Referenceless Digital Clock Data Recovery with counter-based Kp/Ki Control

EECS, SNU
2017-19047
Sungyoung Lee

Brief Refresher

- Referenceless Digital CDR

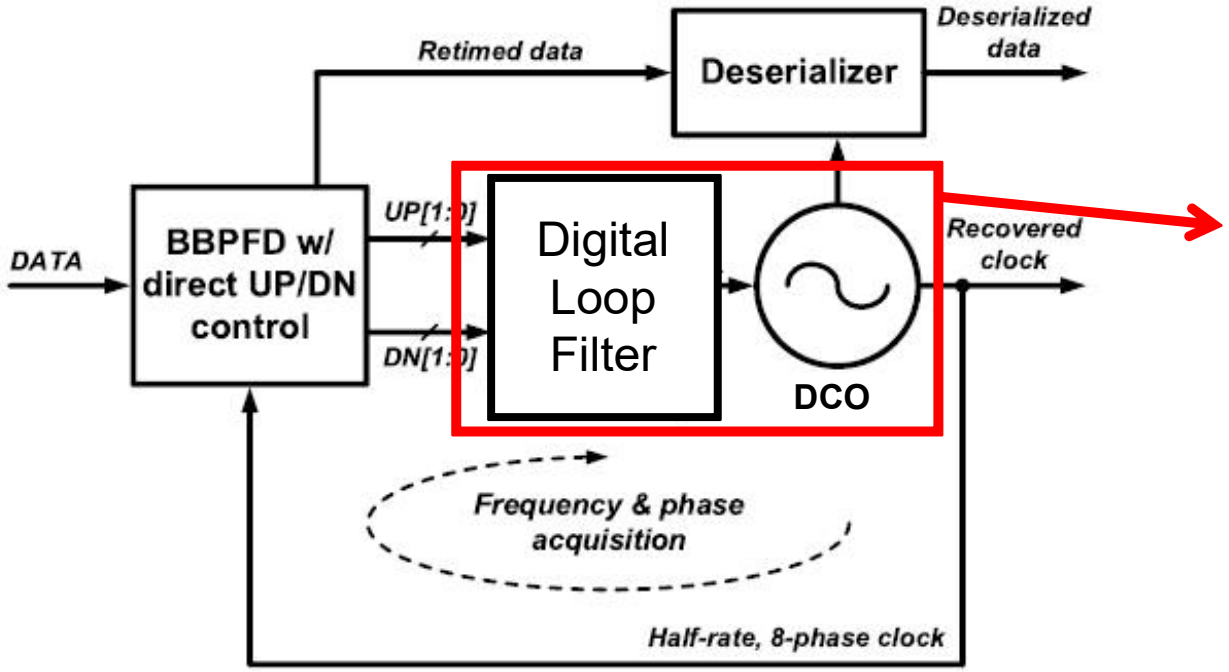


Fig. 10. Proposed referenceless CDR block diagram.

- Dithering is unavoidable

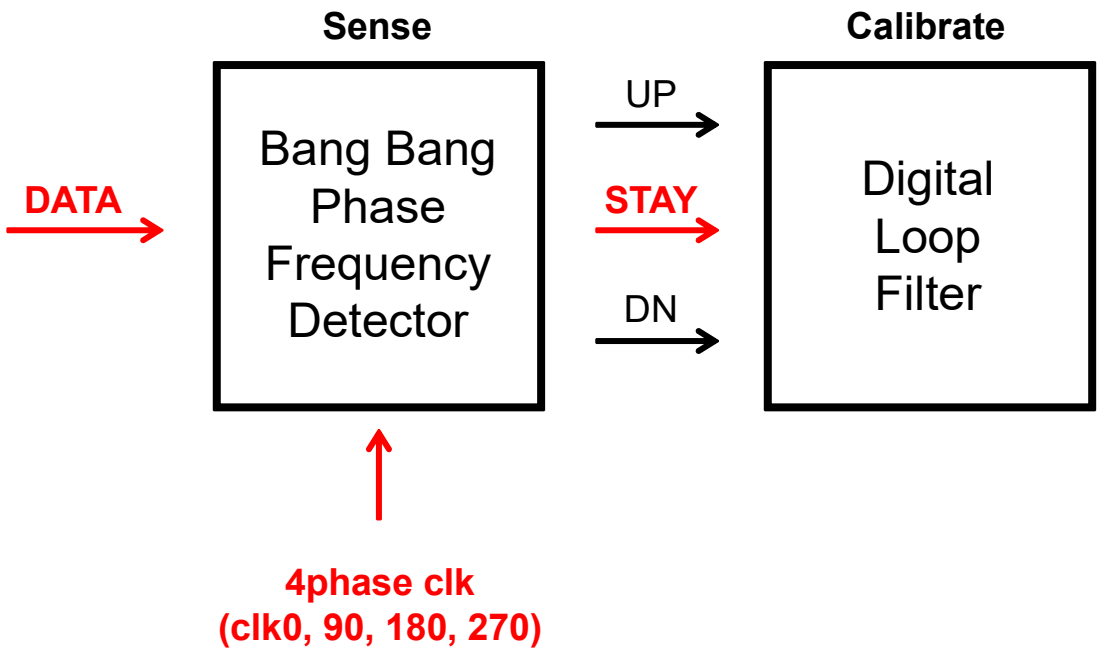
KEY GOAL of CALIBRATION : minimizing the recovered clock's phase dithering step

Kwanseo Park, Woorham Bae, Jinhyung Lee, Jeongho Hwang, Deog-Kyoon Jeong

[A 6.7–11.2 Gb/s, 2.25 pJ/bit, Single-Loop Referenceless CDR With Multi-Phase, Oversampling PFD in 65-nm CMOS]

Brief Refresher

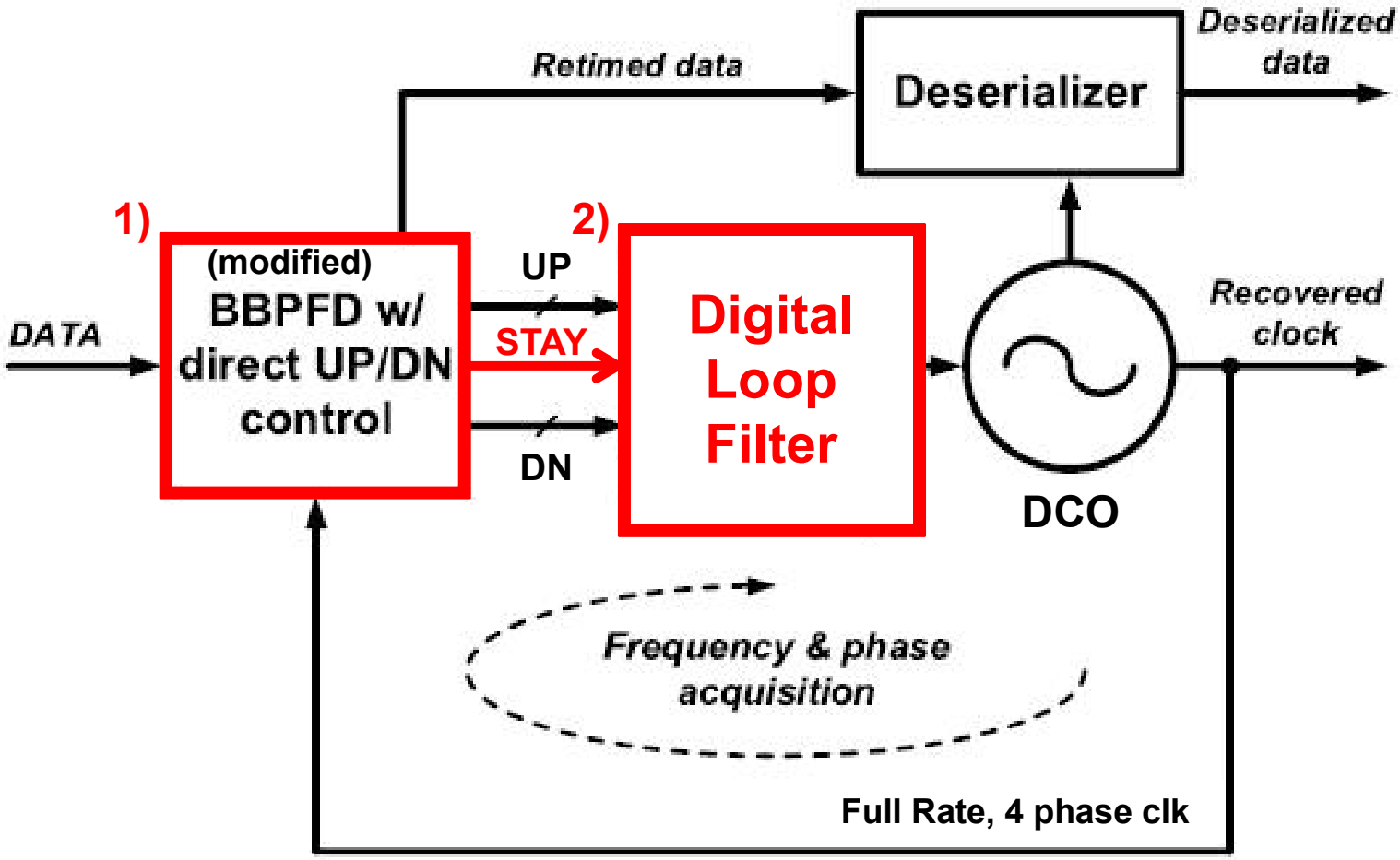
- Sensing & Calibrating DCO Dither (Proposal)



