# CDR CIRCUIT-BLOCKS: DESIGN AND VERIFICATION USING VERILOG

**LAURA CRISTINA NAVARRO GALVÁN**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER**
**FACULTAD DE INGENIERÍAS FÍSICO-MECANICAS**
**ESCUELA DE INGENIERIA ELECTRICA, ELECTRONICA Y DE TELECOMUNICACIONES**
**BUCARAMANGA**
**2017**

# CDR CIRCUIT-BLOCKS: DESIGN AND VERIFICATION USING VERILOG

**LAURA CRISTINA NAVARRO GALVÁN**

**Trabajo de grado para optar por el título de Ingeniero Electrónico**

**Director:**
**Javier Ferney Ardila Ochoa**
**Ingeniero Electrónico**

**Codirector:**
**Elkim Felipe Roa Fuente**
**Pd.H. en Filosofia**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER**
**FACULTAD DE INGENIERÍAS FÍSICO-MECANICAS**
**ESCUELA DE INGENIERIA ELECTRICA, ELECTRONICA Y DE TELECOMUNICACIONES**
**BUCARAMANGA**
**2017**

## TABLE DE CONTEND

## LIST DE FIGURES

## LIST DE TABLES

# RESUMEN

**TITULO:** CDR CIRCUIT-BLOCKS: DESIGN AND VERIFICATION USING VERILOG[*]

**AUTORES:** LAURA CRISTINA NAVARRO GALVÁN [**]

**PALABRAS CLAVES:** Clock and data recovery (CDR), Phase Locked Loop (PLL), Methodology, Flow design, high-speed interfaces.

**DESCRIPCIÓN:**

En este trabajo se presenta una metodología de diseño para circuitos de recuperación de datos y reloj (CDR) basados en una fijación digital de fase (DPLL), utilizando un flujo completo de diseño de circuitos integrados digitales. Con el fin de comprender los trade-offs de diseño y los desafíos involucrados en el proceso, el desarrollo de la metodología permite el diseño y la validación del sistema CDR digital utilizando un modelo lineal, una simulación en el tiempo y una implementación a nivel RTL. Donde, el modelo lineal y el modelo en el tiempo son implementados utilizando Matlab-Simulink y el modelo a nivel RTL se realiza usando Verilog. Dos CDR a 5 Gbps fueron diseñados y validados en tecnología CMOS de 130nm. El primer circuito fue diseñado para especificaciones generales y el segundo circuito fue diseñado para el estándar USB 3.0. Finalmente, los resultados de los diseños validan la metodología usada. Por lo tanto, el desarrollo del trabajo permite diseñar circuitos CDR aplicables a tecnologías de procesamiento digital más baratos, extremadamente bajos en potencia, insensibles al ruido de suministro y fácilmente transportados a través de múltiples tecnologías y objetivos de velocidad. Así mismo, este trabajo ayudaría a los diseñadores a acelerar el proceso de diseño de las arquitecturas CDR. Conjuntamente, el flujo de diseño utilizado incluye la implementación esquemática y layout.

[*] Trabajo de grado

[**] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenieria Electrica, Electronica y de Telecomunicaciones Director Javier Ferney Ardila Ochoa

# ABSTRACT

**TITLE:** CDR CIRCUIT-BLOCKS: DESIGN AND VERIFICATION USING VERILOG[*]

**AUTHORS:** LAURA CRISTINA NAVARRO GALVÁN [**]

**KEYWORDS:** Clock and data recovery (CDR), Phase Locked Loop (PLL), Methodology, Flow design, high-speed interfaces.

**DESCRIPTION:**

A design methodology for clock and data recovery (CDR) circuits based on a digital phase locked loop (DPLL) using a complete digital flow design is presented in this work. In order to understand the design trade-offs and the challenges involved in the process, the development of the methodology allows the validation of the digital CDR system using a linear model, a time step simulation and an implementation at RTL level. Where, the linear model and the time step simulation model are implemented using Matlab and Matlab-Simulink, respectively, and the RTL-level model is performed using Verilog. Two CDR at 5Gbps were designed and validated in 130nm CMOS technology. The first circuit was designed for general specifications and the second one circuit was designed for the USB standard 3.0. Finally, the results of the designs validate the methodology used. Thus, the development of the work allows designing CDR circuits applicable to digital processing technologies, cheaper, extremely low in power, insensitive to supply noise and easily ported across multiple technologies and speed targets. Also, this work would help designers to speed up the design process of the CDR architectures. Additionally, the used design flow includes schematic and layout implementation of the two circuits.

[*] Degree work

[**] Faculty of Physical-Mechanical Engineering. School of Electrical, Electronic and Telecommunications Engineering Director Javier Ferney Ardila Ochoa

# INTRODUCTION

As transmission rates in communication systems have been increasing up to the range of Gb / s, it has been shown that serial communication systems provide better performance in contrast to parallel transmission [1]. In addition to increased data rates, high-speed interfaces also tend to reduce the number of signals sent and integrate the status and control information that needs to be conveyed [2]. therefore, the serial communication are used for get maximum benefit in high-speed interfaces.

Clock and data recovery (CDR) circuits have been widely used in high-speed serial links. At Gbps transfer speed, channel non-idealities such as electrical noise, attenuation, interference between symbols and phase noise affect the signal integrity. Due to channel losses typically high-speed serial link does not send the clock signal in a dedicated wire [3], [4]. Therefore, the receiver block has to extract the synchronization signals from transmitted data, which is the main function of a CDR.

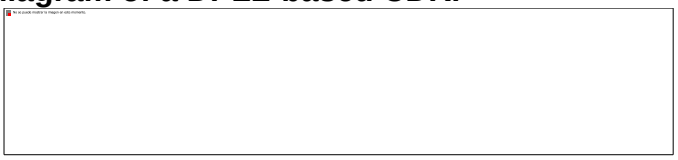
Among the main existing CDR architectures, those based on phase locked loop (PLL) stand out because of their performance and easy implementation. These types of CDRs can be categorized as: analog or digital [5]. One of the advantages of digital PLL-based (DPLL) CDR over the analog implementations is the fact that DPLL can be fully synthesized in a digital flow design [6]. In addition, a digital CDR saves area, reduces process, voltage, and temperature (PVT) variations and improves testability and observability with respect to analog implementation [6], [7].

