

**Design of a 4-bit Flash ADC for use in a $\Delta - \Sigma$
modulator**

A Project Report

submitted by

BARADWAJ.V

*in partial fulfilment of the requirements
for the award of the degree of*

BACHELOR OF TECHNOLOGY



**DEPARTMENT OF Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS.**

August 3

THESIS CERTIFICATE

This is to certify that the thesis titled **Design of a 4-bit Flash ADC for use in a $\Delta - \Sigma$ modulator** , submitted by **V. Baradwaj**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Shanthi Pavan

Project Advisor

Assistant Professor

Dept. of Electrical Engineering

IIT-Madras, 600 036

Place: Chennai

Date: 3rd August 2008

ACKNOWLEDGEMENTS

Though it may sound cliched, I first wish to thank my project advisor Prof. Shanthi Pavan, not only for guiding me through the project, but also for his excellent courses, the constant support and advice he has given me for almost my entire undergraduate studies. I wish to state that had it not been for his excellent teaching and his approachability, it is extremely unlikely that I would have continued in Electrical Engineering. I also wish to thank Prof. Nagendra for his course on VLSI Broadband circuits which helped me unify my view of analog and digital circuit design. I wish to thank Prof. Karmalkar for his course on Device modelling without which I cannot imagine designing anything. I also thank my friends Hariprasath and Manohar for the thought provoking discussions and the memorable time I had with them. I also wish to thank everyone else who has made my stay in the lab a comfortable, enjoyable and a memorable experience.

Abbreviations

DSM	$\Delta - \Sigma$ Modulator
OBG	Out of Band Gain
SNR	Signal to Noise Ratio
FFT	Fast Fourier Transform
DAC	Digital to Analog Converter
NTF	Noise Transfer Function
V_{ppd}	Volts, Peak to Peak Differential
ADC	Analog to Digital Converter

ABSTRACT

KEYWORDS: Flash ; Oversampled Data converters; High speed.

A Flash ADC has been designed in $0.18\ \mu\text{m}$ UMC CMOS technology. It has been designed to operate at a clocking rate of 1 GHz embedded in a $\Delta - \Sigma$ modulator loop with a bandwidth of 20 MHz. It consumes a total 11.31 mW (2.75 mW in comparator array + 4.45 mW in the bias and clock generator and a 4.11 mW of power in the thermometer to binary converter and deserialiser). It is expected to deliver a 15-bit performance in the DSM. It has a clock to output delay of 400-500 ps depending upon the input samples. It occupies an area of $470\ \mu\text{m} \times 220\ \mu\text{m}$ along with its bias generator and clock generator.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
ABSTRACT	iii
LIST OF FIGURES	1
1 Introduction	2
2 Existing Flash Architecture	3
2.1 Comparator Design	4
2.2 Drawbacks of this design	7
3 Design of a 1GHz flash ADC	11
3.1 The Current mode logic latch	11
3.2 Designing the Comparator topology	12
3.2.1 Reset and active inductive load	12
3.2.2 Latch 1	15
3.2.3 Latch 2 and CML buffer	16
3.2.4 Input-Reference subtraction	18
3.3 Clocking	20
3.4 Comparator array and Reference ladder	22
4 Bias and Clock generation	25
4.1 Bias generation	25
4.2 Clock generation	30
4.2.1 250 MHz Clocks	30
4.2.2 Latch Clocks	31
4.3 CML level generation	37
4.4 Flash Reference Level generation	39

5	Binary Code generation and Deserialisation	41
5.1	System level choice of thermometer to binary conversion logic	41
5.2	Implementation	43
5.2.1	Conversion of CML data streams to CMOS data streams	43
5.2.2	Bubble correction	44
5.2.3	Transition detect and Conversion to binary	45
5.3	Deserialisation	46
6	Layout and Simulation results	50
7	Conclusions	56
7.1	Things to be completed	56
8	Appendix A	58
9	Appendix B	61

LIST OF FIGURES

2.1	Block diagram of comparator array	4
2.2	Schematic of the latch used in (1)	5
2.3	Switched capacitor subtraction of input and references	6
2.4	Timing diagram of the clocks used in the comparator shown in figure 2.3	7
3.1	A typical CML latch	12
3.2	Nmos transistor based active inductive load	13
3.3	Approximate Bode plot shapes for the output impedance, load capacitance and the active inductor realized	14
3.4	Inductive peaking load using a explicit g_m to act as a gyrator	14
3.5	Topology of the CML latch which resolves the difference between input and reference	16
3.6	Topology of the CML latch which stores the output of the first latch (which is shown in 3.5)	17
3.7	Topology of the CML which cleans up the output waveform of the second latch (shown in figure 3.6)	18
3.8	Input-reference subtraction scheme used in this design	19
3.9	Block diagram of the comparator used in this design	21
3.10	Timing diagram of the clocks used for the latches	22
3.11	A closer look at the sample comparator output waveform before layout	23
3.12	A sample comparator output waveform before layout	24
4.1	Clock common mode generation	26
4.2	Common mode generation for the input of the first latch	26
4.3	The usually used cascode voltage generation scheme	27
4.4	The cascode generation scheme used in this work	28
4.5	The bias generating section for the comparator's cascode voltages	28
4.6	The bias generating section for the comparator	29

4.7	The negative feedback based bias generation to fix the output swing of the first latch	30
4.8	The negative feedback based bias generation to fix the output swing of the second latch	31
4.9	The negative feedback based bias generation to fix the output swing of the comparator	32
4.10	The C^2MOS based D flip flop used for frequency division	33
4.11	The inverter chain driver for the CMOS clocks for the input -reference subtraction stage of the comparator	34
4.12	The CML clock driver which drives the comparator array taking in the CMOS equivalent of the clocks	35
4.13	The CML clock driver which drives the comparator array taking in the CMOS equivalent of the clocks	36
4.14	The logic which generates the LRST clock to be given to the comparator array and the CMOS clocks for the CML to CML converter	38
4.15	Circuit which generates a $\frac{V_{ref}}{R}$ reference current to produce voltage levels for the CML clocks	39
4.16	Circuit which generates the CML clock levels from the reference current generated using the circuit shown in figure 4.15	40
4.17	Circuit which generates the flash reference levels from the reference current generated using the circuit shown in figure 4.15	40
5.1	Effect of random offsets in the flash's comparators on the output SNR of $\Delta - \Sigma$ loop	42
5.2	Circuit which converts the input CML streams to CMOS streams	44
5.3	3-bit majority coding based bubble correction logic	45
5.4	Transition detect and pre-charge logic	47
5.5	The divide by 2 clock divider used in generating the sampling clocks for the deserialiser	48
5.6	The divide by 4 clock divider realized using the divide by 2 clock divider shown in figure 5.5	48
5.7	The C^2MOS sampling stages to sample off the 1 GHz streams to produce the 250 MHz streams to the output	49
6.1	Clocks generated by the laid out clock generator driving the input-reference subtraction stage of the laid out flash	50
6.2	Clocks generated by the laid out clock generator driving the latches in the laid out flash	51

6.3	A closer look at the output of comparator and ideal DAC cell	52
6.4	Spectrum at the output of a single point sampling DAC in the loop	52
6.5	Waveforms at the output of the comparator and the ideal DAC cell	53
6.6	Layout of the flash with its clock generator	54
6.7	Layout of the thermometer to binary converter and the de-serialiser	55

CHAPTER 1

Introduction

Oversampled data converters are systems which are often preferred over nyquist rate converters for the advantages like greater tolerance to component variations, mismatch which oversampling offers. It is possible to use a poor quantizer to achieve very good performance by making use of noise shaping. However, the very fact that the input is oversampled poses an upper limit on the maximum input frequency these converters can handle. In applications like disk drive read channels, where a high input bandwidth range is desired, it becomes a challenging task to use oversampled converters.

From a designer perspective, designing open loop data converters is easier at high speeds than closed loop data converters like $\Delta-\Sigma$ modulators. This is because as the frequency increases, the percentage delay one can tolerate in the feedback loop comes down. It is therefore necessary to design the loop's components with as little propagation delay as possible. At the same time, the designer can exploit the noise shaping characteristic to compromise on specifications which could have been objectionable in a open loop design.

The following is an attempt at designing a Flash ADC for use in a continuous time $\Delta-\Sigma$ modulator operating at a clock frequency of 1 GHz. The performance desired of the DSM is 15-bit performance with a overall power consumption limit of 25 mW for the entire modulator. The full scale of the modulator and hence the flash is $3V_{ppd}$. The bandwidth is 20 MHz and is clocked at 1 GHz giving a oversampling ratio of 25. Since all the three components - the loop filter, the flash ADC and the current steering DAC will consume a lot of power due to the high frequency specifications, a rough estimate of the power limit for the flash ADC is around 8mW and the current design attempts to meet the same. This flash design has a delay of around 400-500 ps depending on the input it samples and consumes 11.31 mW of power.

CHAPTER 2

Existing Flash Architecture

This chapter as its title indicates involves a in depth analysis of a flash architecture used in two of the previous designs at IIT Madras ((1) and (2)). It also discusses it's pros and cons and it's applicability to high speed designs.

Flash ADCs as their name indicates, constitute the class of the fastest analog to digital converters. The logic behind the conversion goes as follows. The full scale range of the ADC is divided into some number of levels (say $2^n - 1$) and the input is compared with each of these reference levels. The comparison is done by using $2^n - 1$ comparators (one for each level) and is done in parallel. As a result the delay through the comparators is the least in this architecture. The result of the comparison (say 1 if input is higher than the reference and 0 if it is lesser than the reference) is a $2^n - 1$ bit code and is always of the form of a string of 1's followed by a string of 0's (in increasing order of reference). Such a code is called a thermometer code and is evidently a inefficient way of using the available number of bits. This thermometer code is later converted to binary code as per the need of the external circuitry. The drawback of this architecture, is evidently the exponential increase in area per increase in bit and hence is typically limited to use in 6 bit conversions (6 bit represents $2^6 - 1 = 63$ levels).

The references for the comparators are generated using a resistive ladder running between the fullscale voltages. Since the implementation is differential, a pair of differential references are required for each of the comparators. This is achieved by running two resistive ladders in parallel in an intertwined fashion and feeding the corresponding node voltages to each of the comparators. A block diagram of the same is shown in figure 2.1.