Architecture

Overall Architecture

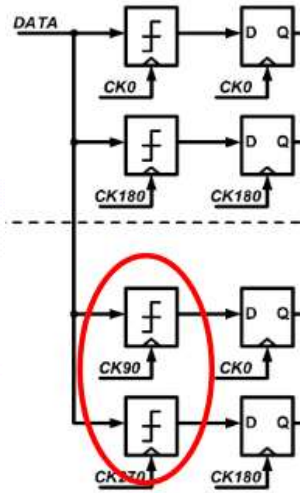
- Proposing Referenceless Oversampling Digital CDR

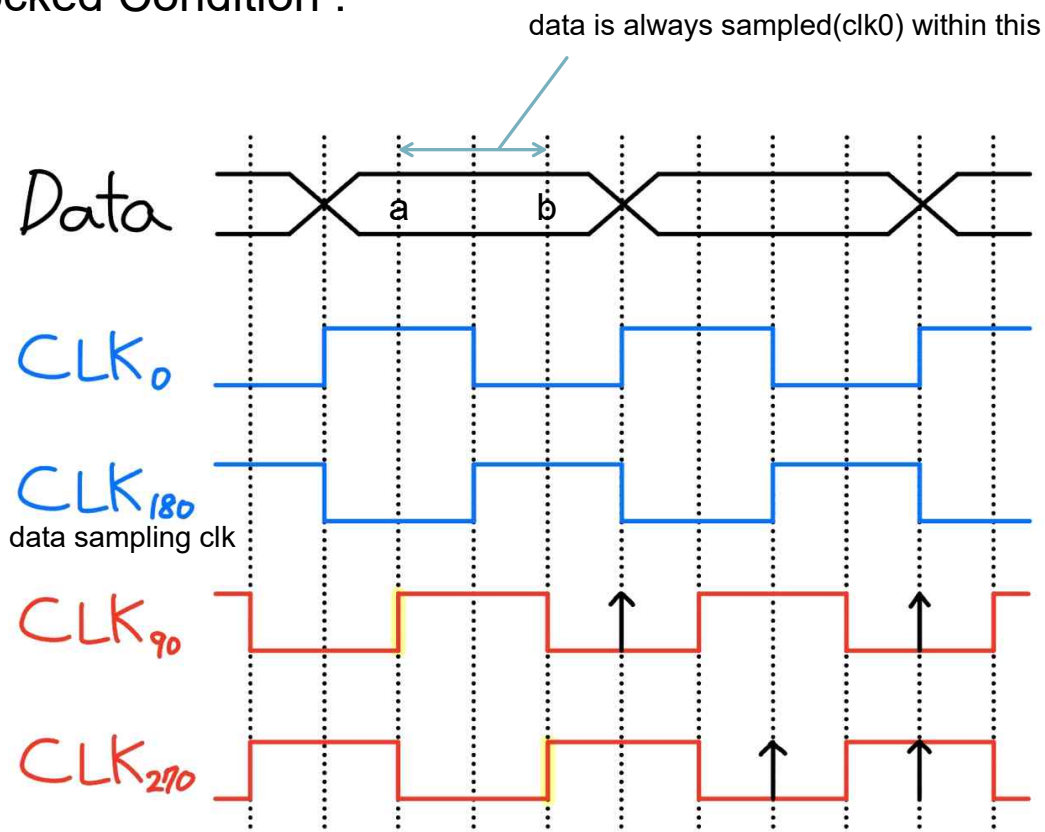


1) Bang Bang Phase Frequency Detector

- Sensing STAY signal

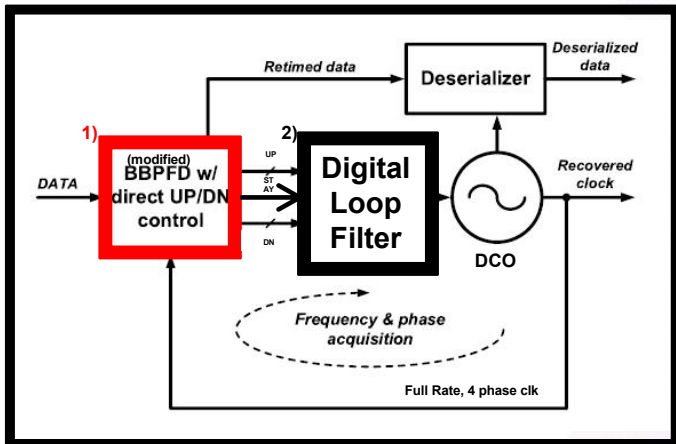
At Successfully Locked Condition :