This work consists in the development of a framework for a designing DPLL-based CDR using a complete digital flow design. The development of a methodology for this type of CDR arises from the need to design CDR circuits applicable to digital processing technologies, cheaper, extremely low in power, insensitive to supply noise and easily ported across multiple technologies and speed targets [7]. Therefore this document presents a design methodology and validation of CDR circuits through three different models, which allows to understand the design trade-offs and challenges involve in the process.

# 1. DPLL-BASED CDR

The block diagram of a CDR is shown in Fig.1. This architecture is composed of a Bang-Bang phase detector (BBPD), followed by a decimator, a digital filter formed by a proportional and an integral path, which is added to a phase integrator and subsequently to a digital-to-phase converter (DPC). The operation of this architecture is described as follows: The BBPD converts the input signal into phase errors (*Phe*) which are decimated by a factor of L, so, the signal resulting from the decimation (*Phe L*) operates at a lower speed. The signal *Phe L* passes through the digital filter driving the frequency integrator which operates at a rate of *L* Unit Interval (UI) or 1/*L*UI. The principle of operation of the phase integrator is an accumulator with *L phase* output bits which are the digital representation of the sampling phase. This sampling phase is sent to the DPC to convert them into an edge place, with respect to a local reference clock [6].

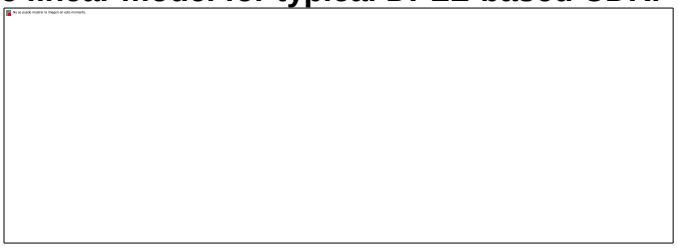
**Figure 1. Block diagram of a DPLL-based CDR.**



## 1.1. LINEARIZED SMALL-SIGNAL MODEL

The Fig.2 shows the equivalent linear model of the architecture shown in Fig.1. Here the parameter $K_{BB}$, $K_D$, phug and frug, are the BBPD gain, decimation gain, proportional and integral gains respectively. The parameter $K_{DPC}$ is the gain through the DPC and corresponds to the resolution of the DPC in units of UI per bit. $N_{EL}$ represents the latency for the whole system loop, including analog and digital pipe stages. Finally, $\varphi_{in}$ and $\varphi_{out}$ are the input data phase and output clock phase respectively.

**Figure 2. Discrete linear model for typical DPLL-based CDR.**



Using the linearized model, two interesting functions can be calculated: the jitter tolerance function, $\varphi_{in}/\varphi_{err}$ and the transfer function, $\varphi_{out}/\varphi_{in}$. To compute either of these, It is necessary to calculate the open loop transfer function, which is determined by the following equation:

The transfer function for the digital CDR model is:


(2)

The Jitter tolerance represents the ability of the CDR to recover an incoming serial data correctly despite the applied jitter and is given by:


(3)

The first parenthetical term in the equation (3) is the remaining horizontal eye opening remaining after considering the presence of Gaussian jitter with a standard deviation of $\sigma_{jin}$. This standard deviation will be discussed later in this paper.

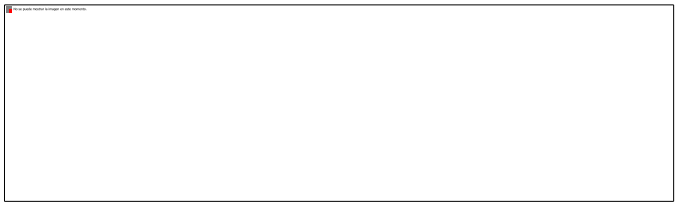
In order to understand how the discrete linear model match with the block diagram of Fig.1, it is necessary to analyze how the CDR blocks are built.

## 2. CDR BUILDING BLOCKS

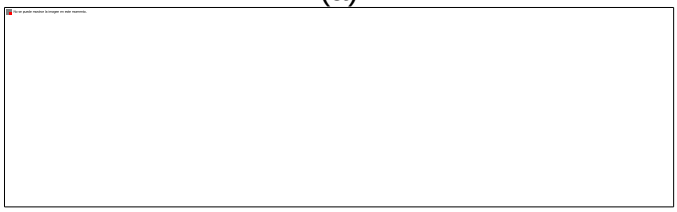
### 2.1. PHASE INTEGRATOR

As mentioned in section II, the phase integrator is a digital accumulator and it is the digital representation of the sampling phase through the digital-to-phase converter. All numbers coming out of the phase integrator correspond to a real sampling time. Therefore, the number of bits of this phase integrator in its output must be at least the number of control bits (N) of the DPC which indicate the number of phases through each UI, as shown in Fig.3(a) together with its equivalent discrete function. e.g., if the output of the phase integrator has a ”1”, this integrator will move the phase $1/2^N$ UI, which indicates the gain ($K_{DPC}$) is equal to $1/2^N$. On the other hand, if the phase integrator has N upper bits for the digital-to-phase converter together with additional $D_p$ bits of sub-resolution allows the loop gain parameter vary and the DPC will change gently. The use of additional bits makes the phase of the integrator to take several steps before moving one step of the DPC (Fig.3(b)). Therefore, now the integrator moves the phase $1/2^N$UI each $2^{Dp}$ cycles and the linear parameter $K_{DPC}$ is $1/2^{N+Dp}$, as shown in Fig.3(b).

**Figure 3. Phase integrator and the equivalent discrete function (a) Basic (b) Sub-resolution.**



(a)



(b)

### 2.2. PROPORTIONAL AND INTEGRAL PATH

The dynamics of the digital filter can be modified with the phug and frug gains, making the CDR works in different ways when these values are controlled. For an

easy digital implementation, the phug and frug gains can be applied by shifting the output of the decimator a certain amount. So, these gains can be adjusted to a power of two for multiply the decimator output easily.