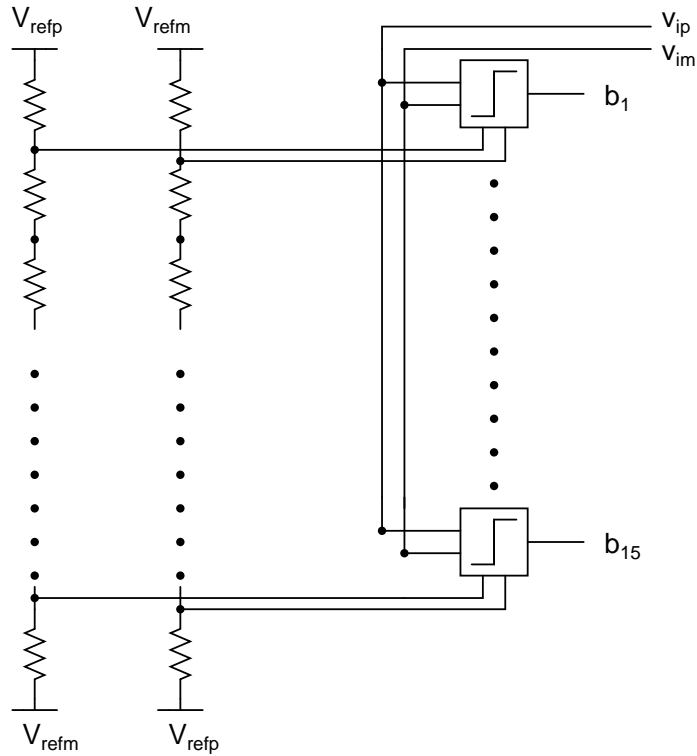


Figure 2.1: Block diagram of comparator array

2.1 Comparator Design

This section discusses the implementation of the comparator shown in the figure 2.1. The comparator is expected to resolve the input in to either of the supply voltages depending on whether it is greater than or lesser than the reference voltage. One way of achieving this is to make the input go through a large gain system whose outputs become saturated if the input is above/below the reference. The drawback of this scheme is that the sensitivity of the comparator is fixed. An alternate solution would be to use positive feedback. The difference between the input and the reference is fed to the comparator and is regenerated using positive feedback. The advantage of this method of resolving is that the *gain* of the comparator increases with time. This also gives us the advantage of *synchronizing* all output transitions to some clock, easing further processing of the resolved data.

In the design in (1), a latch accomplishes the above mentioned task of regenerating the difference between the differential input and the differential references. At low speeds, it is economical in terms of power to use rail to rail outputs and

hence a CMOS latch is used. The circuit diagram of the latch is shown in figure 2.2.

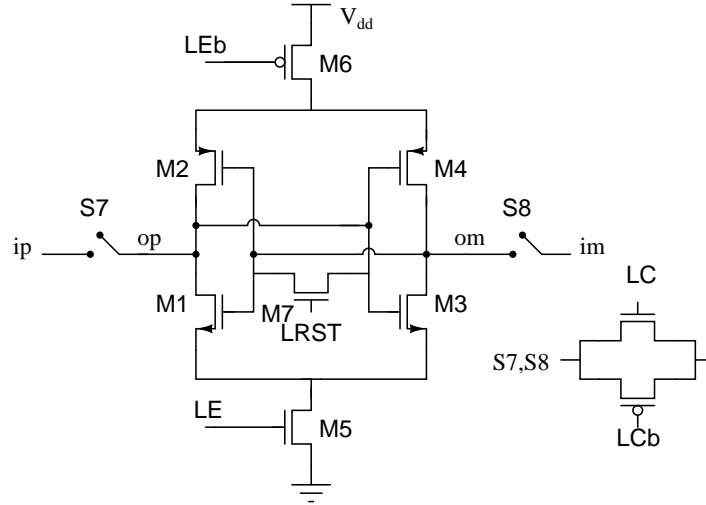


Figure 2.2: Schematic of the latch used in (1)

The latch shown in figure 2.2 consists of two inverters connected back to back (M1-M2, M3-M4) with two switches enabling them (M5,M6). When a differential voltage v_i is applied at the inputs of the two inverters and LE is switched on, the inverters are enabled and the latch regenerates the two nodes to the supply rails. It is to be noted that the input and the output of the latch are the same two nodes if the switches S7,S8 do not exist. It is essential that these two nodes be isolated from the circuit which feeds the latch with the input to avoid a situation where the two circuits are trying to control the nodes. The switches S7,S8 accomplish this by turning off when the latch is regenerating. The latch output is then sampled off and is reset after sampling is complete to remove any hysteresis.

The input to the latch comes from a stage which produces difference between the input and the reference. Since the fullscale is large ($3V_{ppd}$), the only scheme which is viable to do this is a switched capacitor subtraction stage without any transistors operating in saturation. The scheme used in the design in discussion is shown in figure 2.3.

The timing diagram of this comparator's clocks is shown in figure 2.4. The basic operation of the comparator is as follows. During the phase $LE \cap LEa$, the coupling capacitors (marked C in figure 2.3) are charged to $v_{refp} - v_{cm}$ and

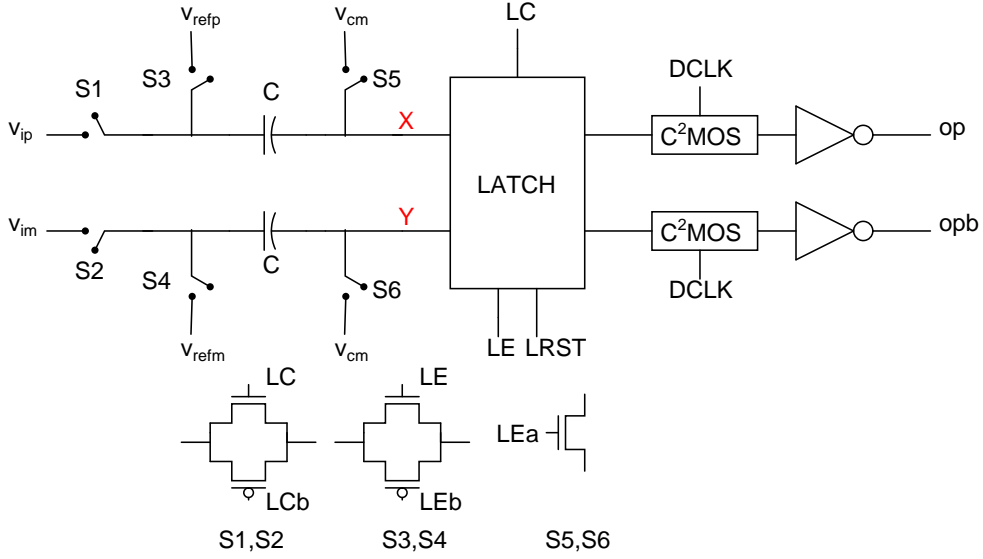


Figure 2.3: Switched capacitor subtraction of input and references

$v_{refm} - v_{cm}$. The clock LEa which is an advanced version of LE turns off first thereby dumping some charge on to the nodes X and Y. This charge would depend on the value of v_{cm} . This results in a corresponding amount of charge leaving the other plate of the capacitor in to the reference ladder. Effectively, the capacitor is charged to $v_{refp} - v_{cm} + \frac{q(v_{cm})}{C}$ where $q(v_{cm})$ is the charge dumped as a function of v_{cm} . The other differential half is charged to $v_{refm} - v_{cm} + \frac{q(v_{cm})}{C}$ and hence the charge dumped effectively is canceled if we look at the differential input to the latch. On the other hand, if $S3$ and $S4$ were turned off first, this cancellation would not have happened. This technique, called bottom plate sampling, is often used in switched capacitor designs to avoid input dependent charge dumping. During the LE phase, the latch regenerates and is isolated from this switched capacitor stage by the switches shown in figure 2.2 controlled by LC. Towards the end of LE, the output is available and is sampled off using the C^2MOS inverters shown in figure 2.3. Once the output is stored, the latch is reset to remove any hysteresis.

In the next phase LC, the input is connected to one end of the capacitor and the nodes X and Y move along with the input at the voltages $v_{ip} - v_{refp} + v_{cm} + \frac{q(v_{cm})}{C}$ and $v_{im} - v_{refm} + v_{cm} + \frac{q(v_{cm})}{C}$. The input is sampled at the instant when LC turns off and is regenerated during the LE phase which follows.

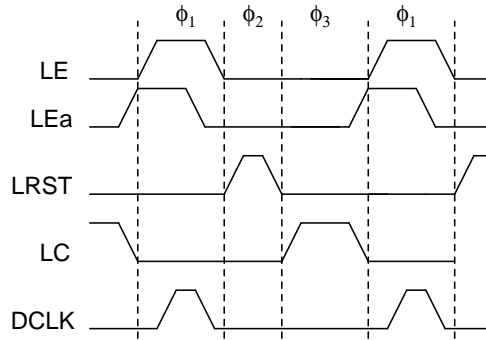


Figure 2.4: Timing diagram of the clocks used in the comparator shown in figure 2.3

2.2 Drawbacks of this design

The flash architecture discussed in the previous section has been designed and tested in two previous designs (1), (2). The first of these works at 3.072 MHz while the second design operates it at 300 MHz with a few modification to the sizes of the transistors used. In this section, we discuss the extendability of this architecture to higher speeds, it's pros and cons to come up with a new architecture if necessary.

The first road block we hit when we try to increase the frequency of operation is the current drawn by the latch when we turn it on or off. Evidently, as we increase the frequency of operation, the reactive currents increase due to a reduction in the capacitive impedances. This would imply increased drops across the switches thereby resulting in subtraction errors. It was found that around 500 MHz, these drops were of the order of 25 mV differential as compared to a differential LSB of 187.5 mV . Tracing the currents drawn by the comparator showed that most of the current flowed into the latch indicating the source of trouble. There are four issues with the operation of the latch described in the previous section.

Shared input and output of the latch

The first problem with the latch is that the drains and the gates of all transistors are connected to the inputs of the latch. At low frequencies, this is not a source of trouble because the transistors M0 and M5 in the latch would turn the latch

off by disconnecting it from the supplies and thus stop any flow of current into any node. However, at high frequencies, the finite amount of time taken by the sources of the transistors M1-4 to track their drains and the parasitic capacitances at the drains of M0 and M5 would result in current flow through the transistors. This clearly shows up in the case of large inputs to the latch. This is because, when the inputs to the latch are large, the transistors of the inverters which were in cutoff (either due to the low V_{gs} or V_{ds}) would begin to turn on. At low frequencies, they would have been immediately turned off by the sources of M1-4 tracking their drains. However, due to finite delay between these processes, current is either drawn/pushed on to the subtraction circuitry's coupling capacitor. This would effectively mean a current flow through the switches just after turn on of the latch connect switches (S1,2,7,8). Fortunately however, this occurs only when the inputs to the latch are large enough to turn the transistors on and since the inputs to the latch are large, we can tolerate greater effective error in subtraction as long as the regeneration is flawless.