a : data value sampled at CLK90
b : data value sampled at CLK270

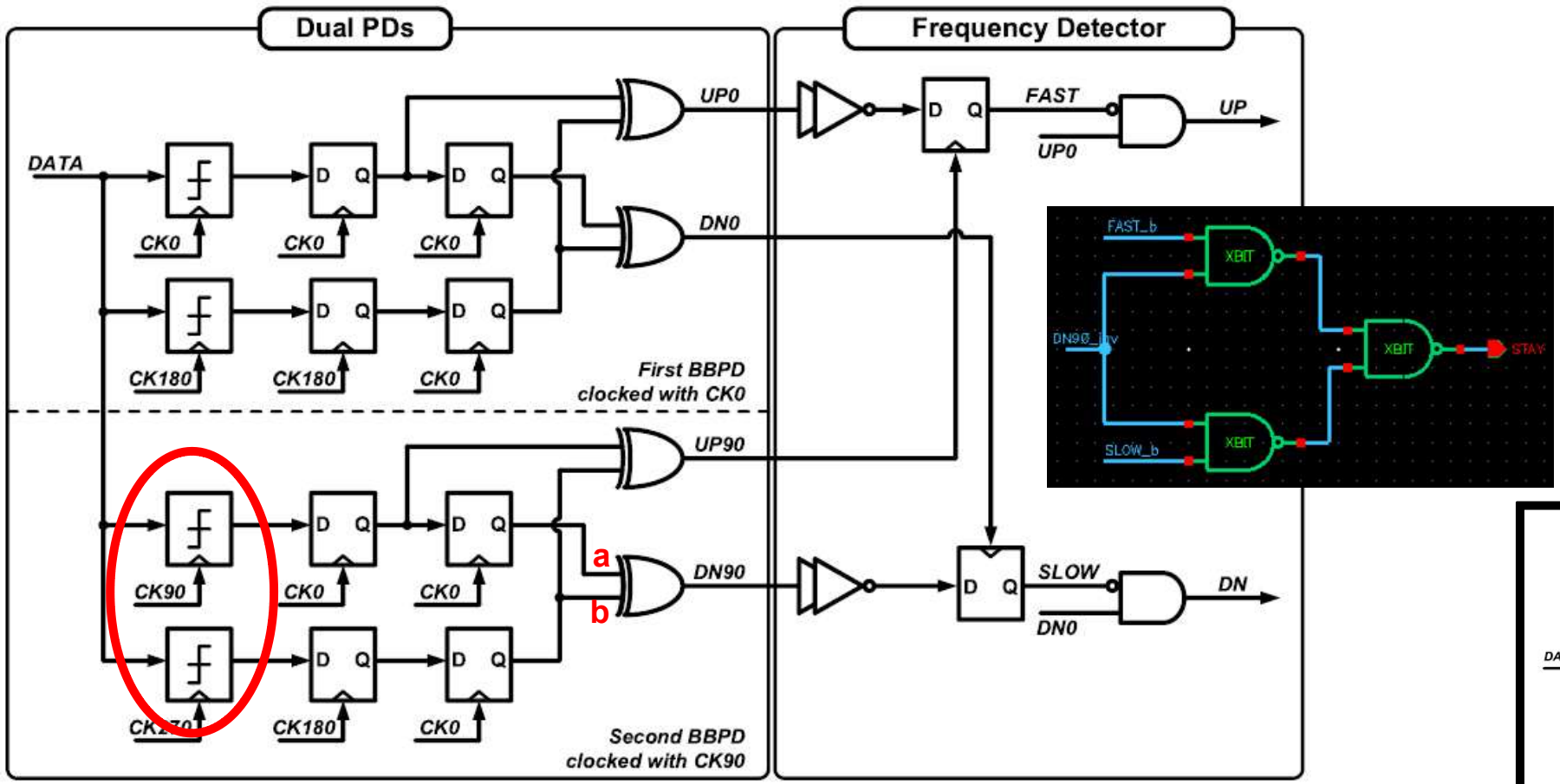
a == b



1) Bang Bang Phase Frequency Detector

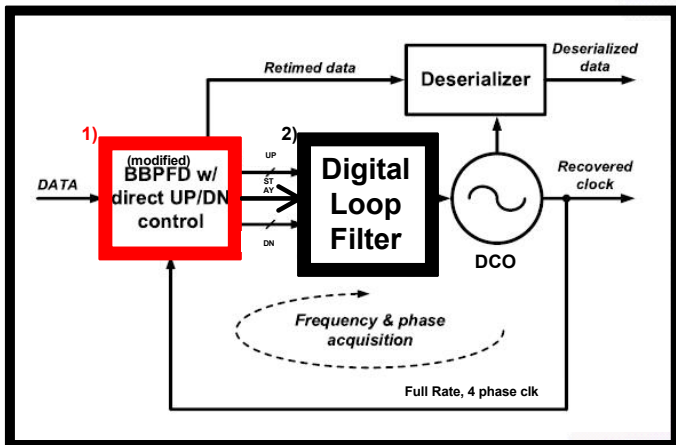
- Sensing STAY signal

Preventing Harmonic Lock



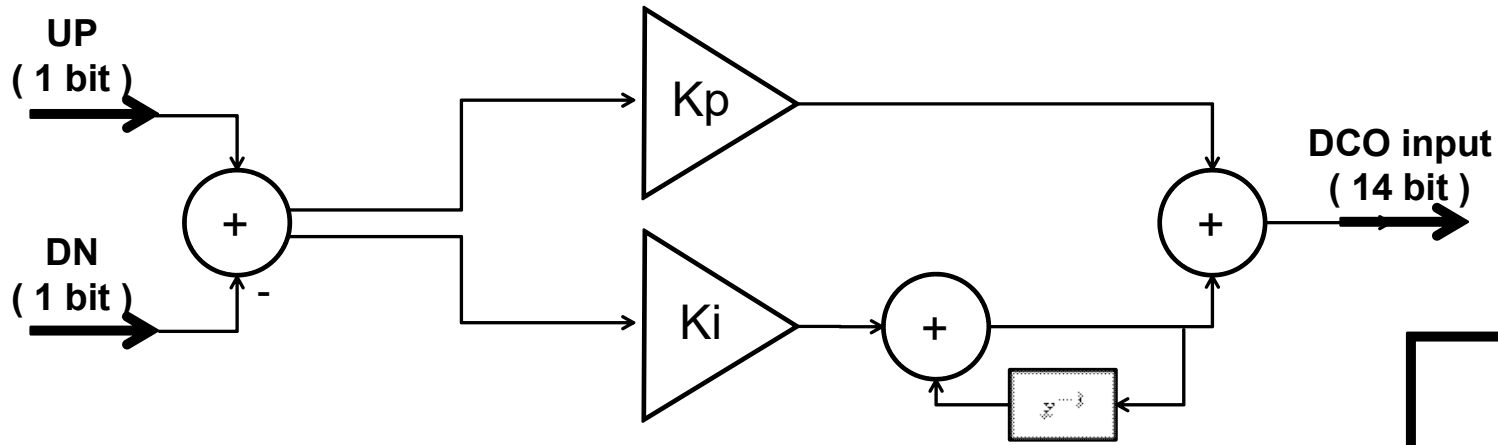
(a == b)
&
(FAST & SLOW == 0) :