## 2.3. FREQUENCY INTEGRATOR DETAILS

The frequency integrator is another accumulator with *L I* bits that are added to the phase integrator every *L*UI to compensate small phase errors in the long term. Here, it is desired to apply the same concept used for the phase integrator, where the frequency integrator has upper bits *M* and a sub-resolution $D_f$ are used. The difference with the phase integrator sub-resolution is that for this case the subresolution of the frequency integrator use a method similar to delta-sigma modulation (4Σ). The accumulator is unsigned with the lowest bits ($D_f$) get a *carry out* signal. The resulting signal is added to the upper bits *M* moving the phase integrator since $-2^M - 1$ until $2^{M-1} - 1$ given L clock cycles. The last sum of the sub-resolution used is signed and are obtained *M* + 1 bits at the output of the integral path, as shown in the Fig.4 with its equivalent discrete function.

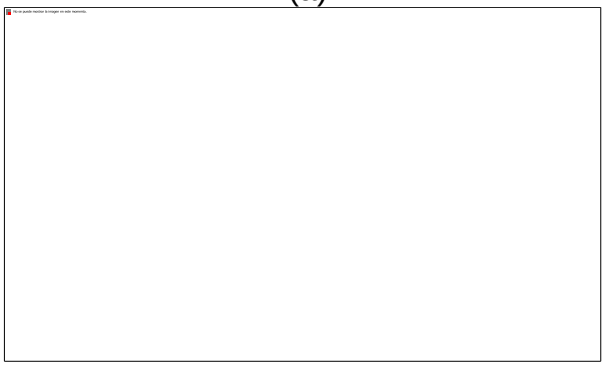
**Figure 4. Sub-resolution of frequency integrator and the equivalent discrete function.**

## 2.4. BANG-BANG PHASE DETECTOR

The Bang-Bang phase detector is commonly used in analog and digital CDR designs due to its simplicity and precision [7]. A BBPD produces a *late* or *early* pulse at each clock transition based on the relationship between the data and the phase samples. This block takes three consecutive flanks of the local clock S1, S2, and S3 and detects if its sampling clock is in delay or in advance of the data transition, as shown in Fig.5(a). If the phase bit agrees with previous data bit, the phase sample is *early* ($4\phi < 0$) if the phase bit agrees with next data bit the phase sample is *late* ($4\phi > 0$). But, if previous data bit and next data bit are the same, the phase sample has no transition. Therefore, the BBPD is a nonlinear system. The Fig.5(b) shows the transfer characteristic of a BBPD, where its output is +1 and -1 for positive and negative phase errors, respectively.

**Figure 5. (a) Three points of data samples and clock (b) BBPD characteristic curve.**

(a)



(b)

## 2.5. LINEARIZED BANG-BANG PHASE DETECTOR

The BBPD nonlinear behavior is averaged over time by the digital filter characteristics of the CDR loop. The transfer function of a Bang-Bang type PD has a finite slope due to the dithering of the clock jitter [8]. As shown in Fig.6, when CDR is locked to data input with small jitter amplitude ($\phi_{lin}$), the phase error is small and the loop behavior can be approximated by the linear region of the PD. Considering a jitter with a Gaussian source, the gain of the BBPD can be approximated by the slope at zero crossing of the transfer curve. The smallsignal gain of the phase detector is given by [7]:

 (4)

An extended explanation is presented in [9].

**Figure 6. Bang-bang PD statistical transfer function and the effect of jitter.**

## 2.6. DECIMATOR

In order to ensure proper operation of the CDR in a specific technology, it is necessary to reduce the sampling frequency of the system by performing a decimation of the BBPD output. The use of decimation via voting has recently increased due to its reduced gain and faster implementation over the use of a finite impulse response (FIR) boxcar filter [7]. Decimation by majority voting takes L phase errors from the BBPD and vote over these samples each LUI. If the samples have more *earlies*, vote -1, if the samples have more *lates*, vote +1 and if the samples have an equal number or no transition, vote 0. Its implementation is as simple as making a shift register L times of the input, add its outputs and perform the voting on this sum, as seen in Fig. 7.

**Figure 7. Decimation by voting.**

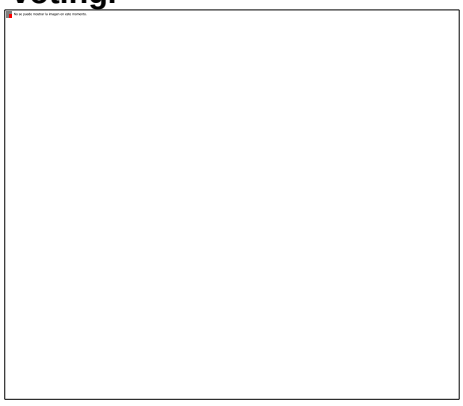


**Figure 8. Digital CDR implementation.**

Decimation via voting produces gain $K_v$, which can be determined by simulating and comparing the BBPD output and decimation output for a vote over L samples. For decimations via voting over 4, 6, 8, 10, 16 and 20 samples the results of simulation show an approximate gain of 2, 2.9, 3, 3.6, 4.2, 4.9, respectively.

## 2.7. SYSTEM DETAILS

The practical realization of this digital CDR is shown in Fig.8, where equations (5),(9) describe the relationship between output bits of each block and the design parameters of this architecture, *Phug*, *Frug*, $M$,$D_f$, $N$ and $D_p$.

(5)

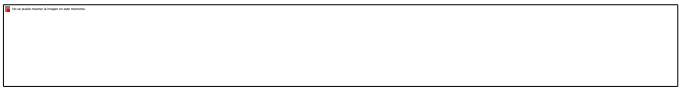
(6)

(7)

$$L\ F = \log2[(2M+1 - 1) + (2P+2 - 1)] \quad (8)$$

$$L\ phase = N + D_p \quad (9)$$

As mentioned previously, These design parameters are relations with the parameters of the linear model, so the open loop transfer function can be rewritten as



where the parameter $K_{DPC}$ and the integral path gain, varies according to sub-resolution of the phase integrator ($D_P$) and the frequency integrator ($D_F$), respectively.

