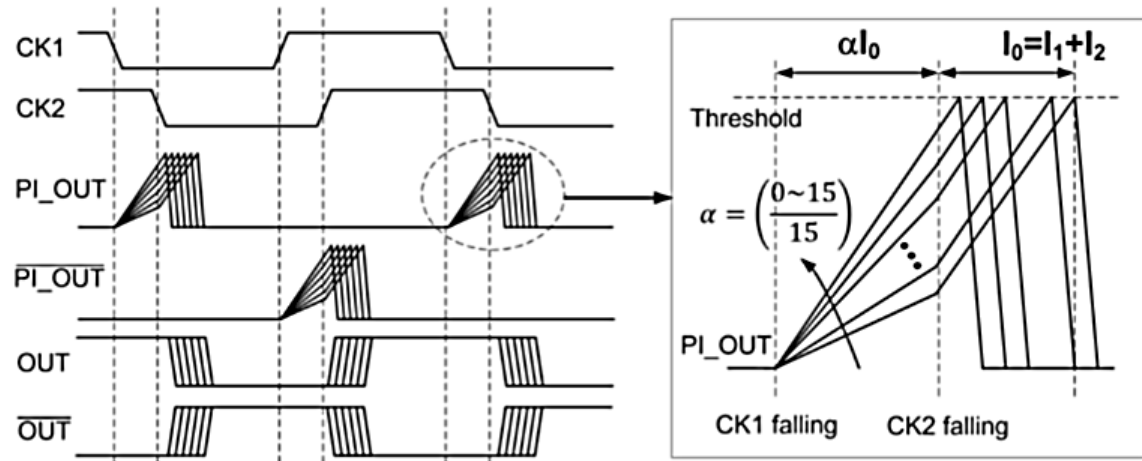
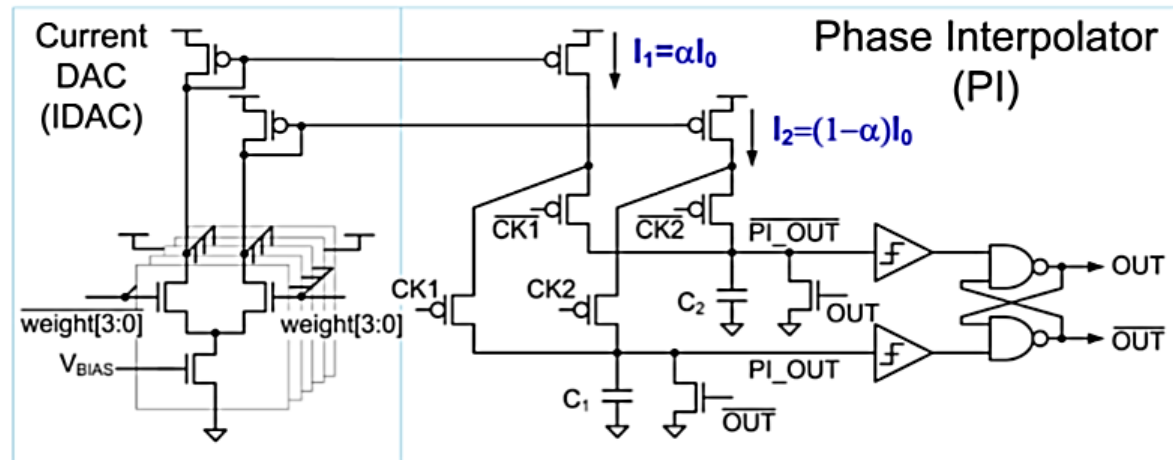
Phase Rotator



- ▶ 'phsel' pin enables 128 steps phase control
- ▶ Interpolates adjacent two clocks from 8 multiphase input clocks (phase interpolator)
- ▶ You can synthesize a clock with the phase you want

