Delay Lock Loop Design Project

Author
Aaron Brown
Boise State University
ECE 418
May 12th, 2013

Abstract

The goal of the project was to design a DLL, delay lock loop, that could but used in a DDR3 SDRAM interface. The clock controlling the data timing for the memory module needs to pass through a clock tree buffer, and the output needs to align with the input. In order to realign the data a DLL with replica buffer components can be used to make the input clock's rising edge lineup with the output clock's rising edge.

1. Introduction

A DLL is a negative feedback network that uses three blocks, a (PD) phase detector, charge pump, and voltage controlled delay line (VCDL). There is also a loop filter involved with the DLL that consists of just one capacitor. The input clock frequency that was used in this project was 200 MHz, which translates to clock periods of 5ns. The supply voltage for the DLL was set to 1.8v using an 180nm CMOS processes. An overview of the DLL can be seen in Figure 1.1.

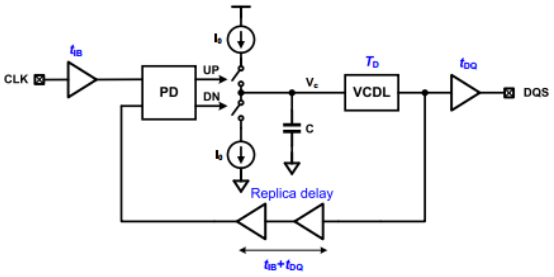

Figure 1.1: DLL used in DDR3 SDRAM interface

2. Layout and Design for the Components

One of the most fundamental and important components of a DLL system is the VCDL. The VCDL controls whether the output has the capability to lock or not. The delay ranges for a VCDL should go from .5Tc to 1.5Tc, where Tc is the clock period. If this range is not satisfied then the DLL may not be able to lock. The range where the VCDL locks is right at a delay of one clock period. Figure 2.1 provides a visual representation of the VCDL's delay ranges.

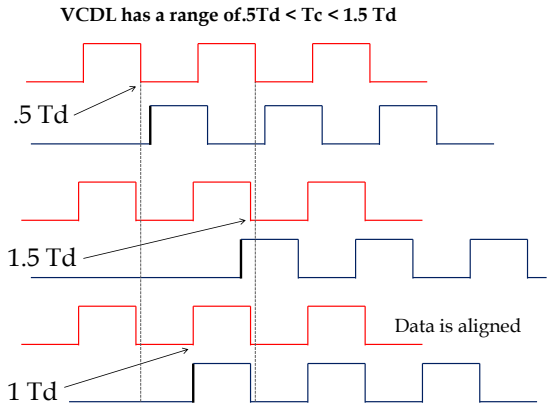

Figure 2.1: VCDL delay range

The modified current starved model was used to design the VCDL. At first a differential stage VCDL was attempted, where Vc could vary the line resistances. The issue with the differential VCDL however was that it had a very poor range where Vc had little to no effect on the system. After trying different adjustments to the differential VCDL to try to get it to have a better delay range a simpler signal ended stage was considered. Figure 2.2 shows the circuit schematic of the VCDL in Cadence.

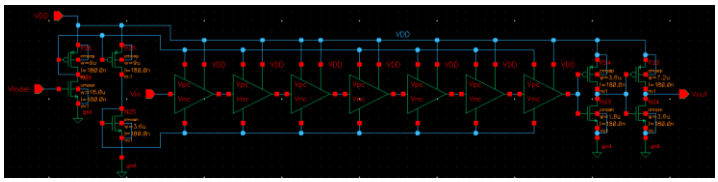

Figure 2.2: VCDL Cadence circuit layout

The voltage control would first enter a biasing stage and set the current. Each of the delay cells would then also be set to the same current by using current mirrors. The result is as the voltage control (Vc) increases, more current enters the system and the delay propagation decreases. The layout for each of the delay blocks is shown in Figure 2.3. Each delay block is simply an inverter, where voltage controls the current that drives the inverter.

