Modeling and Characterization of VBUS Power Discharge for Embedded Superspeed USB Host/Devices

Maneesh Kuamr Pandey[1], Mohit Goyal[2], Parul Kumar Sharma[3] and Rohit Sharma[4]

[1]*{maneesh.pandey,mohit.goyal,parulk.sharma}@nxp.com*
[4]*rohit@iitrpr.ac.in*
[1]NXP Semiconducotrs, Noida, India
[4]Department of Electrical Engineering, Indian Institute of Technology Ropar, India

***Abstract*—After the expiry of serial ports for desktops and laptops, serial interface came into picture and out of serial interface USB (superspeed USB) has become the most common peripheral interface. In addition, built-in (embedded) USB applications are on the rise around the world. But development teams that implement the USB interface in the embedded system can easily produce slight variations such that interoperability with other USB devices could result in uncertain results. These uncertain results generally lead to the enumeration as well as data transfer failures in the embedded system. Sometimes, enumeration failure occurs due to system power management efforts. In an embedded system, it might be desirable to have a USB host dynamically power down an attached USB device as part of its power management scheme. The problem is, the USB 2.0/3.x specification does not provide for a discharge mechanism on the host/device side for VBUS power. Till now no analytical estimation has been provided for this limitation.**

In this paper mathematical modeling and its characterization has been done to resolve above limitation. Electrical characterization of VBUS power discharge mechanism is explored extensively to determine discharge time. This mathematical model has been developed to help embedded system developer and designer to get exact discharge time for their system according to their devices used. Further this model and its characterized results have been verified using circuit simulation as well as at real 16nm SoC as USB 2.0/3.x host and four USB implementers Forum (USBIF) certified devices. First time this kind of analytical estimation has been provided through mathematical modeling and characterization.

***Index Terms*—Post Silicon Validation, Analog Bench Validation, Superspeed USB, USB 3.x, Serial Interface, Modeling, VBUS Power Discharge.**

I. Introduction

Most common serial interface of choice in all modern electronics communication and networking devices for fast and reliable electronic media transfer and control is Universal Serial Bus (USB) [1]. It's use ranges from mostly all hand-held devices to memory storage devices and peripherals like keyboard, mouse etc. A USB 2.0/3.x [2] [3] device controller is present in all these devices to control the communication between host and device.

With the introduction of USB 3.0 [2], power consumption has been optimized and extremely fast and flawless data transfer is possible. This is facilitated by its highly efficient and productive algorithm. When a super-speed device is plugged into the host, hardware based state transitions and negotiations start occurring between them. This is shown in figure-1, which is a flow chart representation for embedded host/device driver approach for link train. Once this sequence is complete, the LTSSM (link Training and Status State Machine) enters into a U0 state and the link training is complete and ready for enumeration sequence [4].

The most commonly used open source software for link train has fixed delay of 50ms, as is described in flowchart (Figure-1), for getting proper power cycle and enumeration of interoperability.

However, this timing delay specification is not defined in USB 2.0/3.x specification. This timing delay is to facilitate the power management in the link train which happens through discharge mechanism between the host/device when the VBUS is turned off and is an important part of link train. Unavailability of such a specification results in enumeration failures with USB devices from different vendors.

This issue of enumeration failure in SoCs during analog bench validation has been discussed before in various articles providing different reasons of failures. Article [5] represents this failure as one of the case study of the issues which were observed in embedded systems and resolved the same using methodology using both low level JTAG programming and application level Linux programming [6]. However, this lacks the discussion of hardware and VBUS discharge mechanism which is the true cause of the issue.

Article [7] discusses couple of issues faced during post-silicon debug related to USB enumeration. The issues mentioned are related to repeated enumeration where the process hangs and time-out failures were observed. Again, these do not talk about the root cause - the VBUS discharge characteristics. While authors in article [8] discuss the VBUS discharge limitation but not provided discharge mechanism, and their solutions/results are based on a single experiment only. In

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00074

practice, these failures occur frequently and the severity of issue varies according to the design. One can not productize the design in the presence of such a bug. Therefore, there is a need for generalize analytical estimation that can aid developers and designers.