Resetting the input nodes of the latch

The second and the most important problem is with the very architecture of the latch. In a CMOS latch, the inputs and the output of the latch are the same. The latch handles "continuous waveforms" i.e, we do not handle discrete levels as is usually the case. This would mean that resetting the latch is absolutely necessary to avoid hysteresis at the output of the latch. If this is not done, the quantization error of the quantizer would not be white and can cause trouble. Resetting the latch is thus absolutely necessary. However, by the very architecture of the latch, resetting the output of the latch is the same as resetting the input of the latch. Now, if the last discussed problem were not there, the current drawn is utilized only to charge the input node of the latch. In this scheme of operation, the input node of the latch just before turn on of switches S1,2,7,8 is at the common mode voltage (The resetting operation brings the input \equiv output of the latch to the common mode). Just after turn on of the switches, the node should be at the input voltage, $v_i(n) - v_{ref}$ where $v_i(n)$ and v_{ref} form the input and reference

voltages in the current clock cycle in either of the differential paths. This would mean that a charge proportional to $v_i(n) - v_{ref}$ has to flow through the switches just after turn on. On the other hand, if we had not reset the latch input, the input node of the latch would have stayed at $v_i(n-1) - v_{ref}$ where $v_i(n-1)$ is the input voltage at the end of the previous clock cycle's latch connect phase (at the time LC is just about to turn off). This would mean a charge proportional to $(v_i(n) - v_{ref}) - (v_i(n-1) - v_{ref}) = \delta v_i$ which will generally be small in comparison to $v_i(n) - v_{ref}$ given that the flash is operating in a delta sigma modulator where the input is over sampled.

Clocking

The third issue is also specific to the architecture of the subtraction circuitry. The subtraction circuitry is a switched capacitor type subtraction circuitry and is inherently limited by the multiple clocks needed in a single clock cycle. It can be seen from figure 2.4 that the clocks LC, LE, LEa, LRST should be non-overlapping to ensure correct operation. In other words, LC should be off by the time LEa goes on and LC should go on only after LE goes off. Similarly, LRST should go on only after LE goes off (to prevent the inverters to get biased at the meta stable point and draw a huge current from the supply) and LC should go on only after LRST goes off to prevent shorting the inputs through the switches and the coupling capacitor (S1 - S7 - 50 fF capacitor - RST switch - 50 fF capacitor - S8 - S2). If we assume that we are able to increase the resetting switch's size to be able to reset the latch in 100 ps, using a modest estimate of the rise/fall times in a 0.18 μm technology of 100 ps, we spend 6 rise/fall times + reset time = 700 ps in these phases. In other words, if we are operating at a 500 MHz clocking speed, out of a 2 ns time period, we are left with merely 650 ps for each - LC and LE to be on. At 1 GHz operation, this is still worse, with only 150 ps for each phase.

Common mode noise

The fourth issue is common to all high speed implementations - It is always better to go for a differential implementation to avoid power supply noise, crosstalk and other common mode noises. This implementation is however pseudo-differential because of the latch.

From the above discussion, it is evident that the comparator topology as it is cannot be extended to a high frequency design and is in serious need of modifications if it is to work at a higher frequency. We observe the following requirements for a new comparator topology if it is to have the pros of this topology. The first is the low dynamic offset of the comparator. The output node of the latch starts regenerating from a voltage whose common mode has been designed to be v_{cm} . As a result, this value can be adjusted to be the trip voltage of the inverters used in the latch to reduce dynamic offset. The switched capacitor subtraction stage of this topology is an inevitable obstacle in achieving high speeds due to the large swing the comparator is expected to handle ($3V_{ppd}$). The property which tops the list of things to be avoided is the shared input and output port of the latch. As discussed already, resetting the input of the latch is a bad idea both in terms of the drops across the switches and the power consumed in the ladder. The next chapter discusses the design of a comparator which would be implemented to work at a clocking speed of 1 GHz.

CHAPTER 3

Design of a 1GHz flash ADC

As noted in the previous chapter, a truly differential topology with isolated input and output for the latch would be the ideal candidate for a high speed design. One such latch is the current mode logic latch and the current chapter discusses the design of a 1 GHz flash ADC employing such a comparator in this work. Throughout this chapter, we assume that the flash is clocked at 1GHz. The detailed clocking scheme is discussed towards the end of this chapter.

3.1 The Current mode logic latch

The simplest form of a current mode logic latch is shown in the figure 3.1. This is constructed from a current mode logic (CML) inverter/buffer which is a simple differential pair. The CML inverter has two outputs one being the exact complement of the other. As a result, to construct a latch we need only one CML inverter and connect its outputs back to the inputs in positive feedback. To separate the input and output of the latch and to supply the input of the latch to the regenerating section, we make use of another CML inverter whose outputs are connected to the regeneratively connected CML inverter. This topology right away assures us an isolated input and output node for the latch.

During the phase when the latch is expected to track/amplify the input, the normal CML inverter branch is enabled by enabling Lb . When it is expected to regenerate, the regeneratively connected CML inverter is enabled by enabling L . If we assume infinite output impedance for all transistors (as they are maintained in saturation), the input common mode does not affect the output common mode for the normal CML inverter block. Therefore the output common mode is $V_{dd} - \frac{IR}{2}$. Now, when the regenerating section is enabled, the initial common mode is such

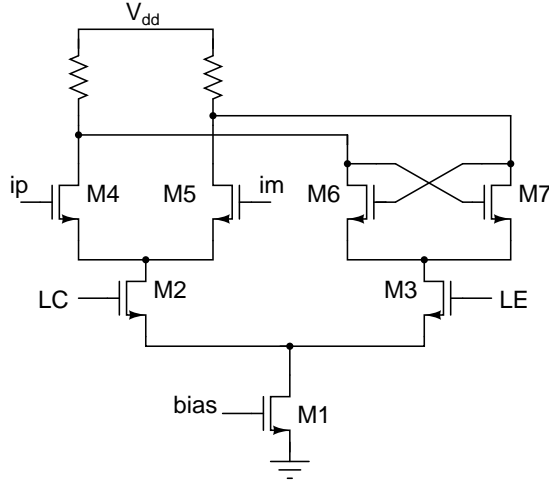


Figure 3.1: A typical CML latch

that, when it is applied to the regenerating inverter's gates, the output common mode produced is again $V_{dd} - \frac{IR}{2}$. Therefore, the output common mode does not tend to change when regenerating is enabled. In other words, the dynamic offset of the latch is greatly reduced.

3.2 Designing the Comparator topology

3.2.1 Reset and active inductive load

As discussed in the previous chapter, resetting the latch is necessary to reduce hysteresis and the time spent by the tracking CML inverter section in bringing the output to desired level from the previously (half/fully) regenerated value. Since the output common mode level close to the V_{dd} rail in a CML latch, the reset switch to be used is a pmos transistor. Using a resistor for the load of the CML would drastically increase the area of the latch and hence we stick to active loads. Using a pmos transistor biased in triode region is one option. However, using a pmos transistor at the output node would add more capacitance to the already loaded output stage. In order to offset these load effects, we choose to use active inductive peaking to enhance the speed of switching.

One candidate for an active load is an active inductor built using a nmos

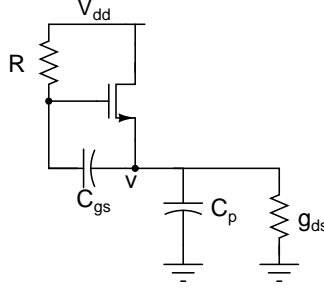


Figure 3.2: Nmos transistor based active inductive load

transistor as the gyrator element. The circuit is shown in the figure 3.2. From the figure, it can be seen that the gate-source voltage of the nmos transistor is a *lossy* integrated version of the source voltage. In other words, the current drawn by the nmos transistor is a lossy integrated version of the source voltage implying a inductive load with a series resistance. Clearly the inductance is given by $\frac{RC_{gs}}{g_m}$ and the series resistance is $\frac{1}{g_m}$. The effective admittance seen is therefore

$$Y = g_{ds} + sC_p + \frac{g_m}{RC_{gs}s + 1} = \frac{g_m + (g_{ds} + sC_p)(RC_{gs}s + 1)}{RC_{gs}s + 1}$$

The natural frequency of the pole is therefore $\omega_n = \sqrt{\frac{g_m + g_{ds}}{RC_p C_{gs}}}$ and a quality factor $Q = \frac{\sqrt{RC_p C_{gs}(g_m + g_{ds})}}{g_{ds} RC_{gs} + C_p} \approx \sqrt{\frac{RC_{gs}(g_m + g_{ds})}{C_p}}$. The approximate bode plots (to depict the shapes) have been plotted in figure 3.3. From the figure, it is evident that unless the DC value of the realized inductance is higher than the value of g_{ds} at the output, inductive peaking cannot be achieved. This yields us a lower limit on the value of g_m that can be used for the nmos transistor. However, given the fact that we would be using this as a load section for a CML latch, the *resistance* shown by the load should be such that regeneration is possible. In other words, $g_{mdiff} > g_m$ where g_{mdiff} represents the regenerating pair's g_m . Moreover, using the nmos transistor which is providing the DC resistance itself as a gyrator gives us lesser degree of freedom in changing the parameters to achieve the required bandwidth and quality factor. Since the common mode of the output of a CML latch will be close to the upper supply rail, to maintain the nmos transistor in saturation, the gate of the nmos should be connected to a voltage level above V_{dd} through the resistor. This however should not be troublesome because no DC

current will be drawn from this supply.

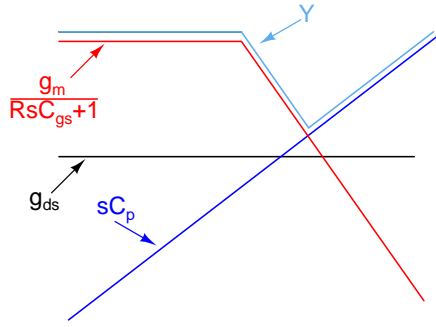


Figure 3.3: Approximate Bode plot shapes for the output impedance, load capacitance and the active inductor realized

The second alternative for inductive peaking is to use a pmos transistor instead of an nmos transistor. This clearly has the advantage of operating with in the usual supply. To provide the gyrator effect, we need to integrate (at least lossily) the output node voltage and feed it to the pmos' gate terminal. This can be done by recognizing the fact that the C_{gs} of the pmos goes to the supply(incremental ground) and all we need is a resistor between the gate and the drain of the pmos. Since this method still uses an implicit g_m for the gyrator, like it's predecessor, it has very little degree of freedom to tweak the parameters. Recognizing this fact, the solution then would be to introduce an extra g_m in the feedback path providing an alternative way to increase the g_m . Since a transistor itself can be used as a g_m and the pmos gate is generally below it's drain (to maintain it in triode), a nmos transistor is the solution. The resultant circuit is shown in figure 3.4.