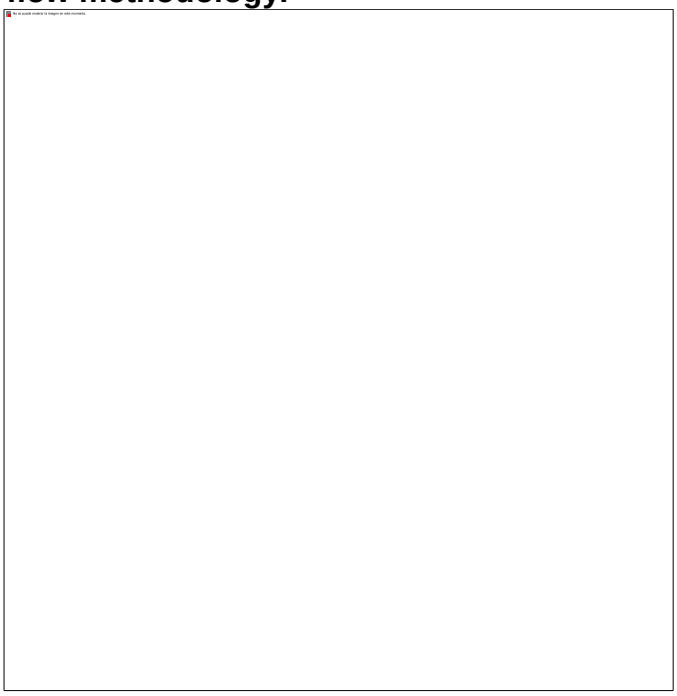
# 3. METHODOLOGY DESIGN

The Fig.9 shows the step-by-step diagram of the design flow methodology and verification for the CDR using three different models.

## 3.1. Specifications

The first step is to establish the CDR specifications such as baud rate (Gbps), tolerance of jitter tracking (PPM), frequency step (PPM), frequency slew rate (PPM/$\mu$s) among others. From these specifications, it is possible to obtain the design parameters using a spreadsheet calculator developed in this project.

**Figure 9. Design flow methodology.**



## 3.2. SYSTEM PARAMETER

The parameters are analyzed and adjusted according to the linear model and a time step simulation of the CDR using Matlab-Simulink. In the analysis of the transfer function of the linear model, the parameters are adjusted considering the

stability of the system and the maximum jitter peaking specified in the reference standard used for the design. In the same way, the jitter tolerance function obtained is compared with the tolerance mask of the standard, determining if the CDR designed is able to tolerate the specified maximum input jitter amplitude.

### 3.3. RTL LEVEL

After validating the parameters using the linear model and the time step simulation, the final parameters are used to perform the proper implementation and validation of the design at the RTL level. Here, Fig.8 is described using Verilog hardware description language. Then, the Verilog code is checking through a functional simulation.

### 3.4. SYNTHESIZED AND PLACED/ROUTED GATES

Finally, the synthesis and place and route of the design are done, adapting the CDR to a specific technology. During the synthesis, the internal structure of the device is taken into account, and restrictions are defined, such as pin assignment, the clock period, the clock uncertainty caused by jitter of the input data, among others. Then, it is verified that the synthesizer has correctly performed the synthesis of the circuit by transforming the Verilog code into logical blocks connected to each other, through post-synthesis simulation.

The place and route process locate the obtained digital blocks in the synthesis optimally, interconnecting the blocks properly, trying to minimize propagation delays to maximize the maximum frequency of operation of the device.

## 4. DESIGN EXAMPLES AND RESULTS

In order to validate the methodology developed in Fig.9, two digital CDR has been designed and validated in 130nm CMOS technology. The first circuit is designed based on the specifications given in [7] and the second one based on USB standard 3.0. Table I shows specifications for both designs.

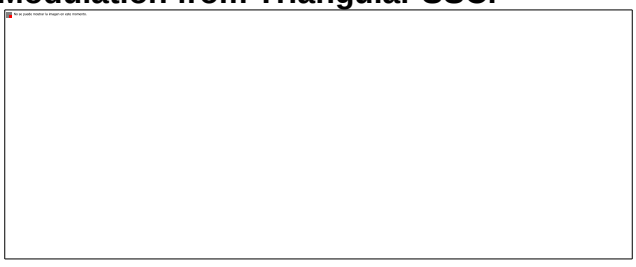
**Table 1. Specifications**

| | [7] | USB standard 3.0 [10] |
|---|---|---|
| Parameter | Value | Value |
| Baud rate [Gb/s] | 5 | 5 |
| Tolerance of jitter tracking [PPM] | ~±1000 | ~±7850 |
| Gaussian input jitter ($\sigma$) [RMS UI] | 0.03 | 0.02 |
| Freq. Steps [PPM] | ~10 | ~10 |
| Freq. Slew Rate [PPM/$\mu$s] | 1320 | 1320 |

### 4.1. GENERAL DESIGN EXAMPLE

Assuming that the first design does not have Spread- Spectrum Clocking (SSC), a jitter tracking tolerance of ±1000*PPM* and a slew rate of 1000*PPM/µs* are set. Using reasonable initial values of 5-bits or 32 steps per UI in the DPC and 3-bits for the sub-resolution in the phase integrator, the equivalent phase resolution ($K_{DPC}$) for the first system is 1/256UI or 0.0039UI. Due to the most of the standard use a 8b10b coding with at least one transition every 4 samples, a decimation by voting over 4 decisions for both paths of the digital filter are used. Therefore, the CDR clock is 4UI/cycle and the frequency integrator needs to be able to put in ±1 per cycle to get ±1000*PPM* or 0.004UI/cycle, so this frequency integrator need 1 upper bit signed to get −1 and $D_f$-bits to get very close to +1 ($2^{Df} - 1/2^{Df}$).