The aim of this article is to discuss the VBUS power discharge mechanism in detail and to provide a mathematical solution through VBUS circuit modeling and its analysis. The resulting equations are plotted which provide theoretical expected discharge times. We also show the variation of this discharge time with different circuit parameters. Further, the simulation of equivalent circuit to study VBUS discharge time has been done. Finally, we prove the correctness of the model and its characterization with data obtained from experiments on 16nm SoC embedded USB system. We begin by introducing the VBUS power discharge problem description. In section II, we discuss the motivation for this modeling and characterization. We then model the embedded circuit and analyze it in section III. The following section IV will provide the characteristics of the equivalent model in form of plots obtained from equations derived in section III. In section V, the simulation of equivalent circuit is shown and the results are discussed. The theoretical results will be compared with actual silicon data in section VI. Section VII concludes the paper by discussing how this modeling can help designers and researchers to get very accurate VBUS discharge time in embedded USB system and consequently, the time delay required in the flowchart of figure-1.

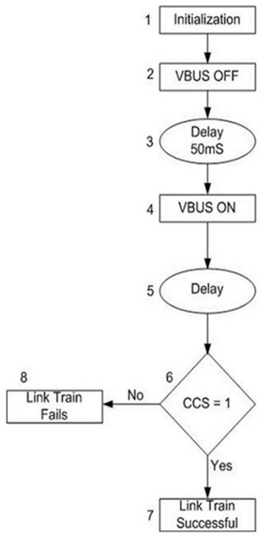

Fig. 1. Embedded Host/Device Driver Approach for Link Train

II. Motivation

USB Designers [9] make a lot of effort to make a system with efficient power management since energy consumption management on USB is very important in modern embedded systems. It is not very crucial in systems like desktops, but it becomes very important in case of small handheld and portable devices. A way to achieve this is to make the host power down the attached USB device dynamically. However, the problem arises due to lack of host/device VBUS discharge mechanism description in USB 2.0/3.x when the host VBUS is turned off. As the VBUS circuit is powered down, capacitors on this lane start to discharge (figure-2). The discharge path will be through the device and as such, the time taken for this discharge will depend on devices' impedance profile. In article [8], discharge event of a USB device is discussed where the specified device takes up to 6-7 seconds to fully discharge.

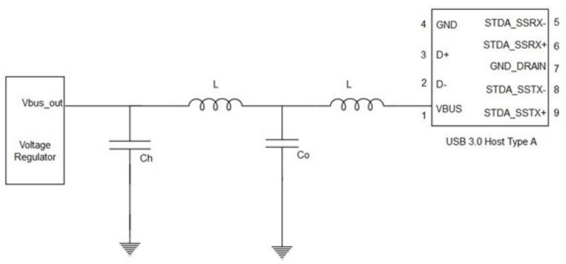

Fig. 2. VBUS Power Discharge Circuit on Hardware

The analog bench validation engineers check their USB 3.x host with various USB devices from various vendors. The following debug is very time consuming and a lot of debugs narrow down to the above-mentioned issue.

As mentioned earlier, there is no specification defined for USB for delay in link train during power management [10] issue and as such, the time for which the host should wait after turning off the VBUS is unknown. Therefore, when USB developers want system to cycle the host's VBUS power, they tend to assert an arbitrary experienced based delay of 50-150ms in the software (see figure-1). The delay in order of seconds seem way too much and the designer is reluctant to assert such a delay. When the host software turns off the VBUS power for only few hundred milliseconds, the VBUS might not have discharged to a proper voltage level and as a result, problems may arise. The device might be in an unknown state. This indeterminate condition can often lead to failure during device attach or connect stages after power returns. During the link power management sequence, if the host software turns off the power supply then it must take the voltage to ground and keep it to that level until discharge is complete and the device is reset.

The specifications do not state and the vendors do not provide a value for minimum "off" time; developers and engineers typically use the trial method to guess such a time. A number of device are tested and a value of "safe" off time is decided. This will be dependent on number of devices and will not be a generalized number.

The above arguments prove the need and motivation for a correct model of the VBUS discharge time. In this paper, we have modeled and characterized VBUS circuit and present the required delay (off time) in terms of host and device characteristics. The results obtained from the model comply with circuit simulation as well as the measurements in the lab which strengthens our claim and equations.