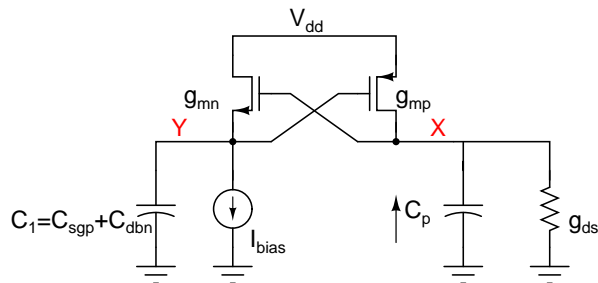


Figure 3.4: Inductive peaking load using an explicit g_m to act as a gyrator

If we neglect the output impedance of the source I_{bias} , a voltage v at the node X would result in the gate of the pmos at $\frac{g_{MN}}{sC_1}$ and thus would produce an

inductance of $\frac{C_1}{g_{mn}g_{mp}}$. The output impedance of the source I_{bias} and that of the nmos transistor itself would however result in a DC impedance of $g_{mp}\frac{g_{mn}}{g_{mn}+g_{dsn}}$. We have therefore achieved our goal of separating the DC impedance and the active inductance we realize. The DC impedance can be controlled by tweaking g_{mp} and can be maintained low enough to enable regeneration in the latch and at the same time maintain a high enough inductance to achieve the desired peaking by adjusting g_{mn} .

One method to control the output swing of the latch would be to vary the current I_{peak} . However, if one employs a feedback loop to control the swing achieved using this load, the quality factor achieved using inductive peaking becomes uncontrolled. This was a source of trouble across process corners in the current design. We therefore resort to partial inductive peaking, in the sense that we use a PMOS transistor as a load in parallel to the inductive load and control the PMOS transistor's resistance using feedback to control the swing achieved using the load. The quality factor achieved by inductive peaking is *nominally* fixed by providing a constant process independent current (a fraction of the latch current) for I_{peak} . This mitigates the variation of the quality factor realized across process corners to a large extent because all the g_m s scale across process corners and thus the quality factor remains almost the same across process corners.

3.2.2 Latch 1

We thus arrive at the topology of the CML latch which *resolves* the difference between the input and reference as shown in figure 3.5. It is to be however, noted that the above analysis holds good only for small swings and the differential pair in it's switching operation is highly non-linear. The above conclusions on the peaking characteristics are therefore valid only for the load which is about to carry the tail current. The output single ended voltages of the latch will therefore show *peaking* only while coming down to $V_{dd} - I_{bias}R_{equiv}$ and do not show *any peaking* characteristics when headed towards V_{dd} . As a result, to see a moderate amount of peaking in the output differential voltage (which is of more concern to us than the

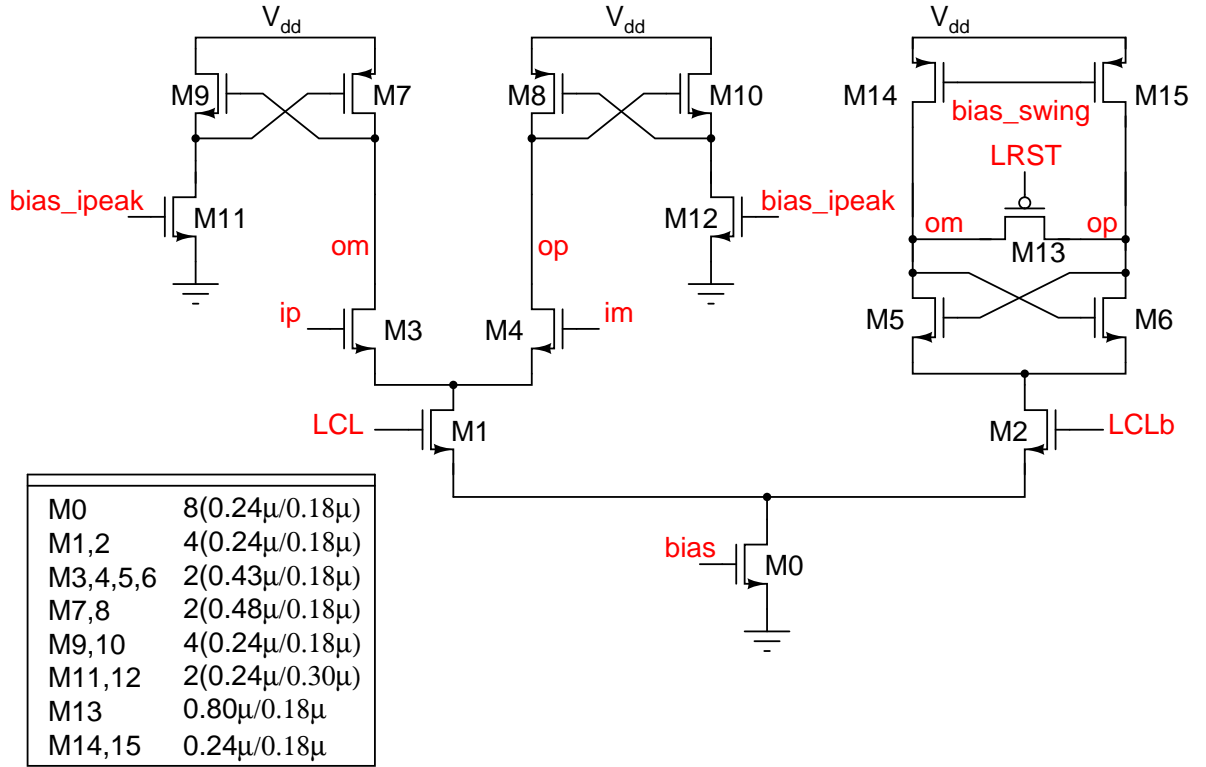


Figure 3.5: Topology of the CML latch which resolves the difference between input and reference

single ended waveforms), the waveform while going down should have much more peaking (to offset the slow rise in voltage of the other arm). Moreover, in this case, since the time available for regeneration is barely 500 ps and the parasitic capacitance at the output being large, even with inductive peaking in place, the differential output voltage would practically seem like it is ramping. However, using inductive peaking enhances the tracking bandwidth thereby improving the overall performance.

This latch needs to be driven by a pair of complementary clocks and the time available for each phase - tracking and regeneration is close to 500 ps (with rise/fall times, it is close to 400 ps). Resetting the latch is therefore done towards the end of the regeneration phase of the latch (LCLb is on).

3.2.3 Latch 2 and CML buffer

Since the output of the latch described in the previous subsection is reset every clock cycle, it needs to be stored using another latch. We accomplish this by

following the first latch with another latch. This latch provides us with some more gain and holds the output for one full clock cycle. In order to save time however, this latch tracks the input when the first latch is regenerating. In other words, we do not wait for the first latch's output to regenerate to enable the second latch. The schematic of the second latch is shown in figure 3.6. It was found that

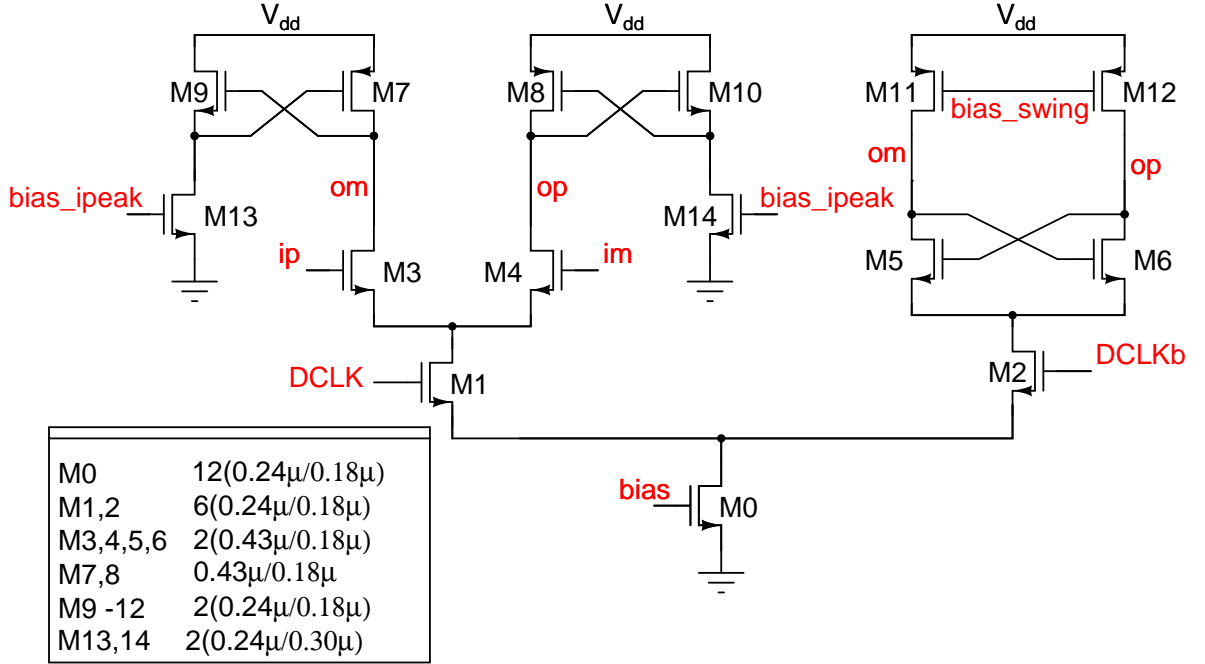


Figure 3.6: Topology of the CML latch which stores the output of the first latch (which is shown in 3.5)

the second latch was also not always able to resolve the input voltages and hence another CML buffer is added at the output to make sure the output waveform is large enough to be able to switch the DAC. Moreover, due to the finite output impedance of the current sources in either of the latches, the tail node parasitic capacitance along with the dip in the output voltage when the current switches from one arm to the other (tracking arm to regeneration arm or vice versa) make the output voltage have a lot of ripple even in the switched state. Having the buffer at the output cleans up this to a certain extent. Adding the buffer at the output was however a trade off between obtaining a cleanly switched waveform at the DAC output and delay in the flash. It was however found that the delay in the flash with CML buffer was any where between 300 ps to 450 ps depending upon the difference between the input and the reference voltage. The schematic of the CML buffer used at the second latch's output is shown in figure 3.7. It can be

seen from the figure that a cascode is provided for the current source of the CML buffer since there is greater head room available in this case with the absence of the clock's switching pair.