**Figure 10. Period Modulation from Triangular SSC.**



The minimum PPM step size of the frequency integrator is set around 10*PPM*, so this means moving the sampling phase 0.00004UI/cycle. Since a phase integrator step is 0.0039UI/cycle, the required number of cycles to move 1 step of the phase integrator is 97.5 cycles, rounding this value to a power of 2 , the frequency integrator need 128 cycles or 7 bits to move the phase integrator (1 in $2^{Df}$), i.e. these 7 bits will allow to move 1/128th of an UI per UI, or per 4 UI.

As discussed in section III, the integral path output can move the phase integrator since $-2^{M}-1$ until $2^{M-1}-1$ given L clock cycles and the phase integrator moves the phase $1/2^{N+Dp}$ each cycle. So, it can be determinate the maximum and minimum PPM offset that CDR can track, as described the equations (11) and (12). Evaluating the parameters obtained for this design, the maximum and minimum PPM offset are 975*PPM* and −1950*PPM*, respectively, which meet the spec 1000PPM. Applying a similar analysis, the effective PPM resolution is 7.6172*PPM* < 10*PPM*, as a result of equation (13).

(11)

(12)

(13)

$$frug = \frac{2^{Df} * L * Freq.slew.rate * Tcycle}{2^{-(Dp+N)}} \quad (14)$$

In addition, equation (14) define for the first circuit a value of 0.1386 for frug. If the value of phug set in 1, the proportional path can move roughly 1/1024UI per UI or 976.6PPM and this is the maximum pull-in range that the CDR can track.

### 4.2. USB STANDARD 3.0 DESIGN

USB standard 3.0 has a SSC deviation of 5000*PPM* at a rate between 30KHz and 33KHz, as shown in Fig.10. Where, $T_{SSC}$ represent the period modulation use for the SSC. Assuming that the two sides of the link have independent SSC domains and considering the superposition of other jitter components, the maximum slew rate that must be tracked is 7.85*ms/s*

Consequently, for the second circuit are set a jitter tracking tolerance of ±7850*PPM* and a frequency slew rate of 1320*PPM/μs*. Applying the same analysis as the one used for the first design, the frequency integrator needs 5 upper bit (*M* = 4 + *sign*)

and 6 dither bits ($D_f$=6) to get ±16 cycles and move 1 step of the phase integrator. The maximum and minimum PPM offset, the effective PPM resolution and the values of phug and frug are 7312.5*PPM*, −7800*PPM* 7.6172*PPM*, 2 and 0.2773, respectively, using equations (11),(14) for the second circuit.

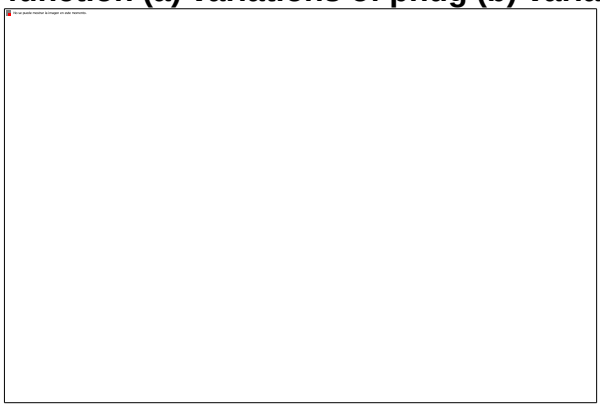
**Table 2. Linear Model Parameters**

| | [7] | USB standard 3.0 [10] |
|---|---|---|
| Parameter | Value | Value |
| *KBB* | 13.3 per UI | 19.9 per UI |
| $K_D$ | 2 (votex4) | 3 (votex8) |
| *Phug* | 1 | 2 |
| *Frug* | 0.25 | 0.25 |
| *KDPC* | 1UI/$2^8$ | 1UI/$2^9$ |
| *NEL* | 20 | 18 |

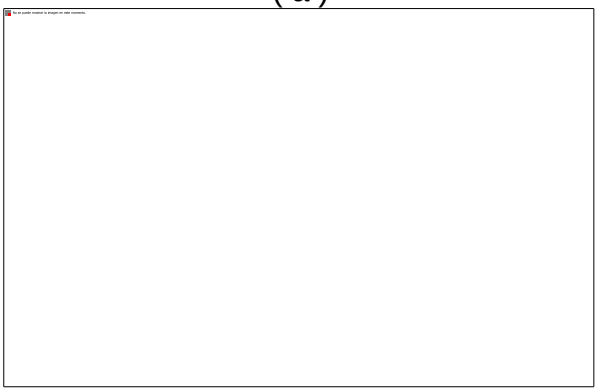
### 4.3. LINEARIZED ANALYSIS OF SYSTEM

Using previously selected specifications, it is possible to obtain the gains of the linear model for both designs. $K_{BB}$ is 13.3 per UI and 19.9 per UI for the first and second circuit, respectively, according to the equation (4). $K_D$ is equal to 2 for voting by 4 and 3 for voting by 8 as explained in Section III-F. The parameter $K_{DPC}$ is equal to 1UI/$2^8$ for the first design and 1UI/$2^9$ for the second one, using equation (9).

**Figure 11. Transfer function (a) variations of phug (b) variations of frug**

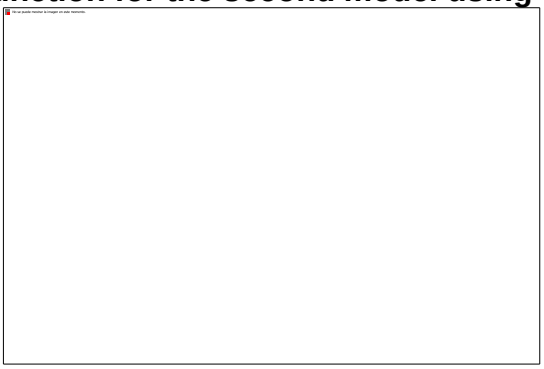


( a )



(b)