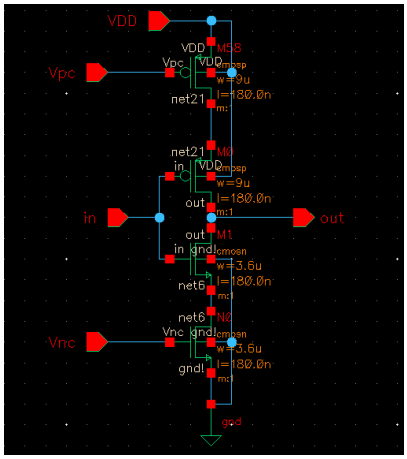

Figure 2.3: layout of a delay block

This particular VCDL used a total of 7 delay blocks and two output buffers. The result is the output waveform is inverted. By inverting the waveform, additional delay is actually achieved by .5Tc, making it possible for this VCDL to be in the range of .5Td < Tc < 1.5Td. An important characterization of the VCDL is the KVCDL value which is a part of the open loop gain. In order to calculate KVCDL a plot of control voltage vs. delay was done using Cadence simulations. The results for the VCDL voltage vs. delay curve can be seen in Figure 2.4 along with a linear regression of the data.

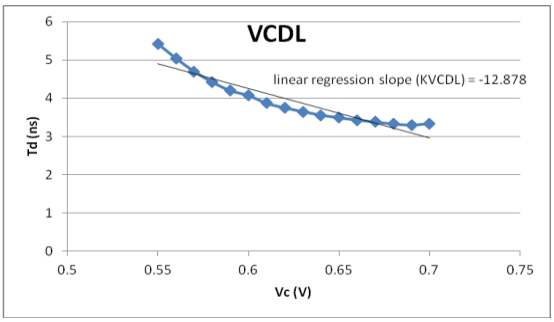

Figure 2.4: VCDL voltage vs. delay curve

Figure 2.4 was generated in excel using data points obtained in Cadence. Using excel a linear regression was performed on the data showing a slope of -12.878 ns/V.

The KVCDL value can be calculated from the slope by multiplying by 2π/Tref. KVCDL then is calculated to be 16.18 rad/(sec*V) or 2.56 Hz/V. The open loop gain of this DLL can is shown in Equation 2.1.

$$Equation\ 2.1: L(s) = \frac{I_0 K_{VCDL}}{2\pi cs}$$

Since the open loop gain only has one pole the system is always stable no matter what the value parameters. The one concern is however to make the DLL appear to be a continuous time control system. In order to do that the Wu,loop, unity gain of the open loop, needs to be much smaller than the Wref, the reference clock frequency in rad/sec. To make the Wu,loop small in comparison to Wref the appropriate size for the capacitor on the loop filter needs to be chosen. This was done mathematically where Wref = 200Mhz and Wu,loop was chosen to be ten times smaller than Wref.

$$Wu, loop = \frac{Wref}{10}$$
$$Wu, loop = 2\pi * 20Mhz$$
$$1 = \frac{I_0 2\pi K_{VCDL}}{2\pi cs}$$
$$c = \frac{I_0 K_{VCDL}}{20Mhz}$$
$$c = 6.44pF, I_0 = 50u,\ K_{VCDL} = 2.56$$

The value for the capacitor then should be 6.44pF to have Wu,loop ten times smaller than Wref. If the capacitor becomes any smaller than the system will start to become discrete and periodic disturbance will be observed when locking. Or if the capacitor becomes much larger the system will take drastically longer to load, so for a good equilibrium the value of 6.44pF for the capacitor was used in the DLL.

The other two main components of the DLL were the PD and charge pump. The reason why a PD was used instead of a PFD, phase frequency detector, is when a PFD is used it's possible for the VCDL to become stuck at its maximum and minimum delays. The circuit schematic for the PD in Cadence is shown in Figure 2.5.

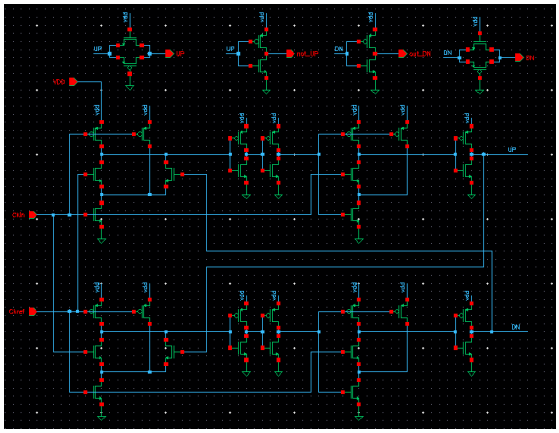

Figure 2.5: PD circuit schematic

The PD circuit outputs the UP output when the input Ckin is leading the Ckref and the DN output is applied in the reverse situation. The PD is essentially made from a simplified dual DFF without the frequency content. The PD output four signals which are UP, DN and there inverses.