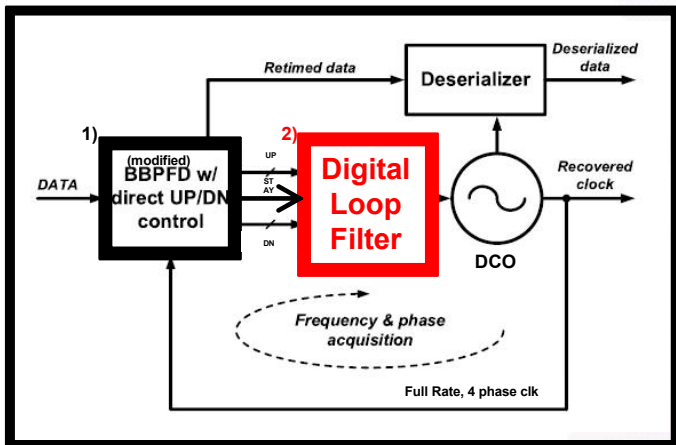
STAY = 1



2) Digital Loop Filter

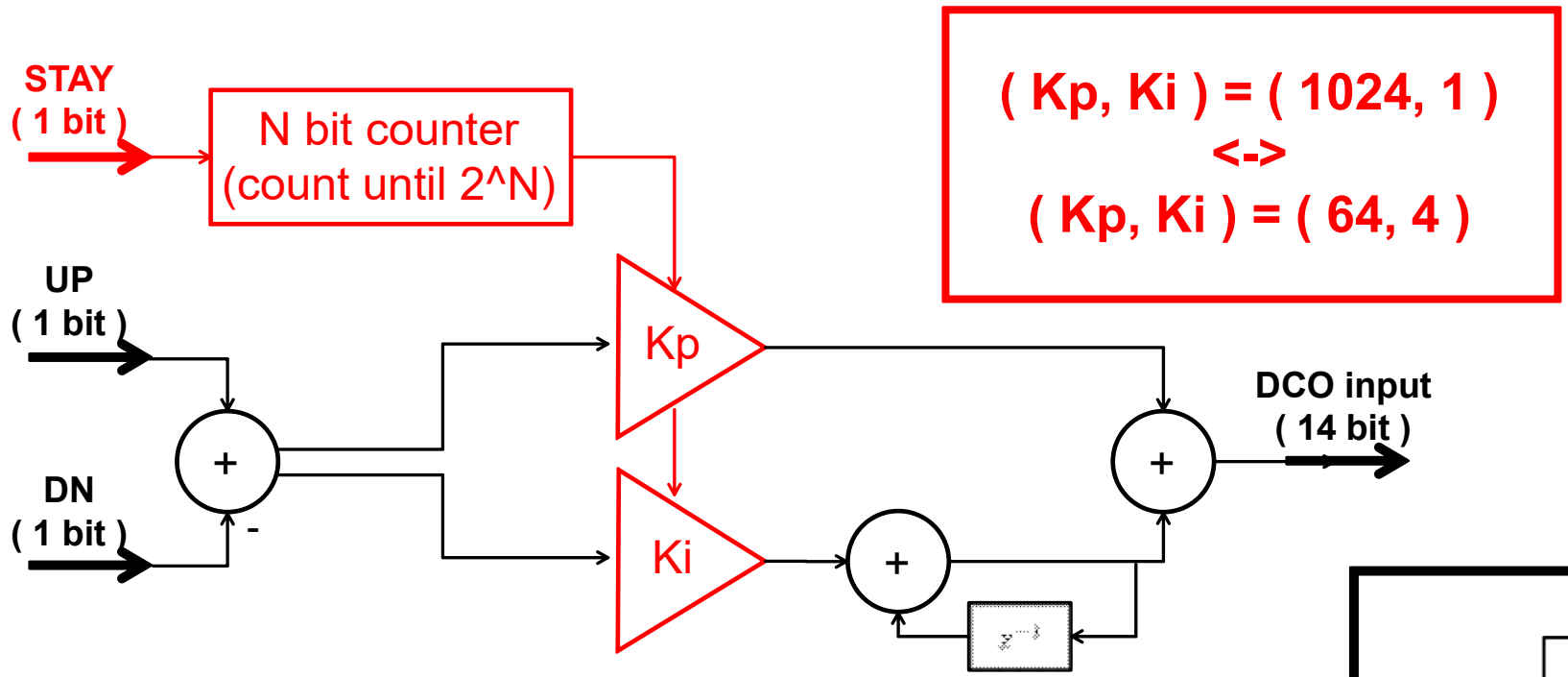
- Calibrating without STAY signal

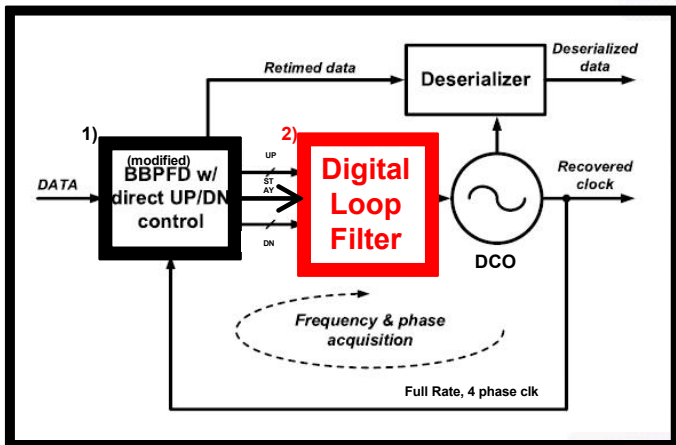




2) Digital Loop Filter

- Calibrating with STAY signal

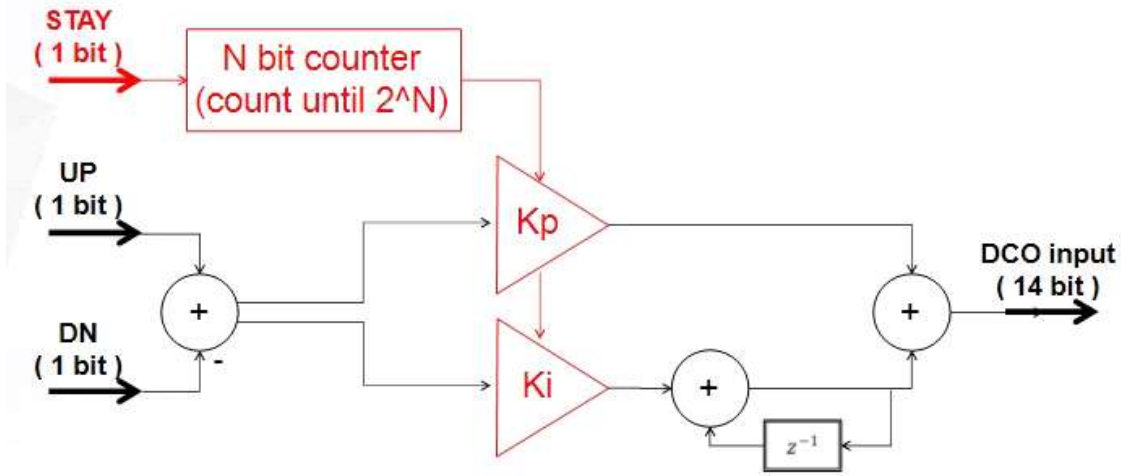


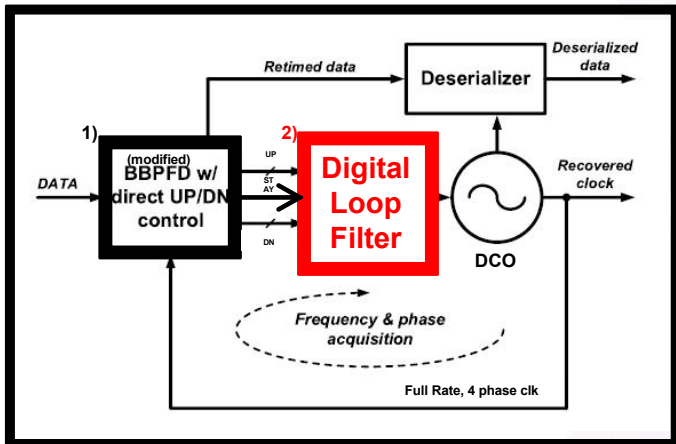


2) Digital Loop Filter

- Calibrating with STAY signal (XMODEL)

```
module digital_lf_z2 #(
    `parameter_integer(width_out,12),    // width of out
    `parameter_integer(Nd,4),            // loop filter delay
    `parameter_bit([width_out-1:0]init_out,12'b100000000000), // initial out
    `parameter_integer(Kp,1024),         // proportional gain
    `parameter_integer(Ki,1),            // integral gain
    `parameter_integer(Kp_locked,256),   // proportional gain when locked
    `parameter_integer(Ki_locked,4),     // integral gain when locked
    `parameter_integer(log_Ns,8)         // count stay signal for 2^Ns times
)(
    output reg [13:0] out,               // output signal
    input bit clk,                       // triggering clock
    input bit up,                        // up signal from pfd
    input bit dn,                        // down signal from pfd
    input bit stay                       // stay signal from pfd
);

    // variables
    reg [width_out-1:0] out_d [Nd-1:0];
    reg [width_out-1:0] acc;
    reg [width_out-1:0] out_p;
    reg lock;
    reg [log_Ns-1:0] stay_counter;

    int i;

    // initialize out
    initial begin
        for (i=0; i<Nd; i++) begin
            out_d[i] = init_out;
        end
        acc = init_out;
        lock = 1'b0;
        stay_counter = {log_Ns{1'b0}};
    end
```





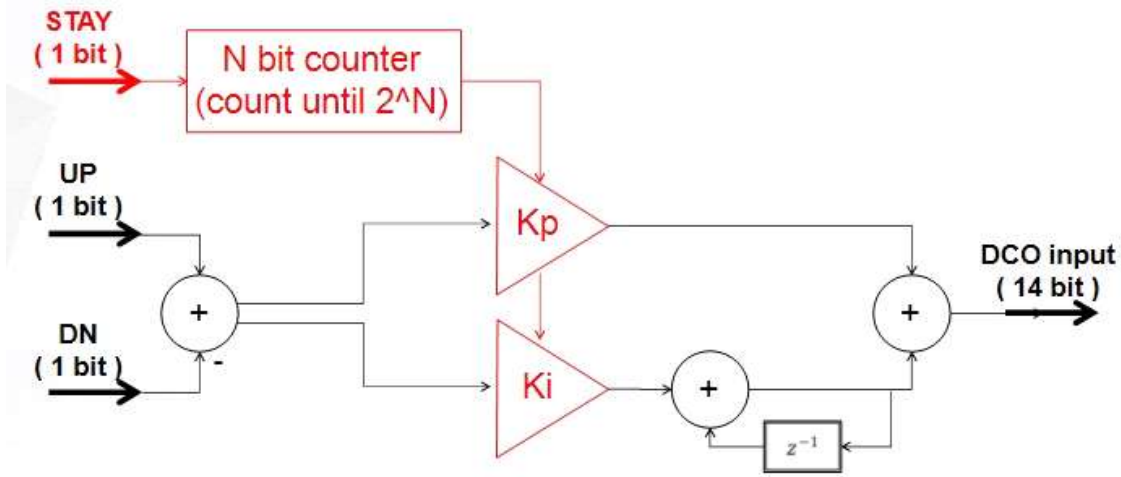
2) Digital Loop Filter

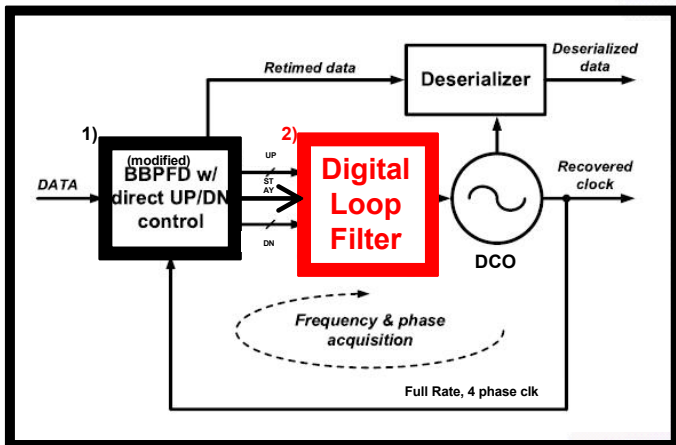
- Calibrating with STAY signal (XMODEL)

STAY = 1
(N consecutive times)

STAY = 0
(Once)

```
always @(posedge clk) begin

    if (~lock) begin
        acc = acc + Ki*(up - dn);  // accumulator for integral path
    end
    if (lock) begin
        acc = acc + Ki_locked*(up - dn);  // accumulator for integral path
    end

    // lock
    if (stay) begin
        if (stay_counter == {log_Ns{1'b1}}) begin
            lock = 1;
        end
        else begin
            stay_counter = stay_counter + 1;
        end
    end

    // unlock
    if (~stay) begin
        lock = 0;
        stay_counter = {log_Ns{1'b0}};
    end

end

always @(acc or up or dn) begin
    if (~lock) begin
        out_p = Kp*(up - dn) + acc;
    end
    if (lock) begin
        out_p = Kp_locked*(up - dn) + acc;
    end
end

    assign out = out_d[Nd-1];

    always @(posedge clk) begin
        for (i=Nd-1; i>0; i--) out_d[i] = out_d[i-1];
        out_d[0] = out_p;
    end

endmodule
```



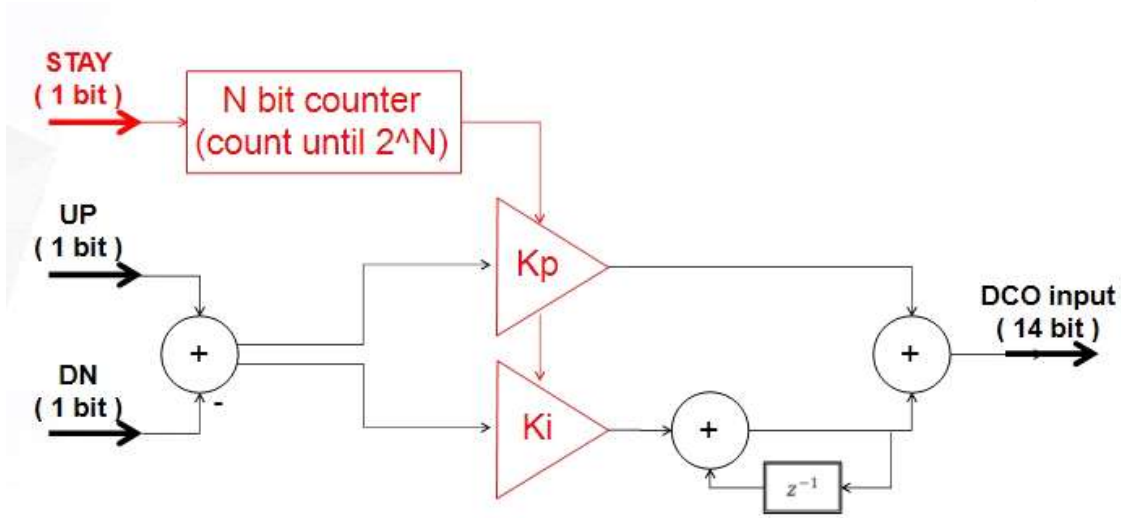


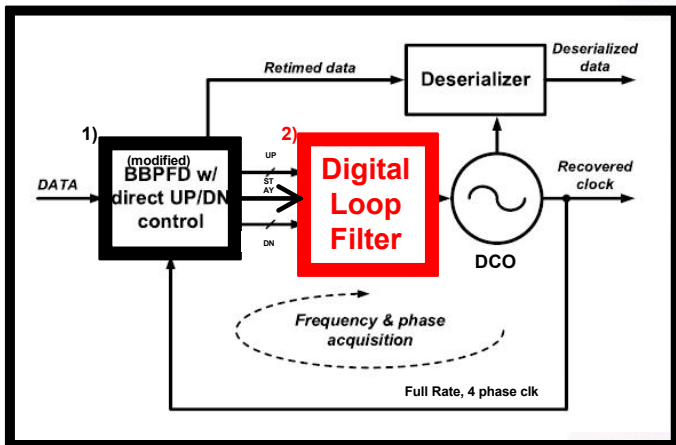
2) Digital Loop Filter

- Calibrating with STAY signal (XMODEL)