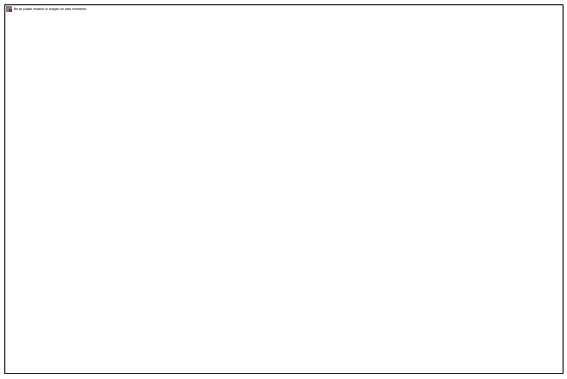
**Figure 12. Transfer function for the second model using USB standard 3.0**



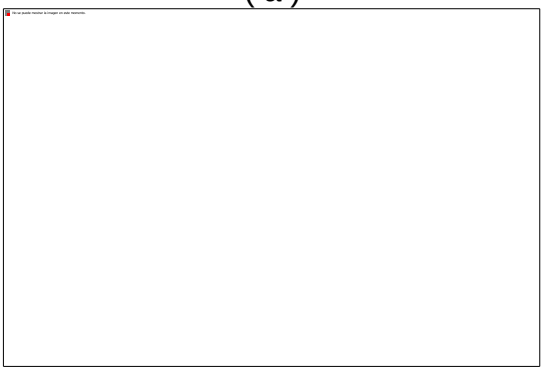
Behavior of transfer functions for the first circuit is observed in Fig.11(a) using three values of phug (0.5, 1 and 2). Previously, the phug gain for the first design was established in 1 because if phug is increased, the bandwidth expands and its peak increases indicating that it has too much gain in the loop when frug is set to 1. On the other hand, if phug decreases, the peak increases again because phug has been lowered to the point where it interacts with frug causing the poles get too

close, similar to a PLL [6]. Therefore phug is kept at 1 and it is decided to decrease the gain of the integral path.

**Figure 13. Jitter tolerance function (a) [7] (b) USB standard 3.0**



( a )



( b )

In Fig.11(b) are shown three different values of frug (0.5, 0.25, and 0.125) and how the bandwidth changes with these values. After observing the behavior for these gains and adjusted the maximum jitter peaking below 1dB, it was decided to set the gain of the integral path in 0.25. The problem with this value of frug is that it is not possible to multiply a fractional number in integer math, but the integral gain is $frug/2^{Df}/L$, so there are two ways to solve this, either add bits to the bottom of the frequency register, increase $D_f$, or increase the decimation rate just through the frequency path. If the decimation rate is increased, voting over 16, instead of voting over 4, then, the integral path has an additional reduction of 4. Consequently, the output of frequency register can only change every 16UI, but it still gets added to the phase integrator every 4UI. This means that the PPM tracking range does not change at all, but it changes how fast the CDR can change the frequency integrator. so the frequency integrator went from 8 bits at 1.25 GHz to being 8 bits at 312.5 MHz and its minimum PPM resolution is now $0.0039/2^7/16 * 10^6$ ' 2PPM.

For the circuit based on USB standard 3.0, the maximum peaking of the transfer function must be less than 2 dB [10]. Therefore, the values of phug and frug are 2 and 0.5, respectively, as shown in Fig.12. As explained before, the gain of frug can not be a fractional number, so the decimation of integral path vote over 16 and the gain of frug is now 1. In order to guarantee the proper operation of system, stability test was performed for all cases.

Parameters listed in Table II are used to obtain Jitter tolerance curve for both systems, as shown in Fig.13. The two curves acquired are shown along with the jitter tolerance mask established for the USB standard 3.0. In Fig.13(b) it can be seen that the chosen parameters for both circuits easily beat the jitter tolerance limit, indicating that the designed system is appropriated.

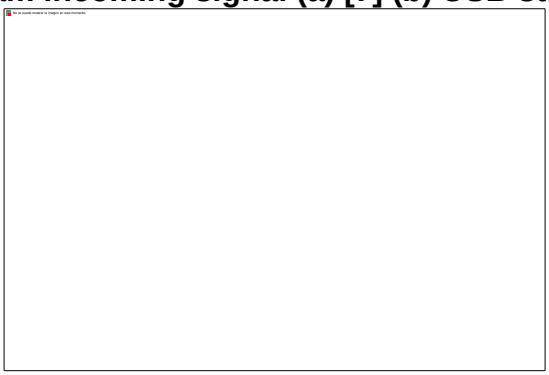
**Table 3. Digital CDR Parameters**

| | [7] | USB standard 3.0 [10] |
|---|---|---|
| Parameter | Value | Value |
| *Phug* | 1 | 2 |
| *Frug* | 1 | 1 |
| *V ote P* | 4 | 8 |
| *V ote I* | 16 | 16 |
| $D_f$ | 7 | 6 |
| *L I* | 8 | 11 |
| $D_p$ | 3 | 4 |
| *N* | 5 | 5 |
| *L phase* | 8 | 9 |

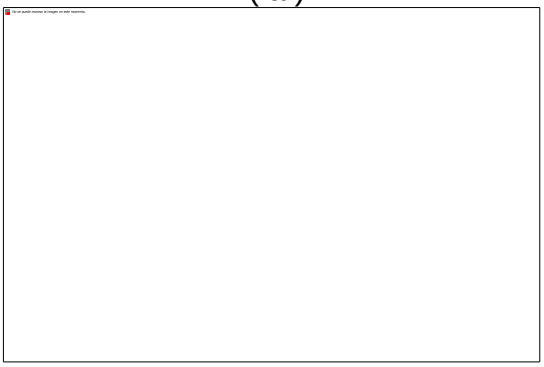
## 4.4. IMPLEMENTATION DETAILS

The signal present at the receiver for all the results described above for both circuits is shown in Fig.14. These input signals pass through the system obtaining the output observed in Fig.15. Clearly, it can be observed that the input jitter is considerably reduced, indicating that the parameters obtained are appropriated for both cases. In this section, it will be described how the parameters obtained from the linearized model coincide with the implementation of Fig.8 and the parameters of Table III by mean of a step-time simulation. Due to the DPC will have 32 steps or 32 possible clock

**Figure 14. Eye diagram incoming signal (a) [7] (b) USB standard 3.0 [10]**



( a )