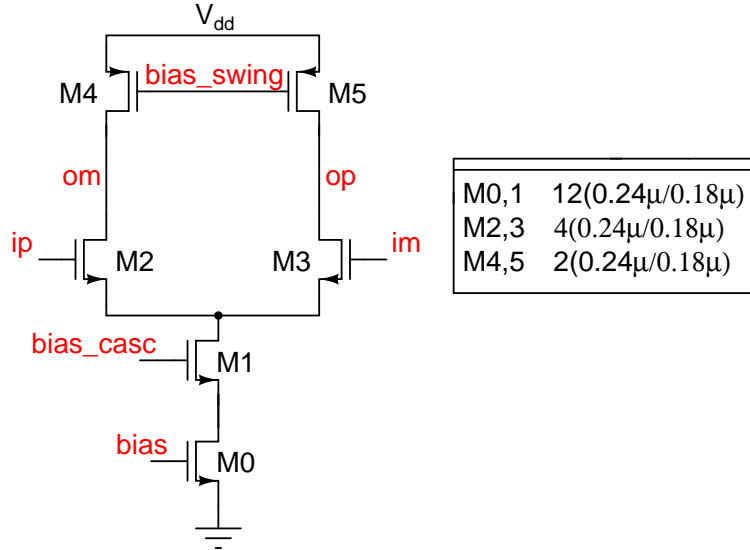


Figure 3.7: Topology of the CML which cleans up the output waveform of the second latch (shown in figure 3.6)

3.2.4 Input-Reference subtraction

As discussed in chapter 2, the currents drawn by the parasitic capacitances at the input of the latch draw currents causing a relatively large drop across the switches in the signal path (S1,2,7,8 in figure 2.3). The obvious solution is to eliminate the switches in the signal path. In the topology shown in figure 2.3, the switches were necessary because, when the coupling capacitor is being charged to the reference voltage (during the phase $LE \cap LEa$), if the input is not isolated from the capacitor, we create a short between the input and the reference voltage. One way to circumvent this problem is to charge the coupling capacitor *differentially* and not in a *single ended fashion*. In such a case, the individual nodes can still maintain the absolute voltages but the difference in the voltage across the coupling capacitor's terminals is fixed. However, since we only have *single ended references* i.e., v_{refp} , v_{refm} , v_{cm} , we need to convert these to $v_{refp} - v_{cm}$ and $v_{refm} - v_{cm}$. This can be done by charging another capacitor's terminals to the *absolute* voltages, say,

v_{refp} and v_{cm} and connecting it in parallel to the coupling capacitor in the signal path periodically to charge the coupling capacitor. This illustrates the functioning of the circuit shown in figure 3.8 which is used as the input-reference subtraction stage. The switches S1-4 and S5-8 shown in the figure 3.8 are controlled by non-

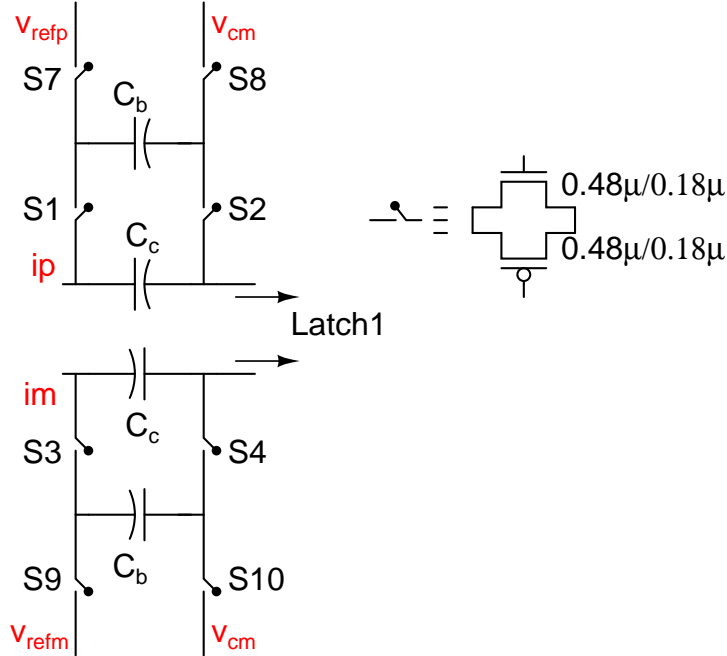


Figure 3.8: Input-reference subtraction scheme used in this design

overlapping clocks(say LE, LC respectively). If this stage is clocked at the same frequency as the latch, it would be preferable to use the tracking phase of the first latch (3.5) to be the same as the LC phase of the subtraction stage to ensure that the references are transferred on to the coupling capacitor when the first latch is not tracking.

Since the charge on the coupling capacitor C_c keeps accumulating because of the charge transferred by C_b , the eventual charge on the capacitor C_c would be $C_c(v_{refp} - v_{cm})$. Since the charge only keeps accumulating, it can be concluded by inspection that the only effect of switch resistance (in absence of other non-idealities) would be an increase in the time required to reach the *eventual* value. If the switches were ideal but there was parasitic capacitance at each node of this circuit, one can by inspection, conclude that the output voltage is a scaled version (only in amplitude) of $v_{ip} - v_{refp}$ or $v_{im} - v_{refm}$ around the common mode voltage v_{cm} . This is also benign because the latch would still detect the sign of it's input

correctly. However, when both the non-idealities are present, it is equivalent to filling a leaky tank (C_c) with a leaky bucket (C_b , leak due to RC time constant). In other words, we can expect to have an input dependent reference across the coupling capacitor C_c .

Rephrasing this, we can conclude that the output of the subtraction stage will be the input minus a shifted version of the scaled input. This can again be modelled as a kind of an offset in the comparator. It was however found that the switch resistance has to be drastically reduced to around ($1\text{ k}\Omega$ or smaller) to achieve an offset less than a tenth of an LSB. The solution to this problem lies in recognizing the fact that this subtraction stage, in practice can be clocked asynchronously with respect to the rest of the circuit. There would be no difference in performance in the ideal cases but it gives us the freedom to clock this stage even at a different frequency. This feature is exploited in this design and the input-reference subtraction section is clocked at $\frac{f_s}{4} = 250\text{ MHz}$ instead of $f_s = 1\text{ GHz}$. This improves the performance considerably because there is 4 times more time for the rise/fall time crippled waveforms at each node to reach their settling values (Since the rise/fall times affect the node voltages exponentially, we get a substantial improvement).

In a conventional flash however, this could be a problem, because, the output of the flash might show a tone at $\frac{f_s}{4}$. However, in this specific application of the flash, where we use it in a $\Delta - \Sigma$ modulator loop, this tone falls out of the signal band and can be modelled as noise injected along with the quantization noise. This tone is therefore attenuated by the NTF and is filtered out by the decimation filter. In a stand alone flash however, this could be a serious problem for it reduces the SFDR.

3.3 Clocking

The complete comparator schematic is shown in figure 3.9. It can be seen that the following clocks are used - LC, LE, LCL, LCLb, D_CLK, D_CLKb, LRST. The available time period of $\frac{1}{f_s} = 1\text{ ns}$ has to be now appropriately divided amongst

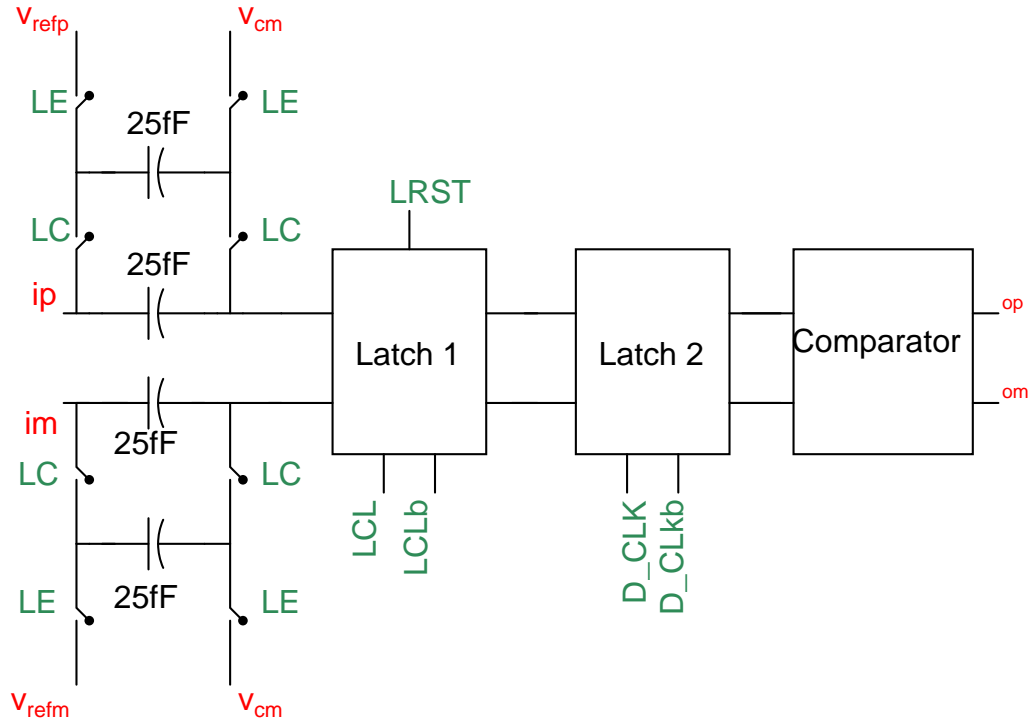


Figure 3.9: Block diagram of the comparator used in this design

these clocks. As already mentioned, LCL and LCLb are complementary and so are D_CLK and D_CLKb. The period for which LCL is on controls the time available for the output of the first latch to start tracking the input waveform. Therefore, increasing this, would to a certain extent increase the tracking ability of the latch. The tracking bandwidth which represents how fast the latch is able to respond to a change in the input sign can only be controlled by tuning the inductive peaking discussed in the subsection 3.2.1. Unlike the CMOS latch, the CML latch need not be turned off when resetting it. We therefore reset the first latch towards the end of it's regeneration phase. We therefore, choose to use 50% duty cycle clocks for LCL and LCLb while placing the reset towards the end of the regenerating cycle turning off along with the rise of LCL. Also, since the output of the first latch has to be stored on to the second latch before reset, D_CLK has to turn off before LRST goes high. In order to save time, we let the second latch to be in tracking phase when the first latch is in it's regenerating phase. This however has the drawback that the output waveform of the second latch shows a lot of ripple even in a completely switched state. The purpose of the CML buffer which follows the second latch shown in figure 3.9 is to clean up this ripple and add some gain to the signal path. As already discussed in the previous section, the subtraction

stage is asynchronous with the rest of the circuit. They are generated by using a $\frac{f_s}{4} = 250 \text{ MHz}$ clock and a non-overlapping clock generator. Details of the clock generation are discussed later. The timing diagram of the clocks is shown in figure 3.10.