III. Analytical Modeling Of VBUS Power Discharge Circuit

Here we analyze the VBUS circuit and model it in s-domain. The equivalent s-domain circuit of figure-3 is shown below.

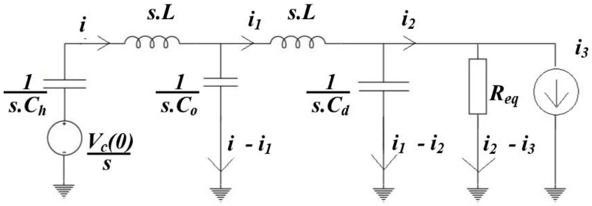

Fig. 3. VBUS Power Discharge Equivalent Circuit

The VBUS discharge occurs mainly because of the charge on the host capacitor discharging through the device. To model the device, we realize that it can-not be modeled by a single resistive load, simply because it involves a lot of circuitry which does not have a constant resistance. We therefore use the term R_{eq} to model the device resistance and use a two-value model. That is, as the capacitor on host discharges and its value reduces, we assume that the resistance of the device will also change. We define a cutoff voltage V_{cut} above which device can be said to have lower resistance (R_{low}) and below which the device is said to have higher resistance R_{high}. A diode (or a constant current source) models the leakage current through the device. C_d is the device capacitor which is present on essentially all USB devices.

We schematize this model and perform analytical as well as simulation based analysis. The equivalent time domain model used in simulation is shown in Fig. 6. Starting the analysis by writing KVL equations for the circuit in Fig.3.

$$\frac{V_c(0)}{s} - \frac{i(s)}{s.C_h} - i(s).s.L - i_1(s).s.L - (i_2(s) - i_3(s)).R_{eq} = 0 \tag{1}$$

$$\frac{V_c(0)}{s} - \frac{i(s)}{s.C_h} - i(s).s.L = \frac{i(s) - i_1(s)}{s.C_o} \tag{2}$$

$$\frac{i_1(s) - i_2(s)}{s.C_d} = (i_2(s) - i_3(s)).R_{eq} \tag{3}$$

Finding the values of i_1 and i_2 from (2) and (3) and substituting in (1)

$$\begin{aligned}&\frac{V_c(0)}{s} - \frac{i(s)}{s.C_h} - i(s).s.L - i(s).s.L - \frac{i(s).C_o.s.L}{C_h} - i(s).C_o.L^2.s^3 \\ &+ V_c(0).C_o.s.L - \frac{R_{eq}.i(s)}{(R_{eq}.Cd.s+1)} - \frac{R_{eq}.i(s).C_o}{C_h(s.R_{eq}.C_d+1)} \\ &- \frac{i(s).C_o.L.s^2.R_{eq}}{(s.R_{eq}.C_d+1)} + \frac{V_c(0).C_o.R_{eq}}{(R_{eq}.C_d.s+1)} + \frac{i_3(s).R_{eq}}{(R_{eq}.C_d.s+1)} = 0\end{aligned}$$

Since, most of the analysis is with DC voltages, s will be small. The LC network is essentially a low pass filter and in practice, the numerical values of L and C_0 will be very small in comparison to C_d, C_h and R_{eq}. To solve the above mathematically complex equation, we thus make the justified approximations and simplify the equation to:

$$\frac{V_c(0)}{s} - \frac{i(s)}{s.C_h} - \frac{i(s).R_{eq}}{(R_{eq}.s.C_d+1)} + \frac{i_3(s).R_{eq}}{(R_{eq}.s.C_d+1)} = 0 \tag{4}$$

the above equation can be simplified to

$$i = \frac{V_c(0).C_h.(s.R_{eq}.C_d+1)}{(s.R_{eq}.(C_d+C_h)+1)} + \frac{i_3(s).R_{eq}.s.C_h}{(1+R_{eq}.s.(C_h+C_d))} \tag{5}$$

simplifying further and taking the approximations justified for (4) - for small C_o :- $i_1(s) \simeq i(s)$,the following equation is obtained

$$i_2(s) - i_3(s) = \frac{V_c(0).C_h}{(R_{eq}.s.(C_d+C_h)+1)} - \frac{i_3(s)}{(1+R_{eq}.s.(C_d+C_h))} \tag{6}$$