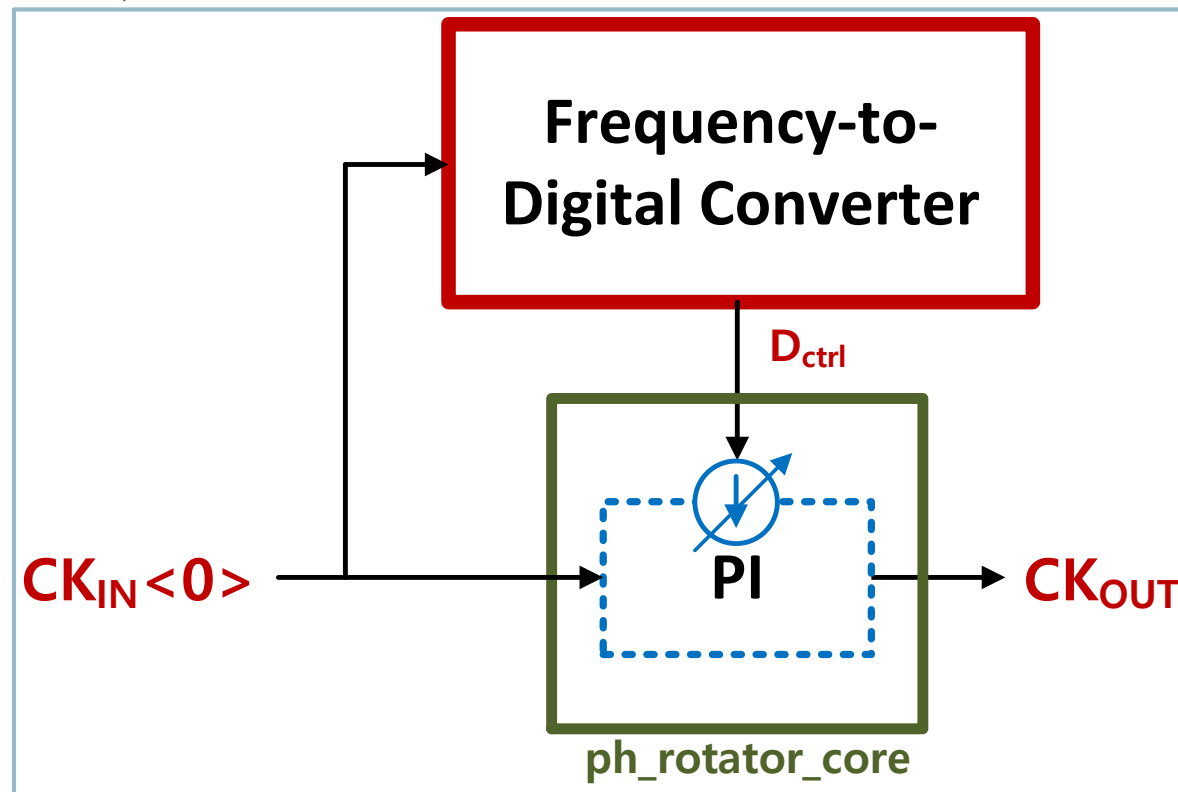
PH_ROTATOR Core Architecture

- ▶ The PI consists of two complementary parts, each interpolates the rising or falling edges of the two input clocks separately
- ▶ The interpolation weight is adjusted by a current DAC steering a fixed current I_0 into I_1 and I_2