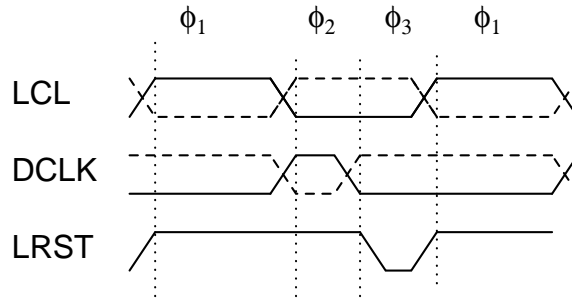


Figure 3.10: Timing diagram of the clocks used for the latches

3.4 Comparator array and Reference ladder

15 comparators for the fifteen references and 2 dummy comparators are connected to a pair of resistive ladders running between two reference voltages which fix the fullscale of the flash. The differential references for the ladder are generated by using two ladders running in parallel in opposite directions with the corresponding nodes connected to place the two ladders in parallel. MOS capacitors are placed at each reference node to minimize the ripple in the node voltages due to the currents drawn by the comparators.

The nominal values of the resistors that are to be chosen are decided by the currents drawn by the comparator. In every clock cycle, the input parasitic capacitance of the latch has to be charged to $v_i(n) - v_i(n - 1)$ (as discussed earlier). This would result in a non-zero average current drawn by the comparator. If this average current is comparable to the current running in the ladder, the ladder references are distorted. This produces third harmonic distortion in the reference voltages. This can equivalently be modelled as a *negative* distortion in the input to the comparator and hence, a third harmonic component will show up in the spectrum of the flash ADC. The CML latch based comparator by construction draws

lesser current from the inputs and the references and hence very little current has to flow in the ladder as compared to the previous 300 MHz design (2). A value of $2\text{ k}\Omega$ has been chosen for the ladder resistors (each ladder has $2\text{ k}\Omega$ resistors which form $1\text{ k}\Omega$ when the ladders are connected in parallel. The AC currents drawn by the comparator will again cause a ripple in the reference voltages. To minimize this, capacitors are placed at each reference node voltage and they provide a total capacitance of 1 pF at each reference voltage(after the two ladders are connected in parallel).

The comparator described in this chapter was simulated for a test input of $3V_{ppd}$ sinusoid at $\frac{5}{256}\text{ GHz}$ with a 6LSB peak to peak differential sinusoid (562.5 mV amplitude) riding over it modelling the quantization noise at $\frac{f_s}{2} = \frac{31}{64}\text{ GHz}$. The output waveform of the comparator, that of a DAC cell (with ideal resistive termination), the input sinusoids (differential reference to the comparator is 0) and the sampling clock (it is v(LCLb,LCL) the rising edge of which is the sampling instant) are shown in figure 3.12. A zoomed version of the same is shown in figure 3.11. It can be seen from the figure that the delay in the flash is approximately 260 ps.

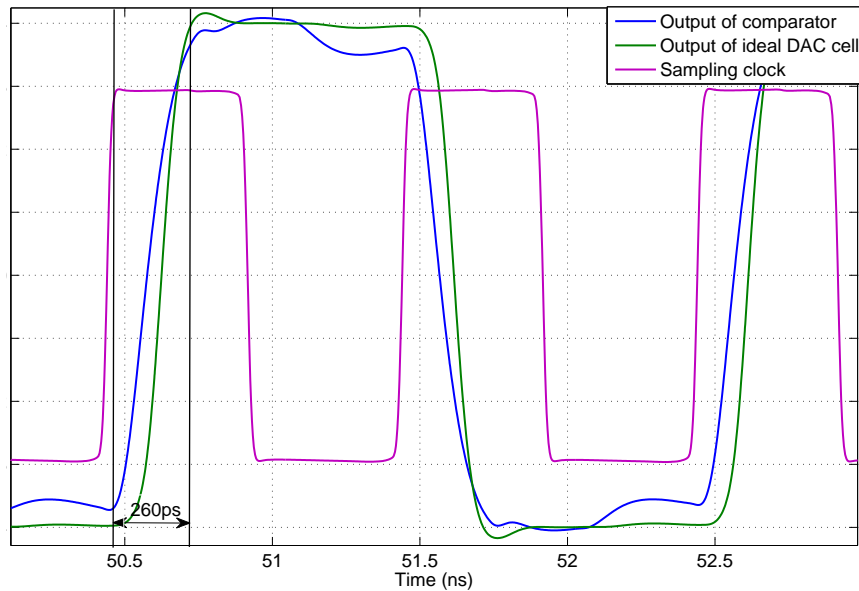


Figure 3.11: A closer look at the sample comparator output waveform before layout

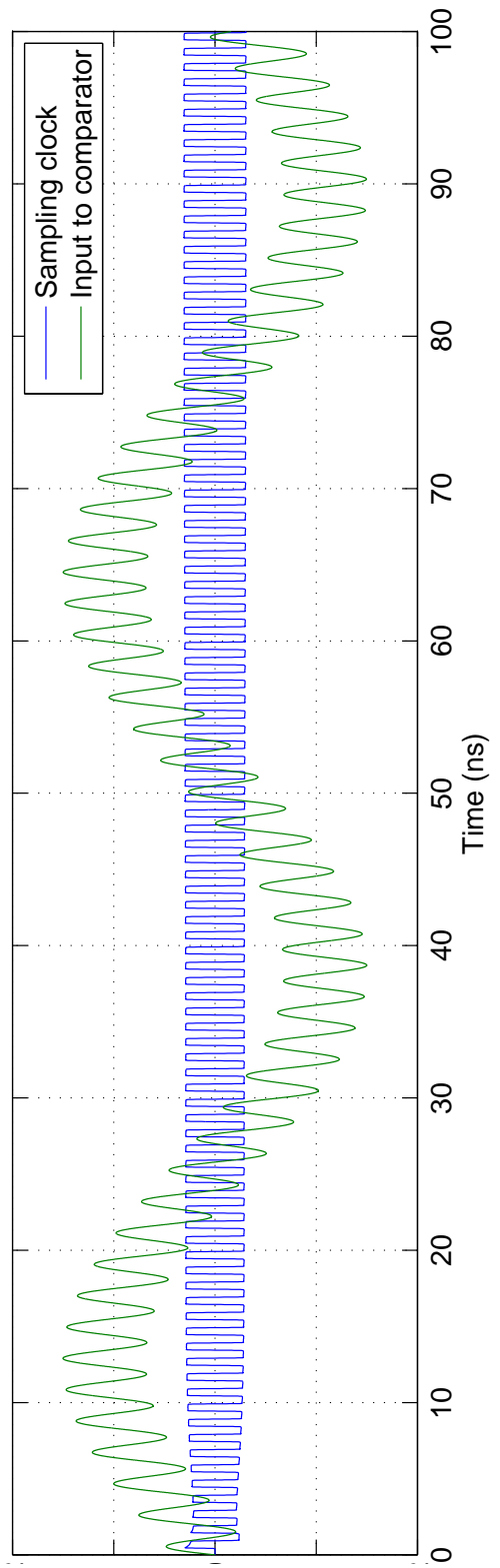
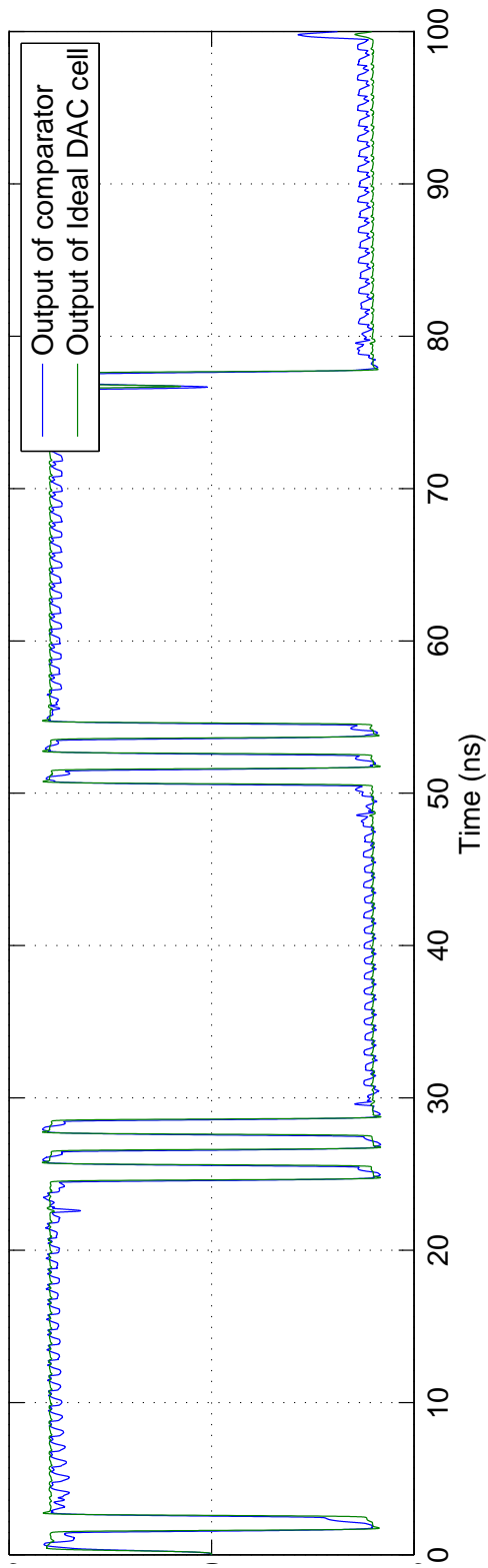


Figure 3.12: A sample comparator output waveform before layout

CHAPTER 4

Bias and Clock generation

Having discussed the design of the comparator, the next step is to design the bias circuitry required for biasing the different current sources, CML clocks and input of the first CML latch. The following chapter discusses the generation of the bias voltages and the required clocks from a master clock.