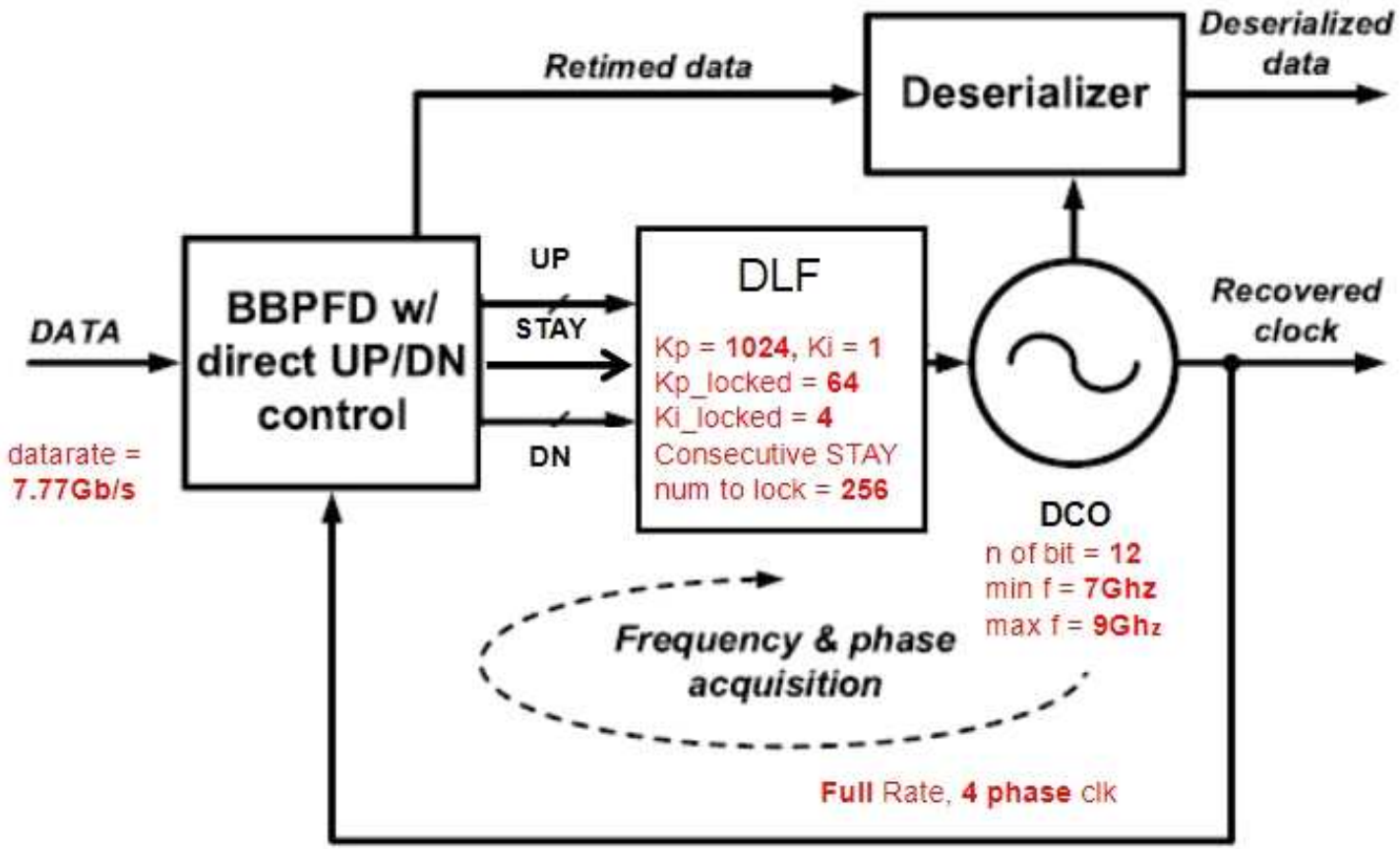
Change Kp, Ki by lock value

```
always @(posedge clk) begin

    if (~lock) begin
        acc = acc + Ki*(up - dn);  // accumulator for integral path
    end
    if (lock) begin
        acc = acc + Ki_locked*(up - dn);  // accumulator for integral path
    end

    // lock
    if (stay) begin
        if (stay_counter == {log_Ns{1'b1}}) begin
            lock = 1;
        end
        else begin
            stay_counter = stay_counter + 1;
        end
    end

    // unlock
    if (~stay) begin
        lock = 0;
        stay_counter = {log_Ns{1'b0}};
    end

end

always @(acc or up or dn) begin
    if (~lock) begin
        out_p = Kp*(up - dn) + acc;
    end
    if (lock) begin
        out_p = Kp_locked*(up - dn) + acc;
    end
end

    assign out = out_d[Nd-1];

    always @(posedge clk) begin
        for (i=Nd-1; i>0; i--) out_d[i] = out_d[i-1];
        out_d[0] = out_p;
    end

endmodule
```





Test Conditions



For Analog CDR in phase noise comparison

CP up/dn current = 200uA

Initial Control Voltage = 0.5V

Rs = 0.7k ohm, Cs = 14.4pF, Cp = 3.6pF

VCO range : 7Ghz ~ 9Ghz

For Digital CDR in functionality and performance comparison

1) Kp = 1024, Ki = 1

2) Kp = 64, Ki = 4

(all the other conditions are same)

Functionality Test

data rate : 7.77Gb/s

simulation time : 1us

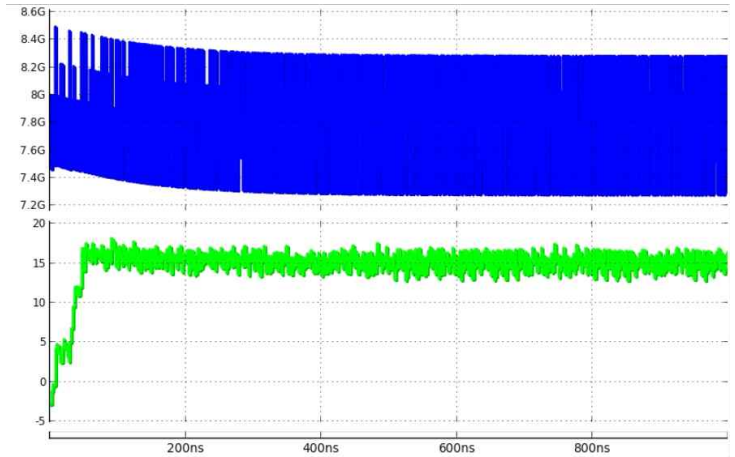
timescale : 100fs/100fs

Frequency/Phase of the Recovered Clock

- without STAY signal (Kp, Ki) = (1024, 1)

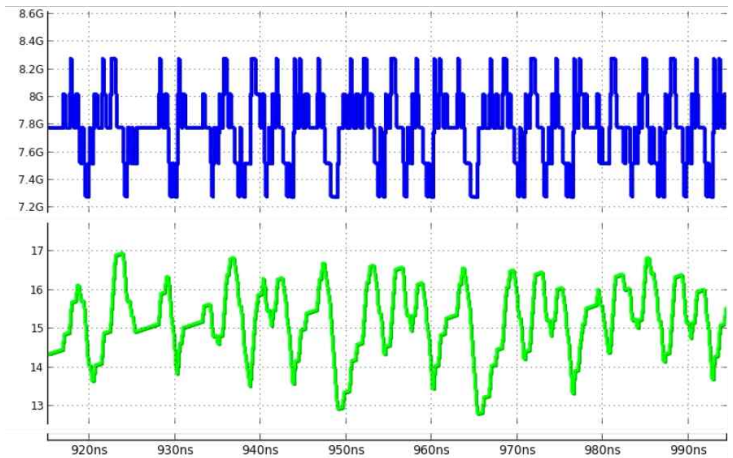
frequency

phase



frequency

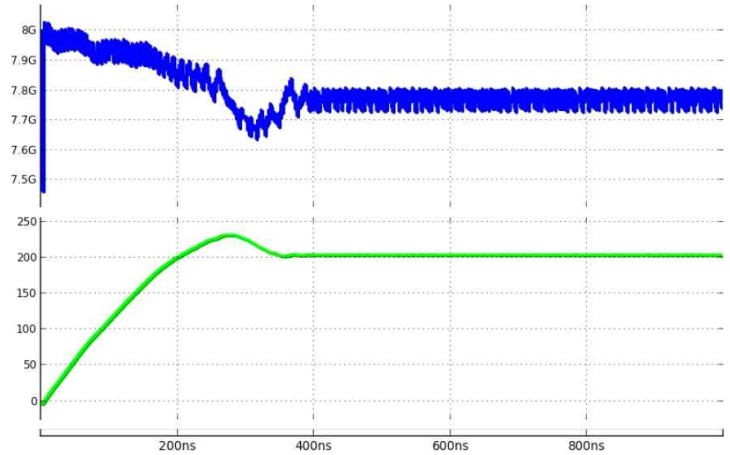
phase



- without STAY signal (Kp, Ki) = (64, 4)

frequency

phase



frequency

phase

data rate : 7.77Gb/s

simulation time : 1us

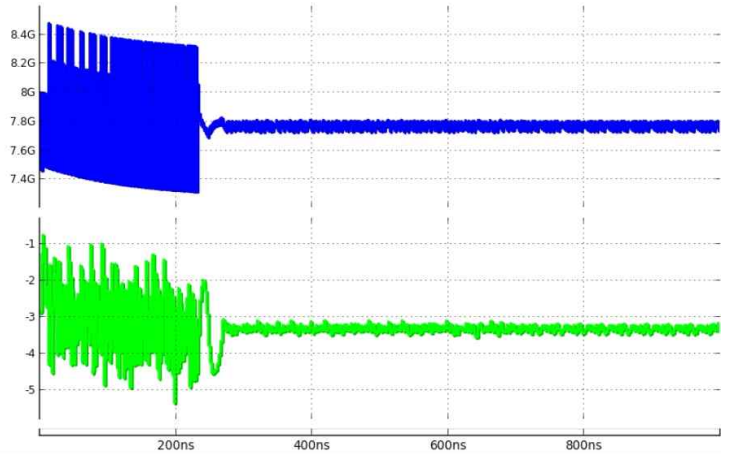
timescale : 100fs/100fs

Frequency/Phase of the Recovered Clock

- with STAY signal (Kp, Ki) : (1024, 1) <=> (64, 4)