The PD is driving the charge pump which is shown in Figure 2.6. When the Up bar is mostly high the charge pump sends current to Vc, making the control voltage rise. If the DN bar is mostly high then current is pulled from Vc, lowering it. In the situation were UP and DN are both mostly low, current passing through without disturbing Vc, allowing it to stay constant. Transisotrs on the charge pump make up the control switches and the current source drives a replica biasing stage using current mirror logic. The biasing stage then goes to the nmos and pmos current sources on the charge pump. Lastly a operational amplifier is used to regulate the second branch node equal to Vc for proper biasing.

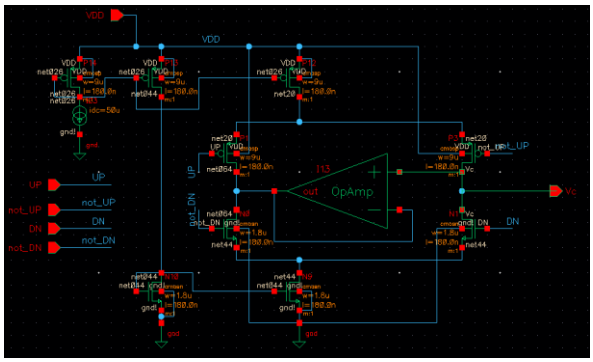

Figure 2.6: Charge pump circuit schematic

3. Overall DLL Design and Results

The overall layout for the DLL is shown in Figure 3.1.

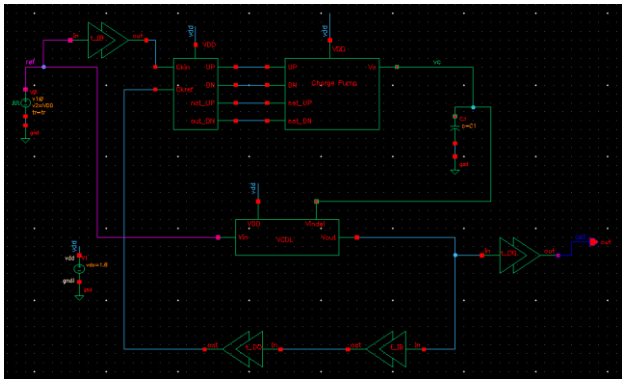

Figure 3.1: DLL schematic layout

The overall system is connected so that if the feedback signal is lagging the input clock the PD component tells the charge pump to increase current to Vc. The increase in the Vc signal makes the current decrease until the output signal and input signal are aligned. This may seem counterintuitive to have delay decreasing when the input leads the output but remember that the VCDL has a negative voltage to delay relationship. This simply means the VCDL is inverting the output so the reference position of timing changes. Notice also the T_IB and T_DQ delay lines. These delay lines represent the buffer tree that the input clock has to pass through. After going through the buffer tree the input clock becomes delayed and no longer has proper alignment, this is where the DLL is used. By making it so the feedback also has the same delay lines as the input and output T_IB and T_DQ lines the DLL is able to produce an output clock that is aligned with the original input clock. Figure 3.2 shows how locking is successfully achieved when Vc starts at its minimum value and Figure 3.3 shows how locking is achieved when Vc is at its maximum value.