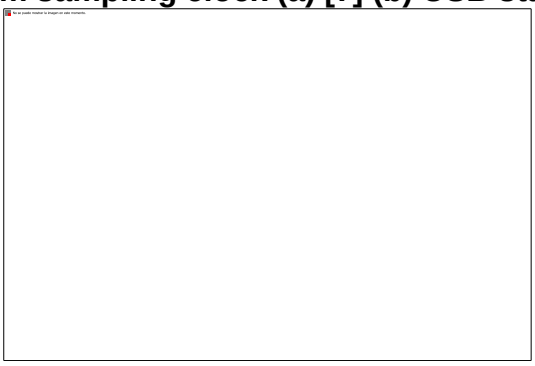
( b )

edge positions in both systems, the phase integrators according to equation (9) will have 8 and 9 output bits (*L phase*) , respectively. Therefore, the output of the phase integrators will take values between 0 and 32 as it is the digital representation of the sampling phase through the DPC. The Fig.16 shows the phase integrators output for each system. In Fig.16 it can be seen that the second-order system used for the circuit has an underdamped behavior and a step value is maintained around half-bit period upon reaching establishment time, which is desired behavior.

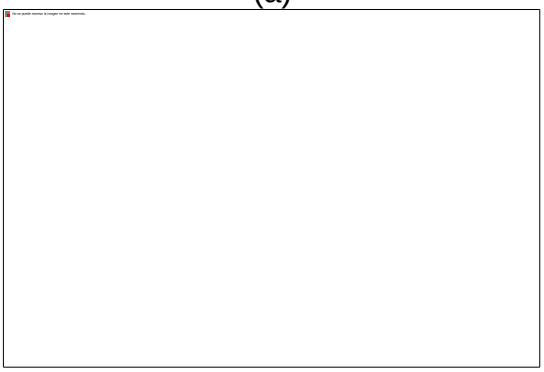
### 4.5. DIGITAL FLOW ( SYNTHESIS )

Following the methodology of Fig.9 and after having validated the parameters selected for linear model and the timestep simulation for both designs, the models of CDR are implemented in software according to Fig.8, using HDLVerilog. Here each block is behaviorally validated by test bench.

**Figure 15. Eye diagram sampling clock (a) [7] (b) USB standard 3.0 [10]**

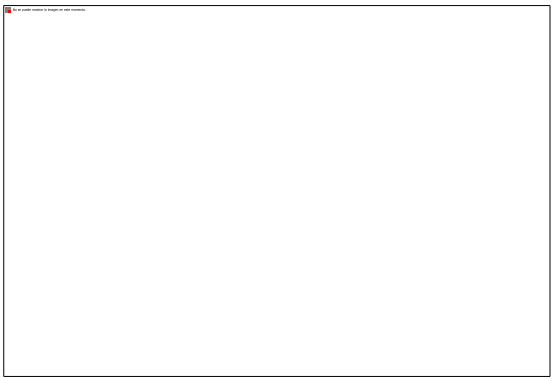


(a)

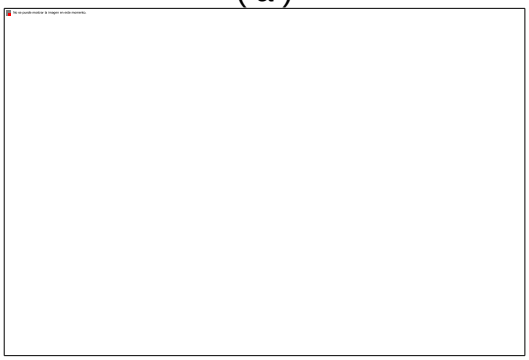


( b )

**Figure 16. Phase Integrator Output (a) [7] (b) USB standard 3.0 [10]**



( a )



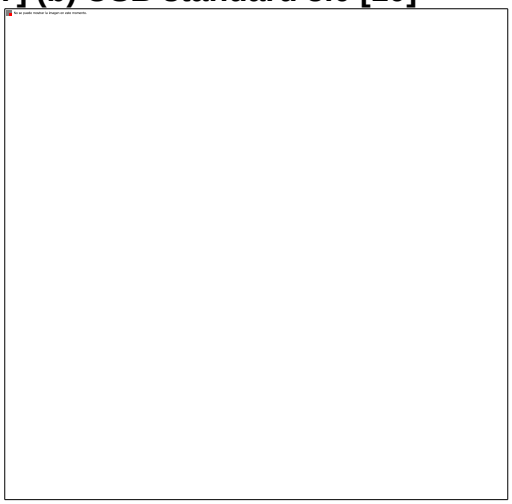
(b)

## 5. THE CONFIGURATION CAN BE CLASSIFIED INTO TWO MAIN PARTS:

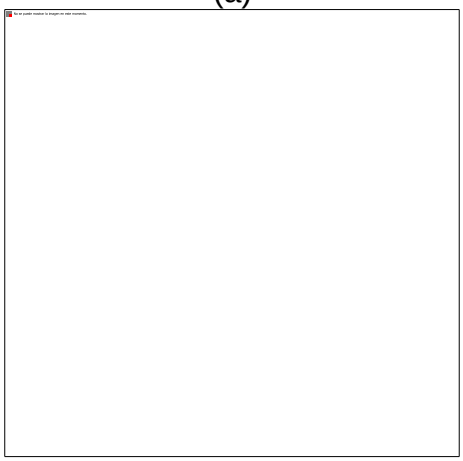
1. Script configuration: In this script, the library of the geometric description of the cells, time description and parasitic extraction are charged. Here, the behavior of the desired circuit for synthesis is described. Furthermore, the script has the proper commands of the synthesis and the reported time restrictions, used cells, used area, power consumption described in leakage for each cell, net-list, among others.

2. Restrictions: In this section, the limit parameters are defined such as the clock period, the clock uncertainty caused by jitter of the input data, rising and falling edge, capacitive charge for each output and final delay time in which clock arrives at inputs. These declarations are made in order to verify the maximum operation frequency of the circuit.