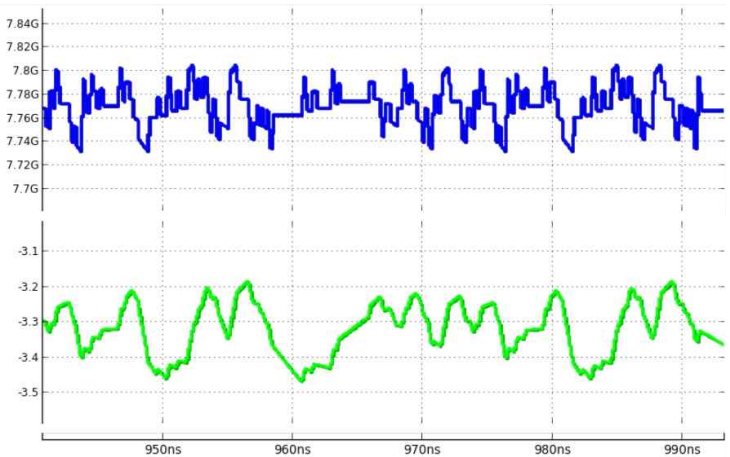
frequency

phase



frequency

phase



Phase Dithering Step :

(Kp = 1024, Ki = 1)
> (Kp = 64, Ki = 4)
≈ Proposing Method

data rate : 7.77Gb/s

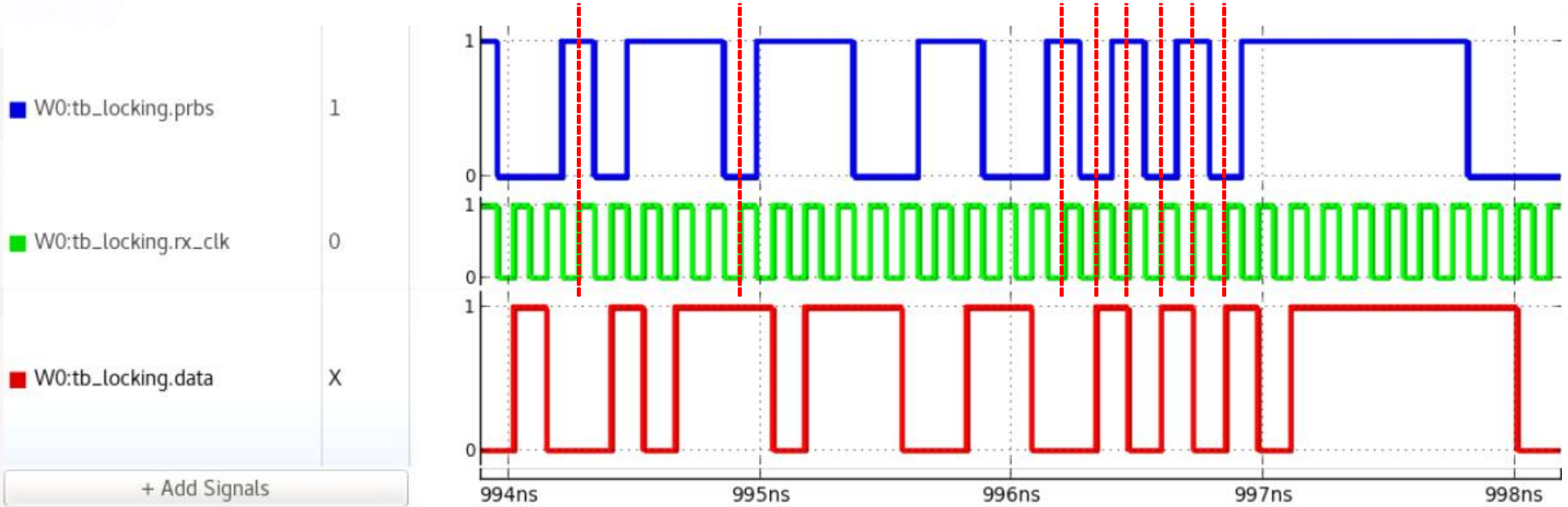
simulation time : 1us

timescale : 100fs/100fs

Frequency/Phase of the Recovered Clock

- with STAY signal (Kp, Ki) : (1024, 1) <=> (64, 4)

PRBS Data & CLK180 aligned successfully



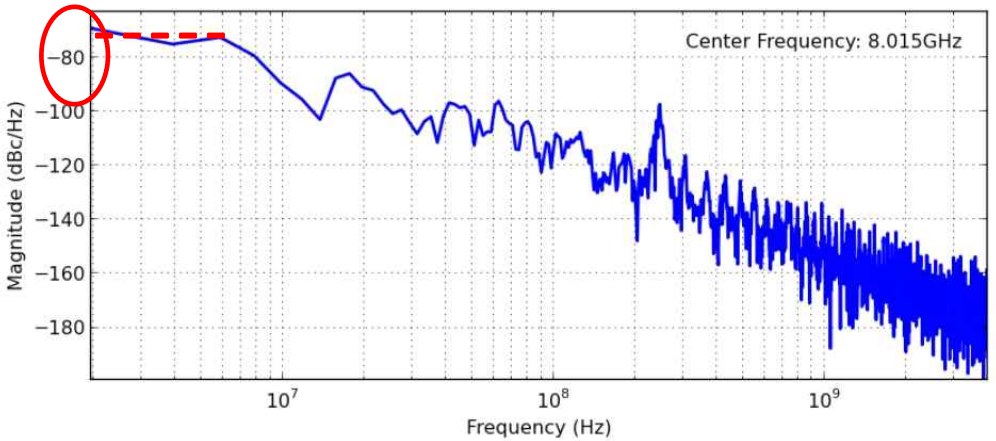
Performance Test

Phase Noise of the recovered clock

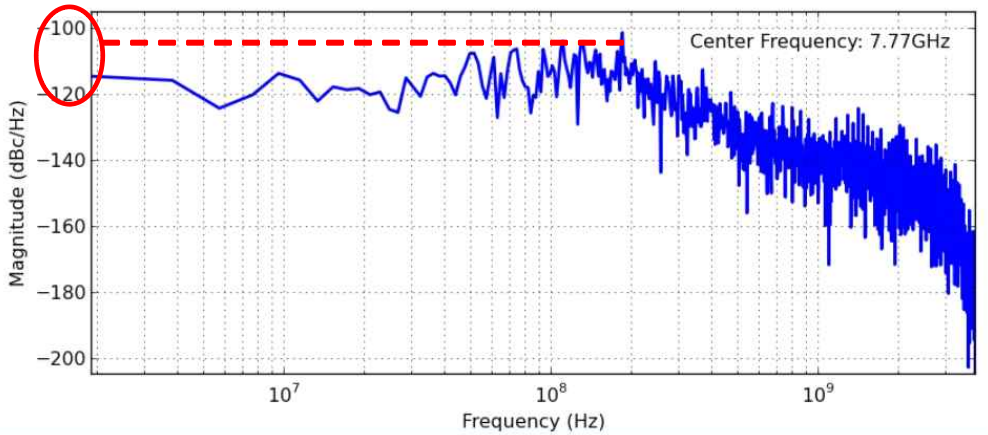
CP up/dn current = 51/49uA
Initial Control Voltage = 0.5V
Rs = 1.4k ohm, Cs = 7.2pF, Cp = 1.8pF
leakage current : 1uA, VCO range : 7Ghz ~ 9Ghz

data rate : 7.77Gb/s
cdr locking time : 2us
simulation time : 3us
timescale : 100fs/100fs

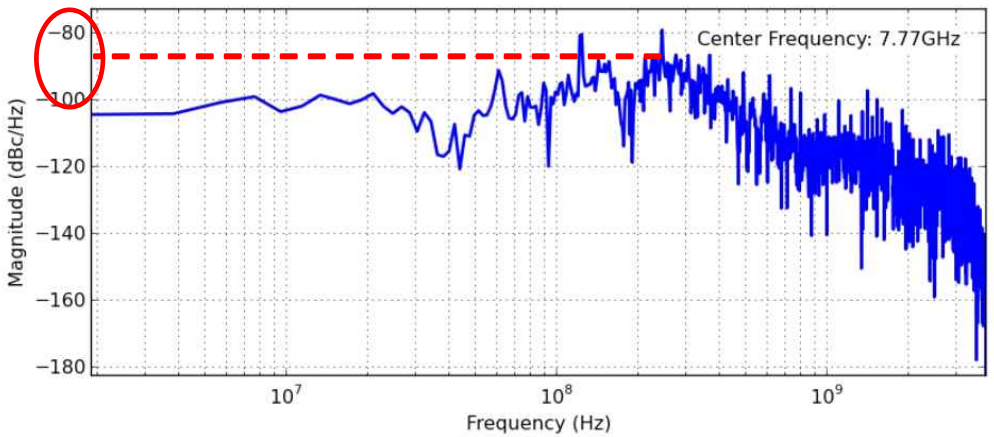
- Analog CDR



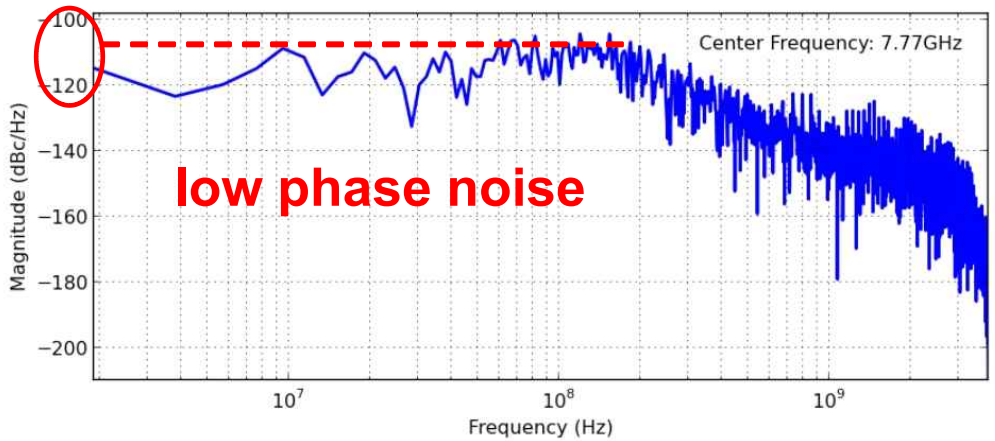
- without STAY signal (Kp, Ki) = (64, 4)