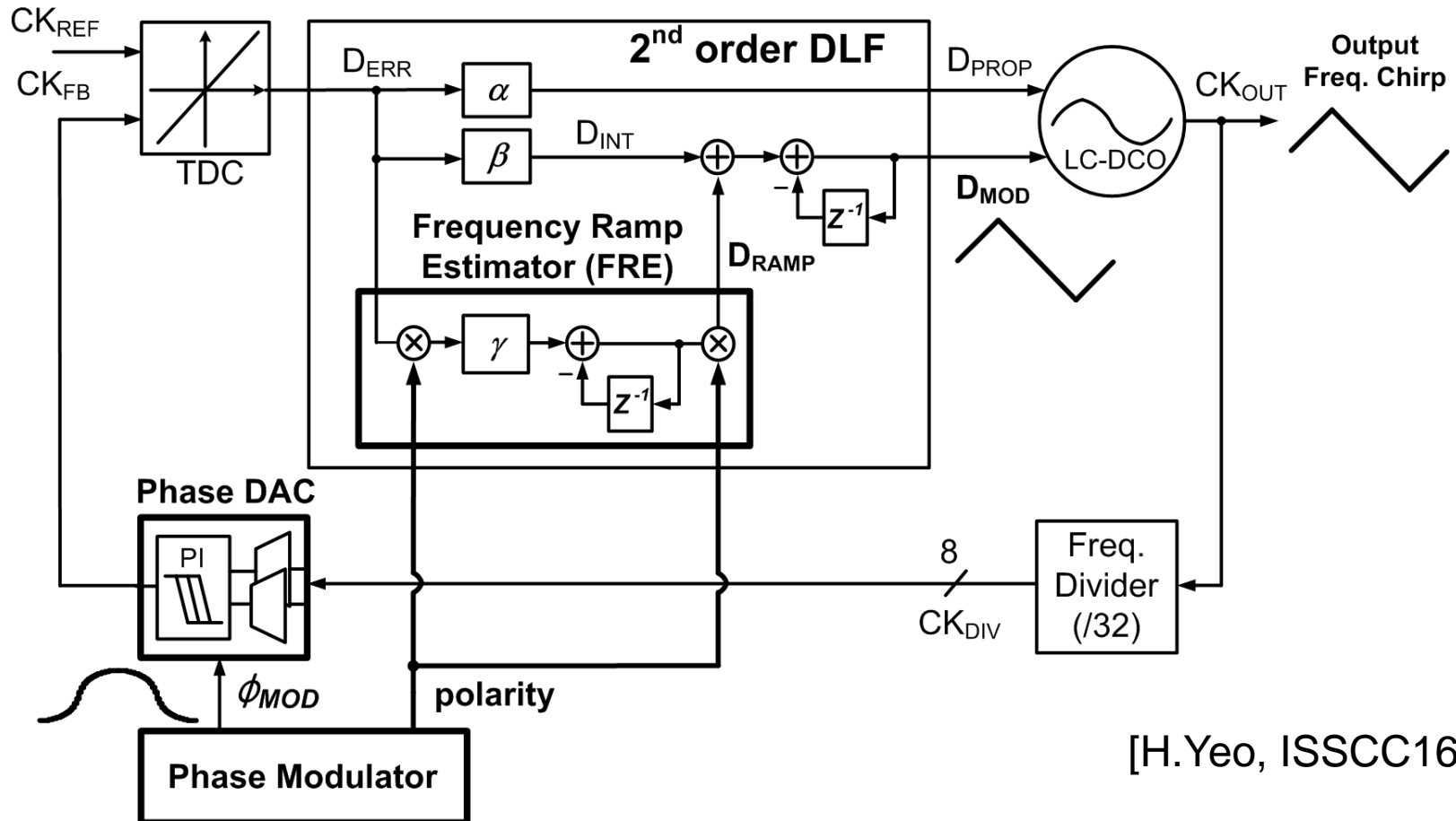
PH_ROTATOR Operation

- ▶ It also has a **frequency adaptation circuit**
 - ▶ Need sufficient time to acquire frequency (similar to GRO_TDC)