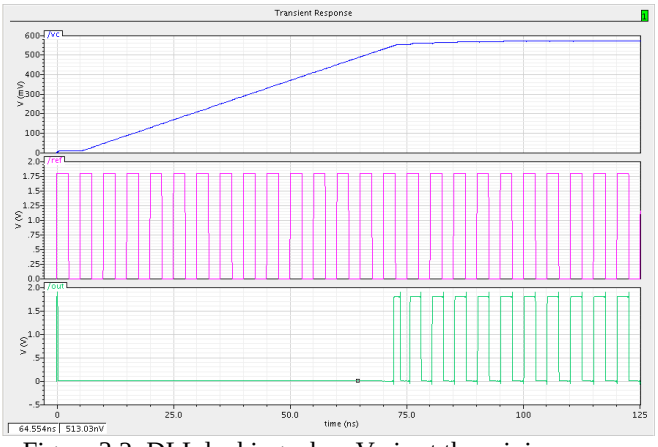

Figure 3.2: DLL locking when Vc is at the minimum

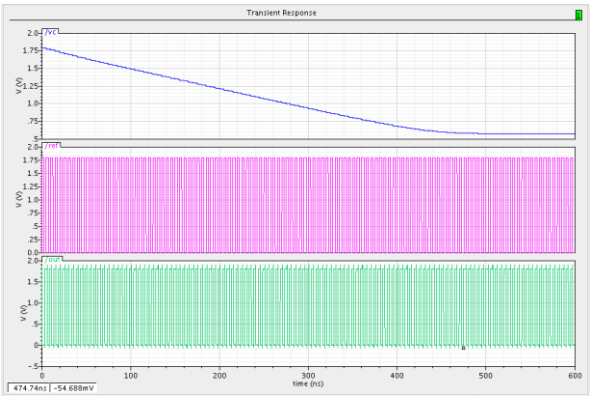

Figure 3.2: DLL locking when Vc is at the maximum

When Vc started at both the maxium and minium intial conditions it was able to lock to a constant Vc of about 571mV. The locking value of Vc depends on the amount of delay for both T_IB and T_DQ, if either is adjusted then the DLL will converge to the value of Vc that allows the input to be aligned with the output signal. For further observation let's make sure that the input clock and output clock really are aligned in this particular case where Vc is converging to 571mV. Figure 3.3 shows both the input clock and the output signal for comparison in locking conditions.

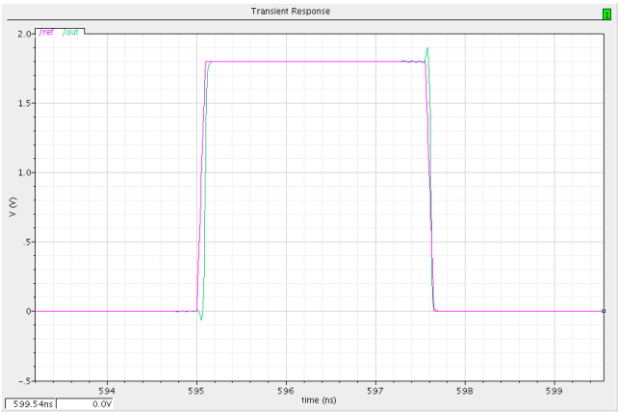

Figure 3.3: input clock superimposed onto the output

As Figure 3.3 shows the input clock is indeed aligned with the output with both rising and falling edges lining up. One small observation to make is there are some slight transient signals at the rising and falling edges. These small voltage spikes however are very small in height, about 50mV and are so brief they should cause any issues in the DDR3 SDRAM interface when used for timing. The transient signals are generated from the inherent nature of the VCDL when propagating the original waveform through the delay blocks.

4.1 DLL Jitter Characterization

As discussed in Section 2, if a capacitor is chosen that is much smaller than 4.66pF then Wu,loop will also increase. The result of when Wu,loop is comparable to Wref the system no longer be can be approximated as being continuous. The DLL system in its self is not a continuous system sense current is being pushed and pulled in incremental cycles to the loop filter. Jitter becomes very distinguishable when the capacitor value is small while in Figures 3.1 and 3.2 hardly any disturbance can be seen when convergence is reached. Figure 4.1 shows the performance of Vc when capacitor is only 500fF. The kind of jitter that can be observed here is periodic deterministic jitter. The current pushing and pulling from the charge pump is becoming very apparent because the capacitor is small enough where it can be very quickly filled and depleted.

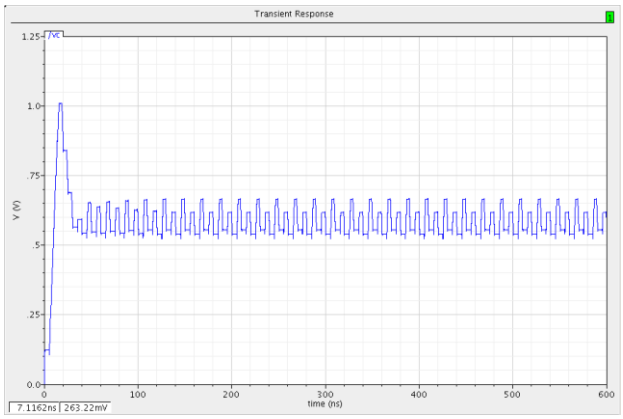

Figure 4.1: Deterministic jitter caused from small cap

In order to avoid deterministic jitter such as that seen in Figure 4.1 the Wu,loop should appear much smaller than Wref, and so a capacitor much smaller than 6.44pF should not be used.