- without STAY signal (Kp, Ki) = (1024, 1)



- with STAY signal (Kp, Ki) : (1024, 1) <=> (64, 4)

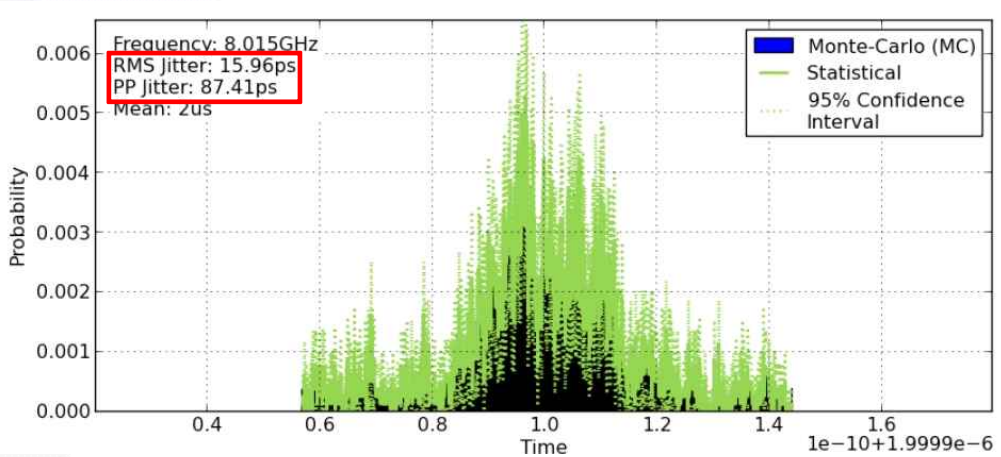


Jitter Histogram of the recovered clock

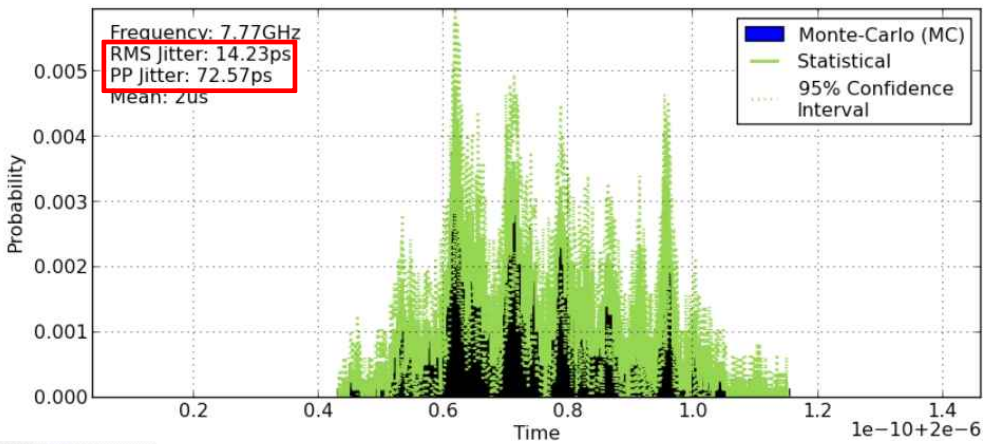
CP up/dn current = 51/49uA
Initial Control Voltage = 0.5V
Rs = 1.4k ohm, Cs = 7.2pF, Cp = 1.8pF
leakage current : 1uA, VCO range : 7Ghz ~ 9Ghz

data rate : 7.77Gb/s
cdr locking time : 2us
simulation time : 3us
timescale : 100fs/100fs

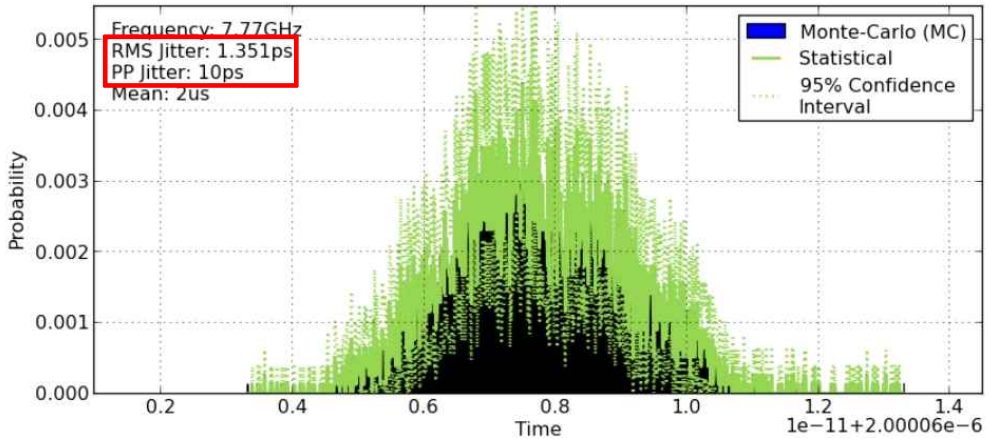
- Analog CDR



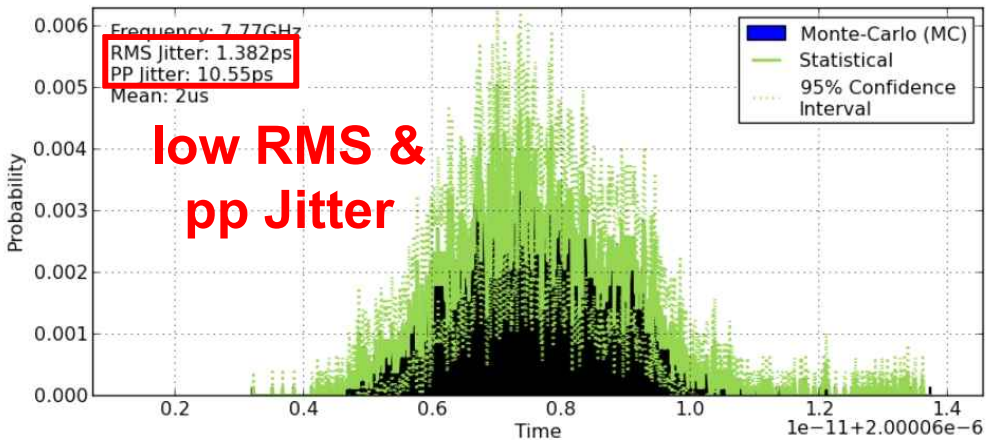
- without STAY signal (Kp, Ki) = (1024, 1)



- without STAY signal (Kp, Ki) = (64, 4)



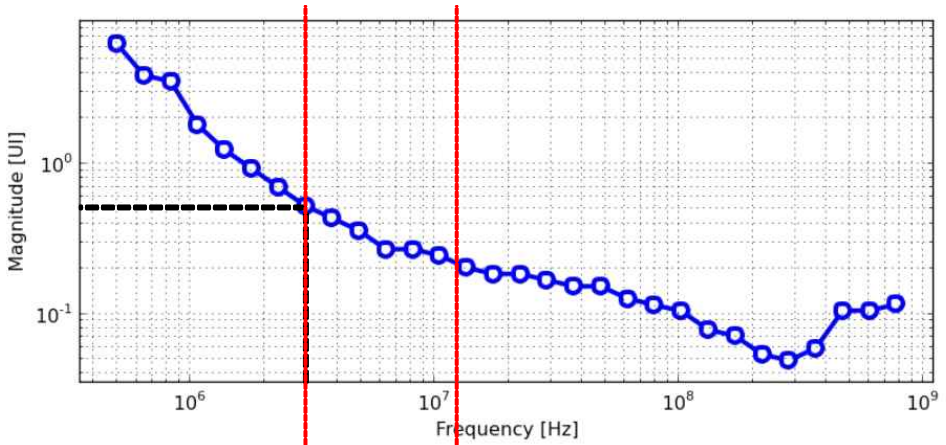
- with STAY signal (Kp, Ki) : (1024, 1) <=> (64, 4)



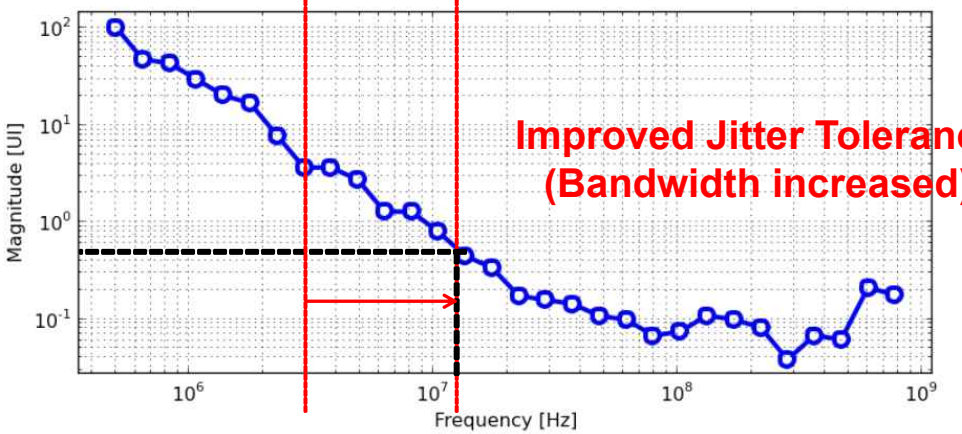
Jitter Tolerance

RJ rms (prbs) : 1ps
data rate : 7.77Gb/s
target BER : 1e-9
cdr locking time : 1us
DJ type : Sinusoidal (SJ)
simulation time : 10us
SJ freq range : 0.5Mhz ~ 100Mhz
timescale : 100fs/100fs