**Figure 17. Layout (a) [7] (b) USB standard 3.0 [10]**



(a)



(b)

Table IV shows the results of the synthesis of each design. The most critical path is the decimation over 16 samples because it is a parallel-serial shift register and circuit takes 16 clock cycles before generating an output. Fig.17 shows the layout view obtained for both designs. The area of the layout were of 120*μm* for the first design and 95*μm* for the second one.

**Table 4. Synthesis Results Of Each Design**

| Synthesized design | [7] | USB standard 3.0 [10] |
|---|---|---|
| Frequency [MHz] | 555 | 555 |
| Timing Slack | 0*ps* | 1*ps* |
| Total Power (mW) | 2.894 | 1.7748 |

## 6. SUMMARY

In this paper, a methodology for CDR digital circuits is stated. Since characteristics of the frequency response of CDR are determined by jitter tolerance and jitter transfer characteristics, developed designs are analyzed and evaluated for three different models. Based on the presented tables and figures, it is clear that the results validate the methodology for used. Thus, this work would help designers to speed up the design process of the CDR architectures.

## REFERENCES


[1] R. R. D. et al., “Parallel vs. serial on-chip communication.” in *Proceedings of the 2008 International Workshop on System Level Interconnect Prediction*, Dec 2007.

[2] J. Moreira and H. Werkmann, *An Engineer’s Guide to Automated Testing of High-speed Interfaces*. Boston : Artech House, Inc., 2010.

[3] U. D. M.Becerra and D. Avendano,˜ “Sistema recuperador de datos y senal˜ de reloj (cdr) con sobre-muestreo 3x (3x oversampling) utilizando la tecnolog´ıa de 0.5$\mu$m de amis,” [Online] Available: http://lfportal.iteso.mx/documents/10448/0/D-2070164 .pdf/752c9ef3-8c0b-42d1-9416-1c97dd4aa 652.

[4] D. Barrientos, V. Gonzalez,´ M. Bellato, A. Gadea, D. Bazzacco, J. M. Blasco, D. Bortolato, F. J. Egea, R. Isocrate, A. Pullia, G. Rampazzo, E. Sanchis, and A. Triossi, “Multiple register synchronization with a high-speed serial link using the aurora protocol,” *IEEE Transactions on Nuclear Science*, vol. 60, no. 5, pp. 3521–3525, Oct 2013.

[5] M. t. Hsieh and G. E. Sobelman, “Architectures for multi-gigabit wirelinked clock and data recovery,” *IEEE Circuits and Systems Magazine*, vol. 8, no. 4, pp. 45–57, Fourth 2008.

[6] “Isscc 2011 / tutorials,” in *2011 IEEE International Solid-State Circuits Conference*, Feb 2011, pp. 510–511.

[7] J. L. Sonntag and J. Stonick, “A digital clock and data recovery architecture for multi-gigabit/s binary links,” *IEEE Journal of SolidState Circuits*, vol. 41, no. 8, pp. 1867–1875, Aug 2006.

[8] Y. Sun and H. Wang, “Analysis of digital bang-bang clock and data recovery for multi-gigabit/s serial transceivers,” in *2009 IEEE Custom Integrated Circuits Conference*, Sept 2009, pp. 343–346.

[9] J. Lee, K. S. Kundert, and B. Razavi, “Modeling of jitter in bang-bang clock and data recovery circuits,” in *Proceedings of the IEEE 2003 Custom Integrated Circuits Conference, 2003.*, Sept 2003, pp. 711–714.

[10] *Universal Serial Bus 3.0 Specificaton*, jul 2013.

## BIBLIOGRAPHIC


BARRIENTOS Diego, GONZÁLEZ Vicente, BELLATO Marco, GADEA Andrés, Dino Bazzacco, Jose Blasco, Damiano Bortolato, Francisco Egea, Roberto Isocrate, Alberto Pullia, Gabriele Rampazzo and Enrique Sanchis, “Multiple register synchronization with a high-speed serial link using the aurora protocol,” IEEE Transactions on Nuclear Science, vol. 60, no. 5, pp. 3521–3525, Oct 2013.

ISSCC 2011 / tutorials,” in 2011 IEEE International Solid-State Circuits Conference, Feb 2011, pp. 510–511.

JRI LEE, KENNETH KUNDERT, and BEHZAD RAZAVI, “Modeling of jitter in bang-bang clock and data recovery circuits”, in Proceedings of the IEEE 2003 Custom Integrated Circuits Conference, 2003., Sept 2003, pp. 711–714.

MING-TA HSIEH AND GERALD E. Sobelman, “Architectures for multi-gigabit wirelinked clock and data recovery,” IEEE Circuits and Systems Magazine, vol. 8, no. 4, pp. 45–57, Fourth 2008.

MOREIRA Jose, WERKMANN Hubert, An Engineer’s Guide to Automated Testing of High-speed Interfaces. Boston: Artech House, Inc., 2010.

ROSTISLAV (Reuven) Dobkin, Arkadiy Morgenshtein, Avinoam Kolodny, Ran Ginosar ,“Parallel vs. serial on-chip communication.” in Proceedings of the 2008 International Workshop on System Level Interconnect Prediction, Dec 2007.

SONNTAG Jeff and STONICK John, “A digital clock and data recovery architecture for multi-gigabit/s binary links”, IEEE Journal of Solid-State Circuits, vol. 41, no. 8, pp. 1867–1875, Aug 2006.

Ulises Domínguez, Marcos Becerra and Dr. Víctor Avendaño, “Sistema recuperador de datos y se˜nal de reloj (cdr) con sobre-muestreo 3x (3x oversampling) utilizando la tecnolog´ıa de 0.5_m de amis,” [Online] Available: http://lfportal.iteso.mx/documents/10448/0/D-207016-4.pdf/752c9ef3-8c0b-42d1-9416-1c97dd4aa652.

UNIVERSAL SERIAL Bus 3.0 Specificaton, jul 2013.

YEHUI SUN and HUI WANG, “Analysis of digital bang-bang clock and data recovery for multi-gigabit/s serial transceivers,” in 2009 IEEE Custom Integrated Circuits Conference, Sept 2009, pp. 343–346.