4.1 Bias generation

The following are the bias voltages to be generated - input common mode to the first latch (fed through the voltage v_{cm} in the subtraction stage), the gate voltage for the nmos to fix the current in the latches, the clock common modes (same for both the latches) and the swing control bias for the CML buffer to fix the swing across process corners and temperature variations and the cascode voltage for the current source in the CML buffer. Since a CML switch always operates in saturation, it can be regarded as cascode device for the transistors below it. In the reasoning that follows, the current which we wish each of the sections should carry is assumed to be I . The following points regarding the different bias voltages needed can be observed:

1. The bias voltage for the current source in each of the sections - latch 1 , latch 2 and the CML buffer is to be decided by the current that should flow in the sections.
2. The worst case condition for the bias current source is when it's drain voltage is at it's lowest. This would happen when the switches above the bias current source (the switching pair which takes the clocks) are at the same value (or equivalently the clocks' common mode voltage and hence carry $\frac{i}{2}$ each). The clock common mode voltage should therefore be generated such that it leaves a $v_{ds} \geq v_{dsat}$ for the current source. Since the sizes for the bias current transistor (M0, $8(\frac{0.24\mu}{0.18\mu})$) in 3.5, 3.6 is a scaled version of the current source and the clocked pair in 3.5) is twice the size of the switches and the

condition is that the two switches should carry $\frac{I}{2}$ each, the clock common mode voltage is the same as a cascode voltage for a transistor of size $8\left(\frac{0.24\mu}{0.18\mu}\right)$ with the cascode transistor also of the same size. By the way we have chosen the sizes, this is the same as the cascode voltage needed in the CML buffer for transistor M1.

3. The worst case condition for the switching pair (M1,M2 in 3.5 and 3.6) which takes the clocks is when its drain voltage is at it's lowest and it's source voltage is at it's highest. This happens when the gates of the input switching pair (M3,M4 in 3.5) are at the common mode level and the clock switching pair (M1,M2 in 3.5 and 3.6) is completely switched. This translates to a second cascode voltage generation for $8\left(\frac{0.24\mu}{0.18\mu}\right)$, $4\left(\frac{0.24\mu}{0.18\mu}\right)$, $4\left(\frac{0.43\mu}{0.18\mu}\right)$ sized transistors, stacked on top of each other (the lowest transistor's size is the first size).

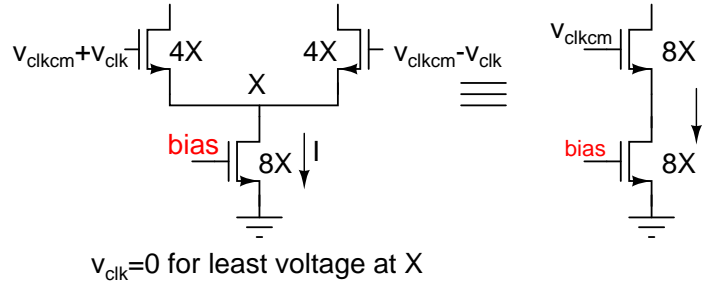


Figure 4.1: Clock common mode generation

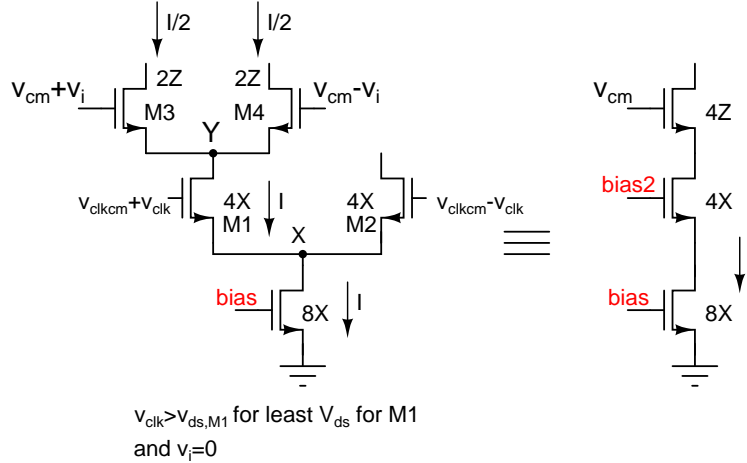


Figure 4.2: Common mode generation for the input of the first latch

The points 2 and 3 are illustrated in figures 4.1 and 4.2. We next proceed to generate the cascode voltages. The usually employed scheme for generating cascode voltages using two current sources as shown in figure 4.3. This however suffers from the drawback that it relies too much on square law characteristics of

the MOS transistor and the ratio $\frac{X}{4}$ is often smaller than $\frac{X}{6}$ and is difficult to size. We therefore adopt a more power efficient and more importantly easily adjustable scheme for generating cascode voltages. The scheme is illustrated in what follows.

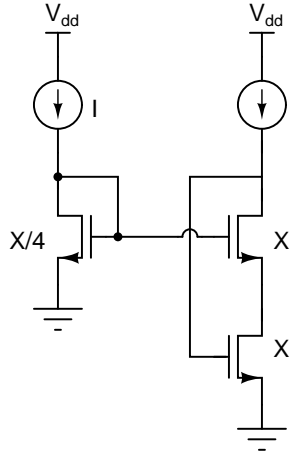


Figure 4.3: The usually used cascode voltage generation scheme

It is a well known fact that when we try to mirror currents using a simple current mirror, the drain voltage of the two transistors widely differ and thus the mirrored current is often smaller than the current we are trying to mirror. One way to make them nominally the same is to reduce a threshold voltage v_{th} from the gate voltage of the transistor and feed it to the drain so that the V_{ds} we achieve is $V_{gs} - V_{th}$. To generate a voltage source of value v_{th} , one can push a current I through a huge transistor (Mb) and put the transistor's gate source across the transistor Ma's gate-drain. This would mean that I flows through both the transistors but the V_{th} drop achieved will bring the drain source voltage of the transistors Ma and Mb nominally close. When we need a cascode voltage, we need a voltage $2V_{dsat} + V_{th}$ and this can be produced by adding a $V_{dsat} + V_{th}$ to the V_{dsat} we have already produced at the drain of the transistor Ma. This $V_{dsat} + V_{th}$ can be generated by pushing I through Mc. But now, $2I$ flows through the transistor Ma. This can be brought back to close to I by changing the first current source I to a ΔI and reducing Mb's size to a normal value. This procedure is illustrated in figure 4.4. As shown in the figure, this method can be extended to generate higher cascode level voltages as well.

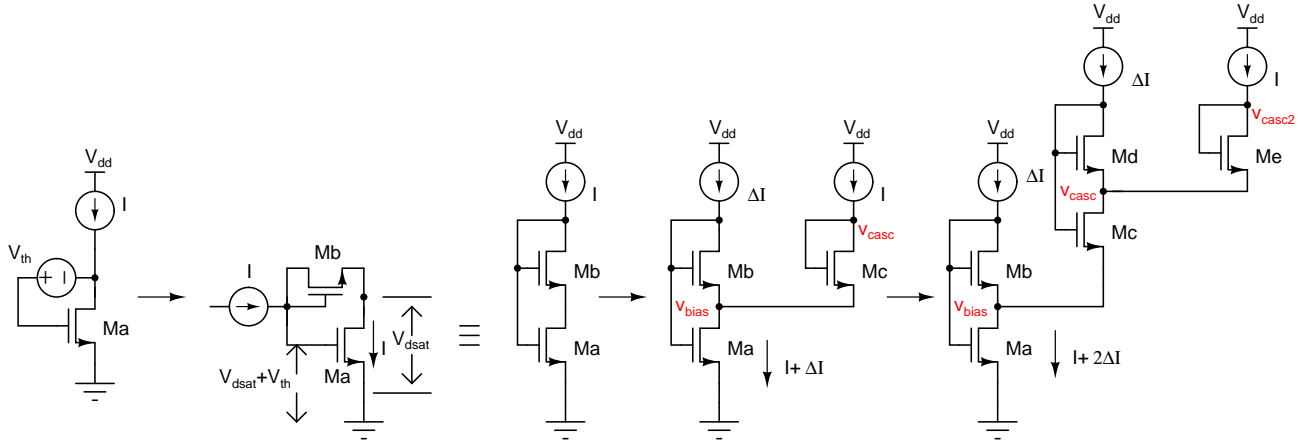


Figure 4.4: The cascode generation scheme used in this work

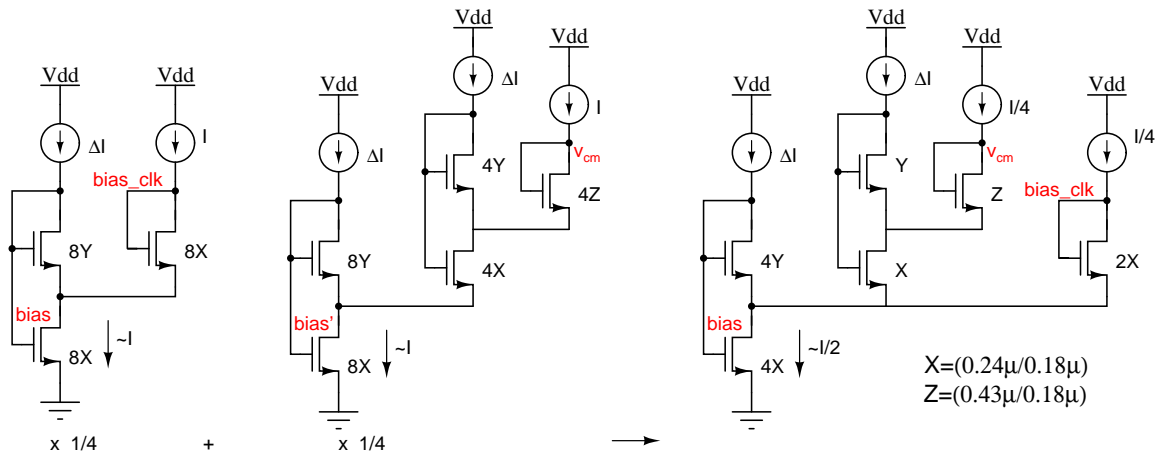


Figure 4.5: The bias generating section for the comparator's cascode voltages

The cascode voltages generated this way can be used to generate the bias voltages for the comparator as shown in figure 4.5. This method of generation of the cascode voltages serves as a good method to optimally utilize the available headroom in generating the different cascode voltages. However, due to use of short channel devices to reduce parasitic capacitances, current mirroring in to the capacitor tends to be imperfect across process corners. To solve this problem, we make use of replica biasing to determine the bias voltage for the current source alone as shown in figure 4.6. The transistors sizes are chosen such that a current I flows through the comparator's current source when the differential pair above the current source is completely switched. The input to the cascode transistor is replaced by the high level of the CML clocks fed to the comparator since this is the state in which the comparator spends most of the time.

circuits are shown in the figures 4.7, 4.8 and 4.9.