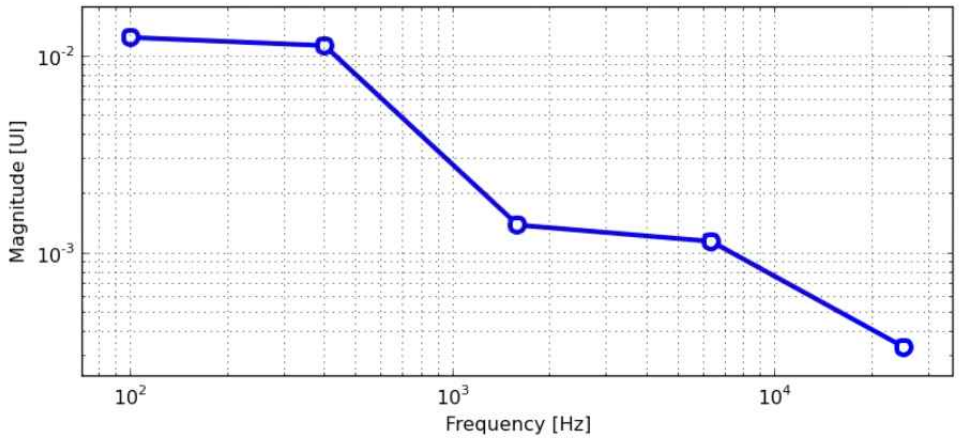
- without STAY signal (Kp, Ki) = (1024, 1)



- with STAY signal (Kp, Ki) : (1024, 1) <=> (64, 4)



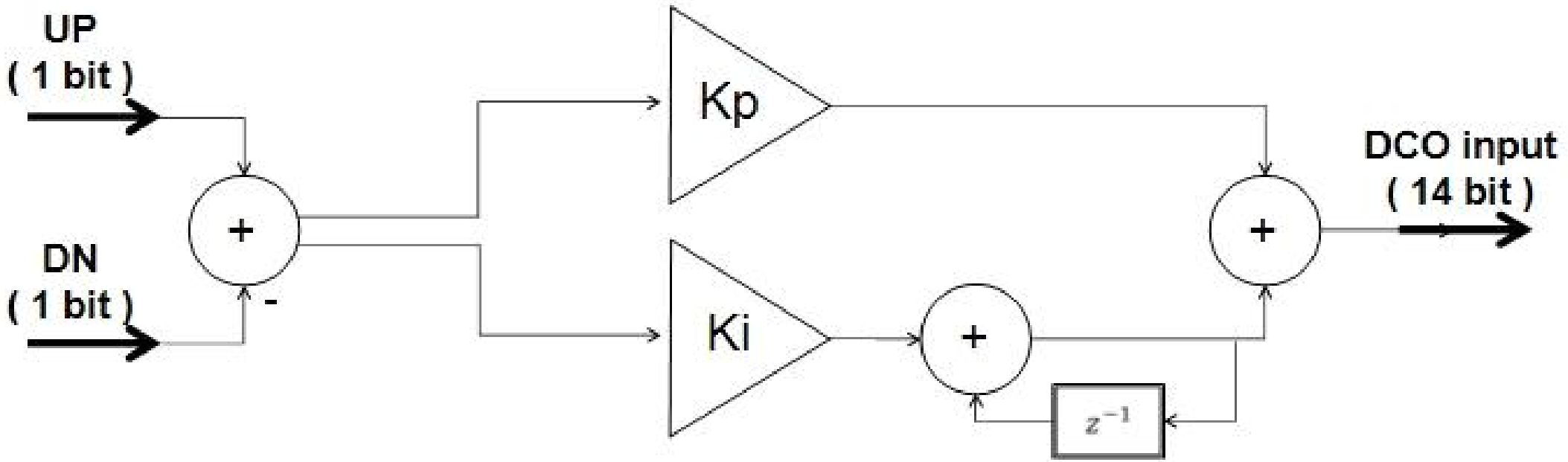
- without STAY signal (Kp, Ki) = (64, 4)



Conclusion

Summarize

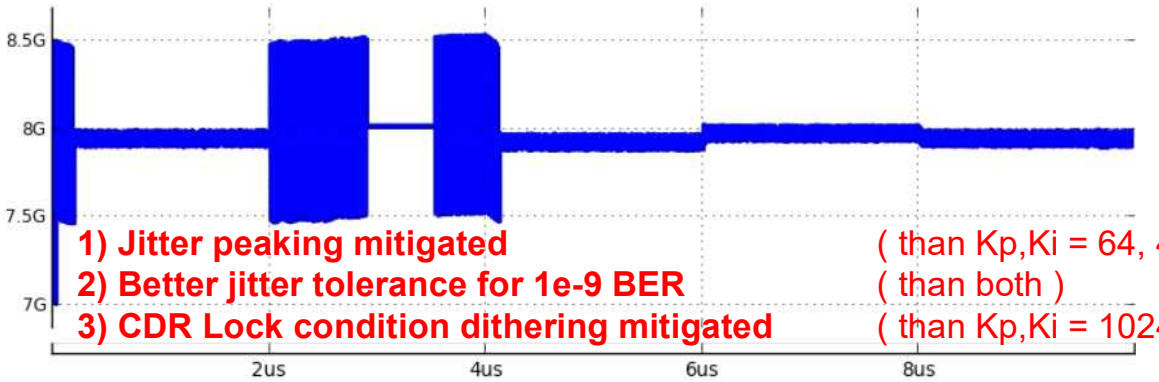
Proportional Path :
Dominant when unlocked



Integral Path :
Dominant when locked

Advantages of Proposed Architecture

Proposing Method



1) Jitter peaking mitigated (than Kp,Ki = 64, 4)
2) Better jitter tolerance for 1e-9 BER (than both)
3) CDR Lock condition dithering mitigated (than Kp,Ki = 1024,1)

data rate :

0~2us : 7.95Gb/s

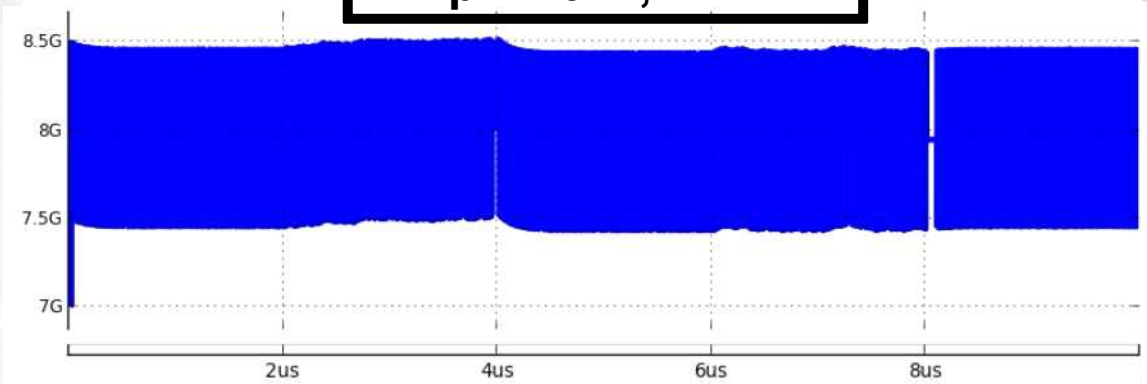
2~4us : 8.02Gb/s

4~6us : 7.93Gb/s

6~8us : 7.98Gb/s

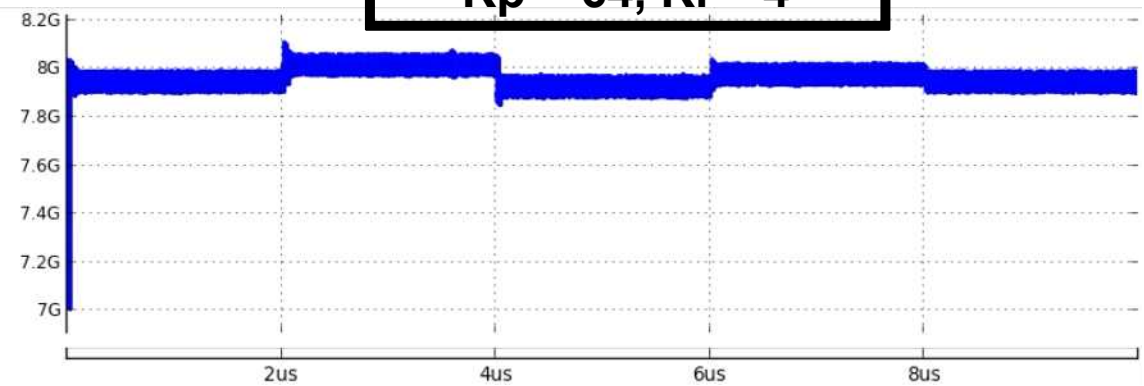
8~10us : 7.95Gb/s

Kp = 1024, Ki = 1



Issue : Large Dithering, Jitter Tolerance

Kp = 64, Ki = 4



Issue : Jitter Peaking, Jitter Tolerance

Q & A