PH_ROTATOR Application Example

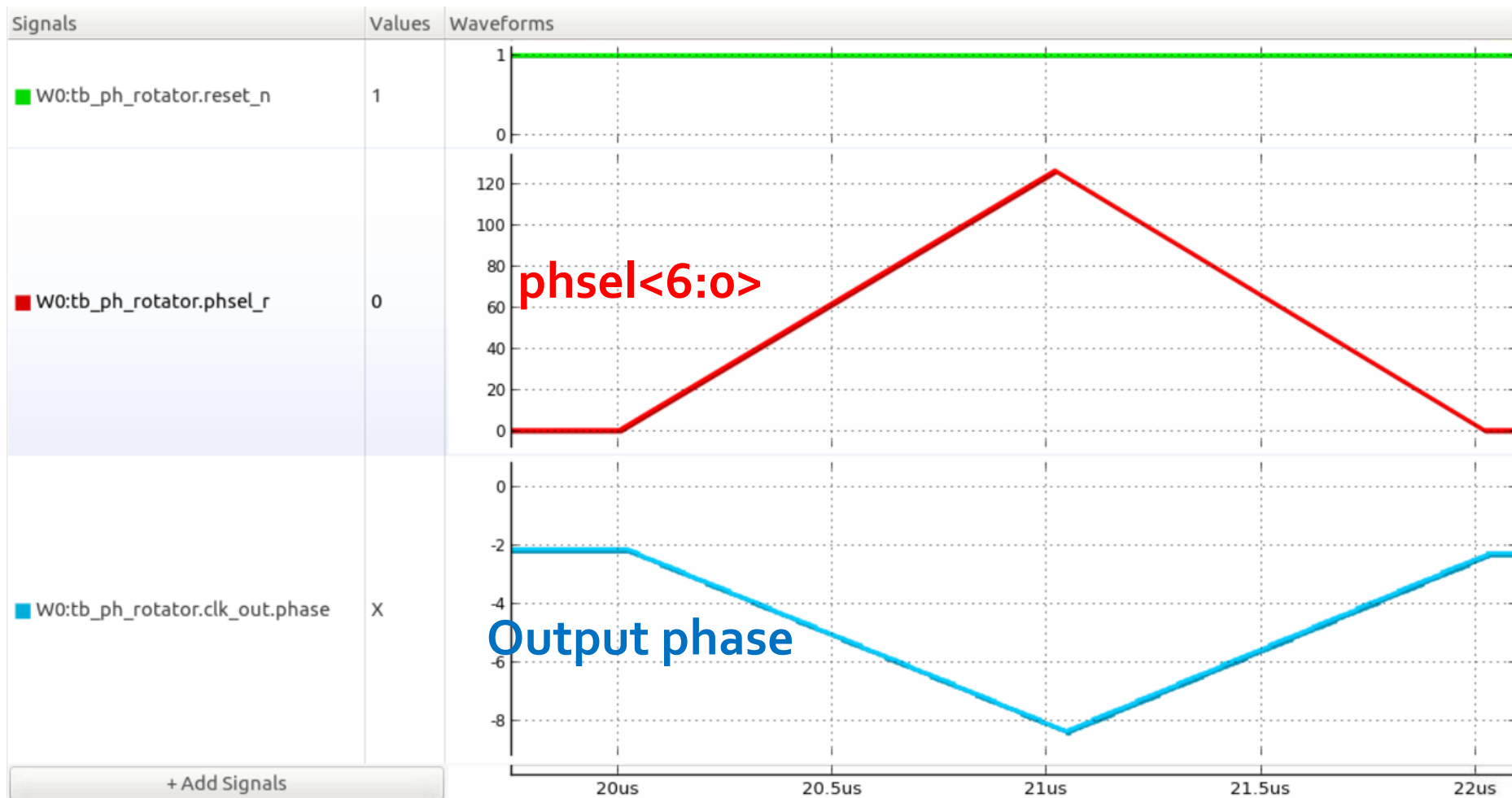
► Phase modulation



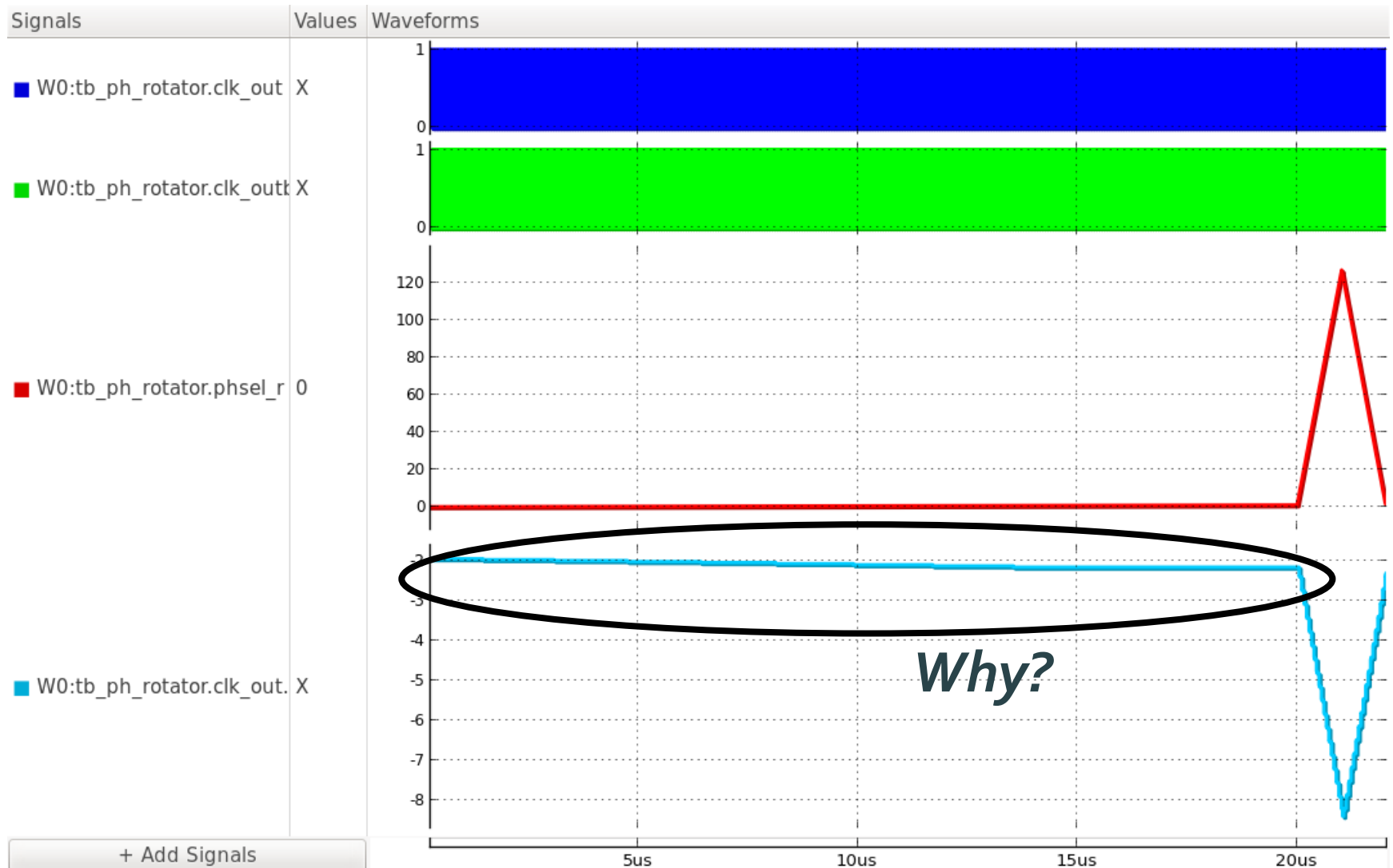
[H.Yeo, ISSCC16]

Exercise: PH_ROTATOR

► cell_dpll/ph_rotator/tb_ph_rotator



Exercise: PH_ROTATOR



Appendix

(Documentation + Cell Library Specification Sheet)

1. BBPD (functionality & pins)

► Functionality

- A bang-bang phase detector **compares phases of two input clocks** and expresses the result with bang-bang up and down output signals.