Finally, the voltage across R_{eq} can be written as

$$V_f = (i_2(s) - i_3(s)).R_{eq} \tag{7}$$

$$V_f = \frac{V_c(0).C_h}{(C_h+C_d).(s+\frac{1}{R_{eq}.(C_d+C_h)})} + \frac{i_3(s)}{(C_h+C_d).(s+\frac{1}{R_{eq}.(C_h+C_d)})} \tag{8}$$

Taking the inverse Laplace transform to get values in time domain

$$V_f(t) = \frac{V_c(0).C_h - i_3}{C_h + C_d}.e^{\frac{-t}{Z.(C_d+C_h)}} \tag{9}$$

and finally, the VBUS discharge time can be computed as

$$t = R_{eq}.(C_h + C_d).\ln(\frac{V_c(0).C_h - i_3}{V_f(C_h + C_d)}) \tag{10}$$

Since R_{eq} takes two different values, we can represent this equation as:

$$t = \begin{cases} R_{low}.(C_h + C_d).\ln(\frac{V_c(0).C_h - i_3}{V_f(C_h+C_d)}) & \text{if } \mathrm{V_f} \leq V_{cut} \\ \\ R_{high}.(C_h + C_d).\ln(\frac{V_c(0).C_h - i_3}{V_f(C_h+C_d)}) & \text{if } \mathrm{V_f} > V_{cut} \end{cases} \tag{11}$$

Equation 11 is the generalized equation which can be used for getting the exact VBUS power discharge time in the embedded USB system. Though the equation has a general form, it must be used carefully as the value of R_{eq} is not always same. As defined earlier, R_{eq} has different values depending upon whether V_f is higher or lower than V_{cut}. i_3 is the leakage current of the device. All other parameters have been defined earlier. In next section, characterization of above model has been done.

IV. Characterization of VBUS Power Discharge

Equation 11 is the generalized solution for embedded system engineer and developer for deciding the VBUS power discharge time. Now using this model and taking host and device capacitor into account, below electrical results has been concluded. As per USB specifications, when there is no data transfer for 3ms, the host goes in low power suspend state and has a current consumption limit of 500uA and 2.5mA ECN.

Further in this section V-T graphs have been generated using MATLAB. In each of these graphs, we keep varying one parameter while others are kept constant. Let's first generate VBUS power discharge time graph varying host cap using equation 11.

1) Characterization with Respect to Host Capacitance:

When device parameters are constant and host capacitance is varied from 100uF to 200uF (Fig. 4). The corresponding time taken for 99.5 percent discharge is shown in Table-I.

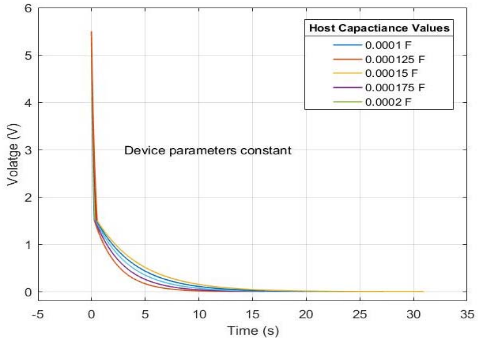

Fig. 4. VBUS Discharge Time Against Different Host capacitor at Fixed Device Parameters

TABLE I
VBUS Power Discharge Time Against Different Host Cap Value (Analytical Model Data)

Host Cap (uF)	Device Cap (uF)	VBUS Discharge Time (s)
100	8	5.608
125	8	6.91
150	8	8.213
175	8	9.515
200	8	10.82

2) Characterization with Respect to Device Capacitance:

When host parameters are constant and device capacitance is varied from 8uF to 100uF (Fig. 5). The corresponding time taken for 99.5 percent discharge is shown in Table-II.

All above graphs show that variation in discharge times can be seen by varying just one parameter.