4.2 Power Consumption

The power consumption of the overall DLL and individual blocks was measured by observing each current being pulled from the power supply. The average power was then calculated as VDD multiplied by the average current. Figure 4.2 show the power distribution between the PD, charge pump, and the VCDL. The VCDL generated the most power at 700uW, second was the charge pump at 279uW, the PD consumed the minimum amount of power at 78.8uW. The entire system consumed about 1.058mW which is boarder line decent.

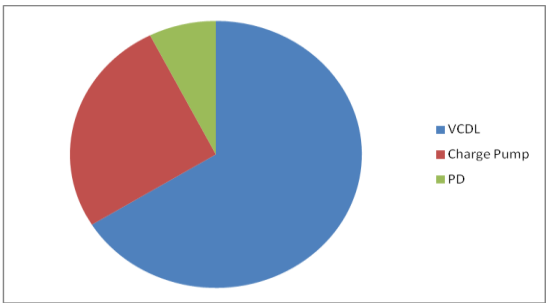

Figure 4.2: Power consumption pie chart

4.3 PSRR Characterization

Power supply rejection ratio was tested on the overall DLL system using Cadence AC analysis on the voltage supply. The results can be seen in Figure 4.3, where at lower frequencies the PSRR is much larger than at higher frequencies. This is to be expected because at higher frequencies the parasitic capacitances of the transistors start to leak more voltage disturbance.

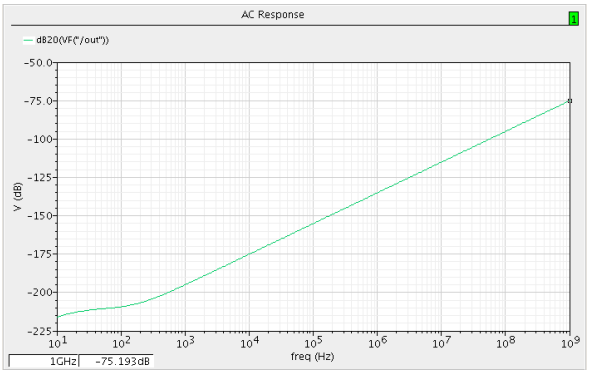

Figure 4.3: PSRR vs. frequency

5. Conclusion

This project successfully showed the process from design and characterization to full transistor level DLL creation. The DLL had 3 main components including the PD, charge pump, VCDL, and Loop Filter. The DLL was able to realign the output clock to the input regardless of initial Vc conditions, and T_IB, and T_DQ delay lines. By calculating the KVCDL value for the open loop gain, it was shown that the capacitor couldn't be much smaller than 6.44pF or else a large amount of deterministic jitter would appear. Power consumption measurements showed that the VCDL consumed most of the total average power of the system following the Charge Pump and the PD. Total power consumed was 1.058mW. Table 5.1 summarizes the parameter values used for open loop transfer function.

C1	6.44pF
Io	50uA
VCDL	2.56 Hz/v
Wref	200MHz
Wu,loop	20MHz

Table 5.1: Parameter Values

Future work in this project could include redesigning the VCDL using a Maneatis Cell in order to obtain higher PSRR for the system. Also noise analysis and more in depth jitter analysis could be done on each of the three individual function blocks.

4.5 Acknowledgments

Acknowledgments go to Dr. Vishal Saxena and his Memory Architecture and Clock Synchronization course at Boise State University.

4.6 References

[1] Dr. Vishal Saxena. Delay Locked Loops [Online]. Available:http://lumerink.com/courses/ECE518/Handouts/DLLs.pdf

[2] Dr. Vishal Saxena. Project 2 Instructions [Online]. Available:http://lumerink.com/courses/ECE518/s13/Homeworks/PLL_Project_2_2013.pdf

[3] R. Jacob Baker. CMOS Circuit Design, Layout, and Simulation."Practical VCO and VCDL Design". pp596-599.