► Pins

- Input
 - 'clk_ref' : reference clock input
 - 'clk_fb' : feedback clock input
- Output
 - 'up' : 1-bit UP signal (0/1)
 - 'dn' : 1-bit DOWN signal (0/1)

2. BBPFD (functionality & pins)

► Functionality

- A bang-bang phase frequency detector compares two input clocks estimating that **which clock is faster at every positive edge of the feedback clock** and expresses the result with up and down bang-bang output signals. Additionally, it expresses how fast or slow it is with up and down pulse width signals.

► Pins

- Input
 - 'clk_ref' : reference clock input
 - 'clk_fb' : feedback clock input
- Output
 - 'up' : 1-bit UP signal (0/1)
 - 'dn' : 1-bit DOWN signal (0/1)
 - 'prop_up' : pulse width UP signal
 - 'prop_dn' : pulse width DN signal

3. GRO_TDC (functionality & pins)

► Functionality

- A gated relaxation oscillator typed time-to-digital converter **measures phase and frequency difference between two input clocks within every time interval of the feedback clock period** and expresses the timing error with digital code output. This cell guarantees the **constant TDC gain**. Additionally, it expresses how fast or slow it is with up and down pulse width signals.

3. GRO_TDC (functionality & pins)

► Pins

► Input

- 'reset_n' : reset (active low)
- 'clk_ref' : reference clock input
- 'clk_fb' : feedback clock input
- '[1:0] f_range' : frequency range selection code
(2-bits, 2 MHz – 12 MHz / 10 MHz - 50 MHz / 40 MHz - 150 MHz)

► Output

- '[5:0] 'up' : timing error output (up, 6-bits, binary)
- '[5:0] 'dn' : timing error output (down, 6-bits, binary)
- 'prop_up' : pulse width UP signal
- 'prop_dn' : pulse width DN signal

4. LCDCO (functionality & pins)

► Functionality

- An LC-DCO (LC digitally-controlled oscillator) generates a pair of differential clocks with an **exponential D-to-f characteristic**. Herein, the exponential relationship guarantees **constant relative-gain of DCO**.
- Note) It has 'clk_lf' pin for the input code retiming in order to prevent unwanted glitch. You need to connect a clock which is synchronized to the loop filter clock.

4. LCDCO (functionality & pins)

► Pins

► Input

- 'clk_lf' : clock synchronized to a DCO's digital driver(e.g. loop filter in PLL)
- '[2:0] dco_pgain' : proportional gain control code (3-bits, binary, 16 / 32 / 64 / 128 / 256 / 512 / 1024)
- 'prop_up' : proportional control input UP code (pulse width)
- 'prop_dn' : proportional control input DN code (pulse width)
- '[9:0] in' : DCO control input code (10-bits, binary code)
- '[2:0] in_dsm' : DCO control DSM input code (3-bits, binary signed code)
- 'reset_n' : reset (active low)

► Output

- 'clk_outp' : positive output clocks
- 'clk_outn' : negative output clocks

5. RING_DCO (functionality & pins)

► Functionality

- A ring DCO(ring-type digitally-controlled oscillator) generates 8 uniformly-spaced multiphase clocks with an **exponential D-to-f characteristic**. Herein, the exponential relationship guarantees **constant relative-gain of DCO**.
- Note) It has 'clk_lf' pin for the input code retiming in order to prevent unwanted glitch. You need to connect a clock which is synchronized to the loop filter clock.

5. RING_DCO (functionality & pins)

► Pins

► Input

- 'clk_lf' : clock synchronized to a DCO's digital driver(e.g. loop filter in PLL)
- '[2:0] dco_pgain' : proportional gain control code (3-bits, binary, 16 / 32 / 64 / 128 / 256 / 512 / 1024)
- 'prop_up' : proportional control input UP code (pulse width)
- 'prop_dn' : proportional control input DN code (pulse width)
- '[9:0] in' : DCO control input code (10-bits, binary code)
- '[2:0] in_dsm' : DCO control DSM input code (3-bits, binary signed code)
- 'reset_n' : reset (active low)

► Output

- '[7:0] clk_out' : 8 uniformly-spaced multiphase output clocks

6. PRESCALER (functionality & pins)

► Functionality

- A prescaler divides differential input clocks' frequency into 2, 4, or 8 and generates corresponding differential output clocks whose frequency is slow enough to operate DIVIDER or DIV8_MPHASE cell.