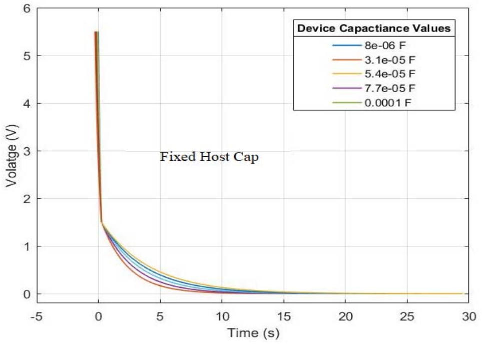

Fig. 5. VBUS Discharge Time Against Different Device Capacitor at Fixed Host Parameters

TABLE II
VBUS Power Discharge Time Against Different Device Cap Value (Analytical Model Data)

Device Cap (uF)	Host Cap (uF)	VBUS Discharge Time (s)
8	100	5.608
31	100	6.79
54	100	7.97
77	100	9.148
100	100	10.33

V. Simulation of VBUS Power Discharge Circuit

In this section, VBUS power discharge circuit is simulated using Keysight's Advance Design System (ADS) interface (EDA tool) [11] and the simulated results are compared with above analytical model results. The equivalent circuit is shown in Fig. 6 below.

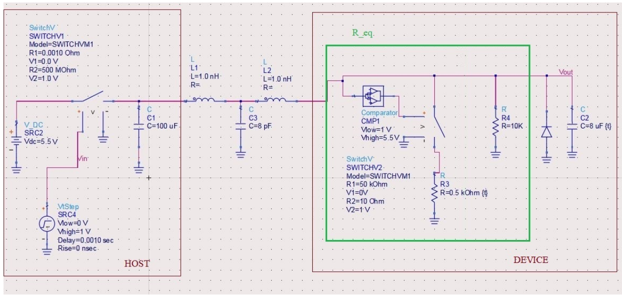

Fig. 6. Equivalent VBUS Power Discharge Time Circuit

The circuit shown in Fig. 6 is time domain equivalent of the s-domain circuit that we analyzed before. We simulate this circuit to find the correctness of our analytical model and finally, to co-relate the results with results obtained on silicon. The circuit has two portions: A host side and a device side, along with a filter connecting them. The left-hand side has a switch which first charges the capacitor C_h, and then opens such that the discharge path is now through the device (as the actual scenario). The right-hand side models the device. The comparator is crucial for choosing one of the two resistance values as per our two-resistance theory. The SWITCHV2 is used as a variable resistance here which controls the net resistance of the device, depending upon the comparator's output. If the discharge voltage from host capacitance is above

V_{cut}, it chooses low resistance, otherwise the resistance is high. We call the final resistance as R_{eq}. A diode is used in reverse bias to take into account leakage current of the device. Capacitor C_d is the net device capacitance.

Like the analytical model, we tabulate the simulation results of the above circuit. In Table-III the host capacitance is varied from 100uF to 200uF (keeping device capacitance 8uF) and corresponding time for 99.5% discharge is shown.

Similarly, in table-IV, device capacitance is varied from 8uF to 100 uF (keeping host capacitance 100uF) and corresponding time to 99.5% discharge is shown. We also show in Fig. 7, a plot obtained from simulation for C_h = 100uF and C_d = 8uF.

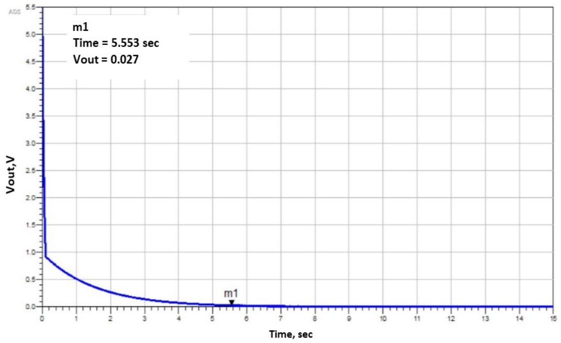

Fig. 7. VBUS Discharge Time for C_h = 100uF and C_d = 8uF

TABLE III
VBUS Power Discharge Time Against Different Host Cap Value (Simulation Data)

Host Cap (uF)	Device Cap (uF)	VBUS Discharge Time (s)
100	8	5.553
125	8	6.915
150	8	8.142
175	8	9.505
200	8	10.79

TABLE IV
VBUS Power Discharge Time Against Different Device Cap Value (Simulation Data)

Device Cap (uF)	Host Cap (uF)	VBUS Discharge Time (s)
8	100	5.553
31	100	6.715
54	100	7.98
77	100	9.155
100	100	10.35

It can be seen that an excellent correlation can be found between simulation and analytical model results. The tables I, II, III and IV shows that the time for 99.5% discharge under similar conditions is almost equal.

VI. Model Performance Results Vs Silicon Results

In this section we take four USB-IF certified devices and make real time embedded host/device [12] setup (Fig. 8) in lab for verifying above model performance results.

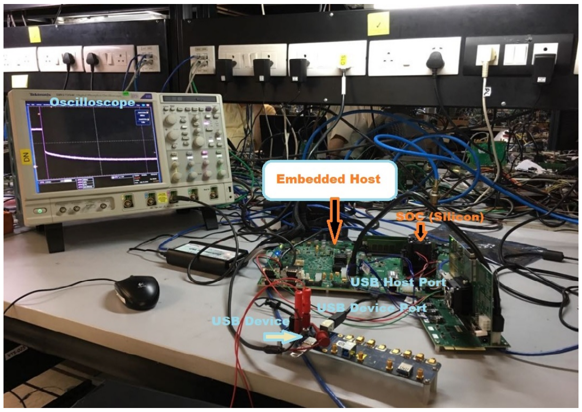

Fig. 8. Real Embedded USB Host/Device setup with oscilloscope for capturing VBUS power discharge

TABLE V
EXPERIMENTS CARRIED OUT AT EMBEDDED SETUP USING DIFFERENT VENDOR DEVICES

Device	Vendor Name	Power Discharge Time at Silicon (Seconds)	Results from Silicon
A	Kingston 32 - GB	6.00	Figure 9
B	Sony (16GB)	5.10	Figure -10
C	Strontium – 16GB	7.61	Figure - 11
D	Kingston -16 GB	5.28	Figure -12

In this setup, host having 16nm SoC and four different vendor devices used. Table V provides four devices with vendor name and respective discharge time of host against these devices.

So, experiments on real time embedded USB system in lab with four devices for VBUS power discharge time are varying from 5 second to 8 second, which is under the same range of results observed during modeling and simulation of VBUS power discharge circuit. These silicon results can be seen for different devices in Fig. 9 to 12.

VII. Conclusion

Interoperability issue with embedded USB host devices frequently occur during power cycle. Because there is no clear specification about this factor, developers and designers observe that the implementation by different vendors may show variations.

In this paper generalized analytical estimation for embedded USB power discharge mechanism is given through dual resistance approximation. The electrical characteristics of VBUS