► Pins

- Input
 - 'clk_inp' : differential clock input (positive)
 - 'clk_inn' : differential clock input (negative)
 - 'reset_n' : reset (active low)
 - '[1:0] div_sel' : dividing factor selection code (2-bits, binary, 2 / 4 / 8)
- Output
 - 'clk_outp' : clock output (positive)
 - 'clk_outn' : clock output (negative)

7. DIVIDER (functionality & pins)

► Functionality

- A divider divides the input clock frequency with a selected dividing factor and generates a corresponding output clock with 50% duty. For the dividing factor, integers from 2 to 9 are available.

► Pins

- Input
 - 'clk_in' : input clock
 - 'reset_n' : reset (active low)
 - '[2:0] div_sel' : dividing factor selection code (3-bits, binary, 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9)
- Output
 - 'clk_out' : output clock

8. DIV8_MPHASE (functionality & pins)

► Functionality

- A multiphase divider divides the differential input clocks frequency into 8 and generates corresponding 8 uniformly-spaced multiphase clock outputs.

► Pins

- Input
 - 'clk_inp' : differential clock input (positive)
 - 'clk_inn' : differential clock input (negative)
 - 'reset_n' : reset (active low)
- Output
 - '[7:0] clk_out' : multiphase clock outputs

9. PH_ROTATOR (functionality & pins)

► Functionality

- A phase rotator interpolates the 8 uniformly-spaced multiphase input clocks' phases and generates a pair of differential clocks with a digitally-controlled phase which is controlled by control inputs (the phase selection code, phsel).

9. PH_ROTATOR (functionality & pins)

► Pins

► Input

- '[7:0] clk_in' : 8 uniformly-spaced multiphase clocks
- '[6:0] phsel' : phase selection code for the interpolation (7-bits, binary, 128 phase steps)
- '[1:0] f_range' : frequency range selection code (2-bits, 10 MHz - 50 MHz / 40 MHz - 150 MHz / 120 MHz - 560 MHz / 500 MHz - 1.8 GHz)
- 'reset_n' : reset (active low)

► Output

- 'clk_out' : clock output (positive)
- 'clk_outb' : clock output (negative)

Specification Sheet (1)

Analog Cell	Feature & Specification
BBPD	Operating frequency range < 300 MHz
BBPFD	Operating frequency range < 300 MHz proportional path pulse width output
GRO-TDC	$K_{TDC} = 64$ bit/UI Operating frequency range = 2 - 150 MHz (2-12 MHz, 10-50 MHz, 40-150 MHz) Proportional path pulse width output

Specification Sheet (2)

Analog Cell	Feature & Specification
LCDCO	Differential clock output K_{DCO} (relative gain) = 0.00010 Phase noise = -112 dBc/Hz (@ 1 MHz offset) Frequency tuning range = 7.6 - 8.4 GHz
Ring DCO	8 uniformly-spaced multiphase clock output K_{DCO} (relative gain) = 0.00056 Phase noise = -85 dBc/Hz (@ 1 MHz offset) Frequency tuning range = 1 – 1.8 GHz

Specification Sheet (3)

Analog Cell	Feature & Specification
PRESCALER	Dividing factor = 2, 4, 8 Operating frequency range < 10 GHz
DIVIDER	Dividing factor = 2, 3, 4, ..., 9 Operating frequency range < 2 GHz
DIV8_MPHASE	8 uniformly-spaced multiphase output clock Operating frequency range < 2 GHz
PH_ROTATOR	Resolution = 2.8125 °/bit (phase selection code : 128 steps) Operating frequency range = 10 MHz – 1.8 GHz (10-50 MHz, 40-150 MHz, 120-560 MHz, 0.5-1.8 GHz)