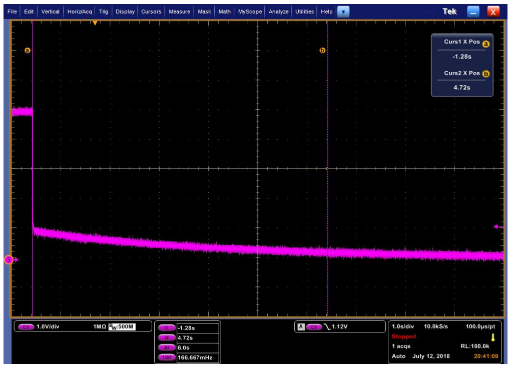

Fig. 9. Host VBUS Power Discharge with USBIF Certified Device A

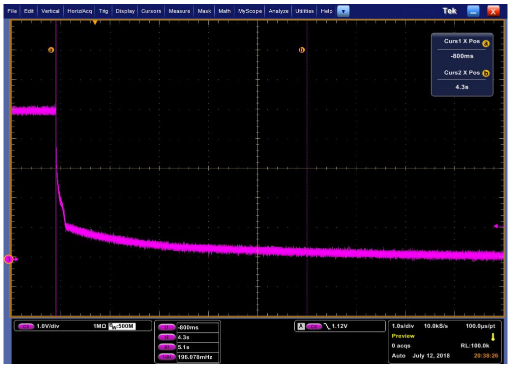

Fig. 10. Host VBUS Power Discharge with USBIF Certified Device B

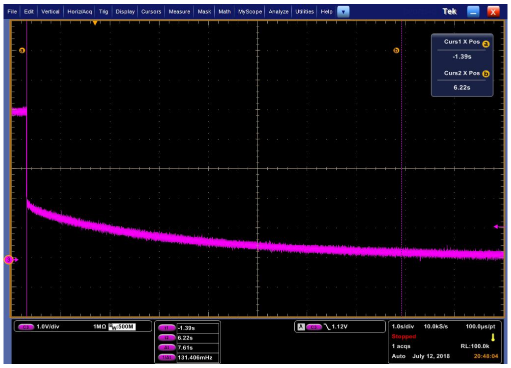

Fig. 11. Host VBUS Power Discharge with USBIF Certified Device C

power have been presented to determine discharge time for embedded USB system. A simulation of the equivalent circuit is done and results are shown to be similar to that of analytical model. Using this modeling technique, developers can ensure that their design will work under all variation in device parameters. All the characteristics of this model match with results from real 16nm silicon using as USB 2.0/3.x host and four USB IF certified devices (as embedded system). Further, to ensure that device vendors target uniform requirements,they should use this analytical estimation as the basis of the USB standard.

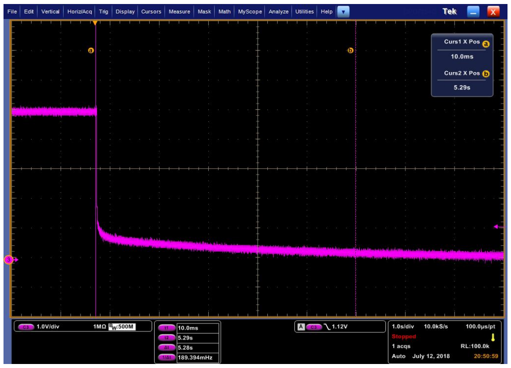

Fig. 12. Host VBUS Power Discharge with USBIF Certified Device D

Acknowledgment

The authors would like to acknowledge and thank Divya Tripathi, Ravi Dixit, Raman Brar, Design Engineers at NXP Semiconductor, Noida; for their help, comments and support.

References

[1] "An introduction to universal serial bus 2.0," Sep 2011. [Online]. Available: http://www.cypress.com/file/134171/

[2] U. I. Forum, "Universal serial bus 3.0 specification," online, 2011. [Online]. Available: www.usb.org

[3] Hewlett-Packard, "Universal serial bus 3.1 specification," 2013.

[4] H. Baig, M. A. Alam, and J. Lee, "Integrated LTSSM (link training & status state machine) and MAC layer of USB 3.0 device for reliable SuperSpeed data transactions," *Research Notes in Information Science*, vol. 9, no. 0, pp. 37–47, may 2012.

[5] M. K. Pandey, S. Shekhar, J. Singh, G. K. Agarwal, and N. Saxena, "A novel approach for usb2. 0 validation on system on chip," in *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*. IEEE, 2013, pp. 1–4.

[6] L. Pierce and S. Tragoudas, "Enhanced secure architecture for joint action test group systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 7, pp. 1342–1345, July 2013.

[7] S. Shekhar, M. K. Pandey, G. K. Agarwal, and N. Saxena, "Post silicon debugging approach for usb2. 0: Case study of enumeration," in *Computational Intelligence and Communication Networks (CICN), 2013 5th International Conference on*. IEEE, 2013, pp. 418–422.

[8] R. Sharma and P. K. Sharma, "Resolve embedded usb interoperability challenges." EDN, Jan. 2016. [Online]. Available: https://www.edn.com/Pdf/ViewPdf?contentItemId=4441184

[9] J. Axelson, *USB complete: the developer's guide*. Lakeview research LLC, 2015.

[10] R. T. N. V. S. Chappa, B. R. Jammu, M. Adimulam, and M. Ayi, "Vlsi implementation of ltssm," in *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, vol. 1, April 2017, pp. 129–134.

[11] S. S. Inc., "Keysight ads interface," vol. 16, May 2018.

[12] I. Sobański and W. Sakowski, "Hardware/software co-design in usb 3.0 mass storage application," in *ICSES 2010 International Conference on Signals and Electronic Circuits*, Sept 2010, pp. 343–346.