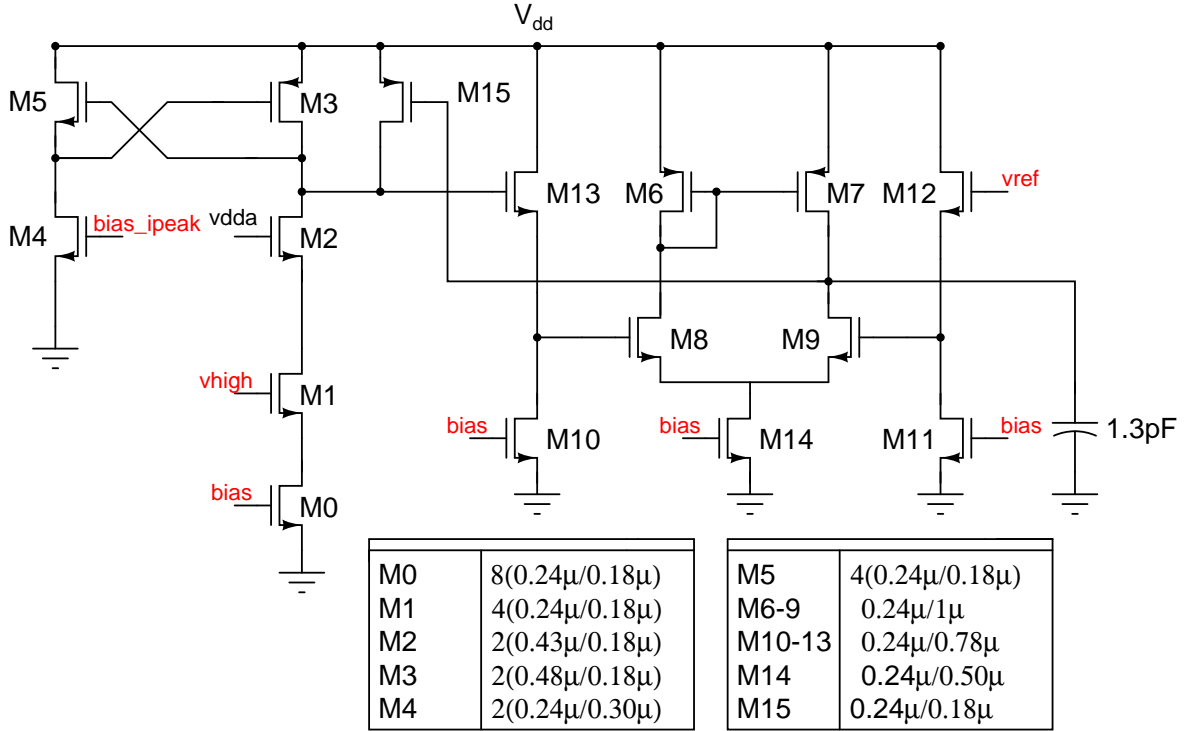


Figure 4.7: The negative feedback based bias generation to fix the output swing of the first latch

4.2 Clock generation

4.2.1 250 MHz Clocks

The clocks controlling the stage which subtracts the input and the reference is controlled by clocks which run at 250 MHz as described in the section 3.2.4. These clocks are derived from a clock which runs at 250 MHz which in turn is derived from the input master clock by frequency division. The frequency division is done using D flip flops which are made of C^2 MOS inverters as the latches. The flip flop is used to construct a counter which does the frequency division. The flip flop used is shown in figure 4.10. The inverted clocks which are required to drive the PMOS transistors of the transmission gates used in the input-reference subtraction section are generated locally to reduce the power consumption.

The clock so obtained runs at 250 MHz and is fed to a non-overlapping clock

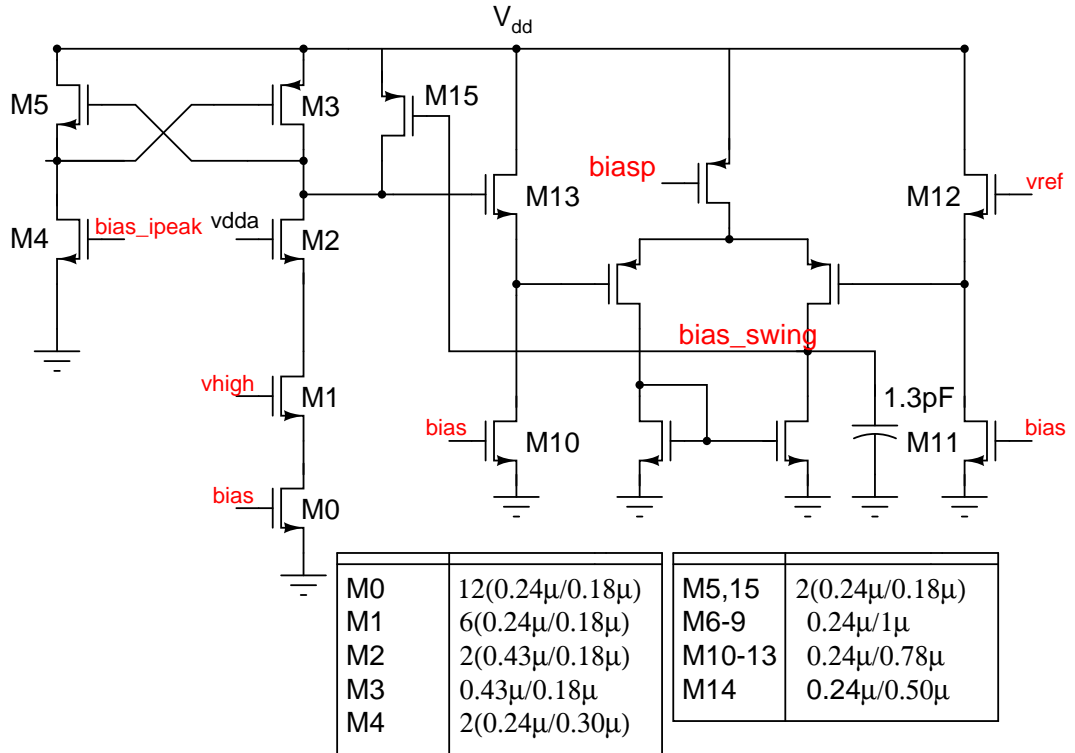


Figure 4.8: The negative feedback based bias generation to fix the output swing of the second latch

generator (which is two NAND gates in feedback). The output of the non-overlapping clock generator are two non overlapping clocks, each at 250 MHz and these clocks are fed to the comparator array through an inverter chain. The circuit generating clocks with the inverter sizing is shown in figure 4.11.

4.2.2 Latch Clocks

This subsection discusses the generation of the clocks to drive the latches in the comparators. The clocks are LCL, D_CLK and LRST and the complements of the first two. The first two are CML clocks while the last is a rail to rail CMOS clock. Since the delay of a CML gate and a CMOS gate are quite independent variables, it is difficult to keep the clocks aligned across process corner variations and temperature variations. Moreover, it was found that the flash, when driven by ideal voltage source based clocks, was drawing peak currents of the order of 1.2 mA. This would imply that the buffer which finally drives the load should at least be carrying 1.2 mA in it's tail current, if not more. Since the sizing

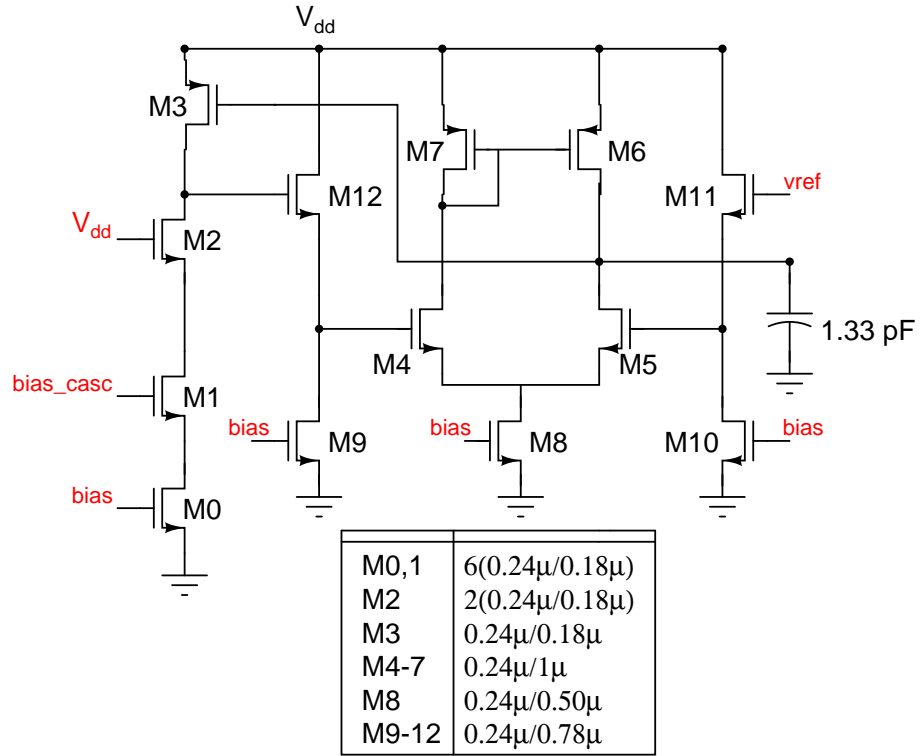


Figure 4.9: The negative feedback based bias generation to fix the output swing of the comparator

factor of a CML buffer is around 2, the power consumed to generate the clocks would be enormous. The problem is that we are consuming a lot of static power in order to meet a specification for the transition. This motivates us to use a CMOS based (specifically, a switched mode) clock driver for the CML clocks - LCL and D_CLK. This offers an additional advantage of almost synchronized CML and CMOS waveforms.

The very need to use CML clocks (low rail to rail swing) stems from the fact that CMOS clocks run rail to rail thereby consuming more power every switching cycle. The swing chosen in this particular design is 600 mV_{ppd} . This would imply that the difference between the levels is 300 mV and the fact that these clocks need to switch a NMOS differential pair implies that the CML levels are closer to the ground rail than the supply rail. The CMOS inverter can be viewed as a switched circuit switched to V_{dd} in one phase and switched to ground in the other phase. This suggests that the CMOS inverter cannot be operated between the desired voltage levels because the PMOS cannot turn on due to lack of enough overdrive. The solution is clearly to replace the PMOS by an NMOS transistor

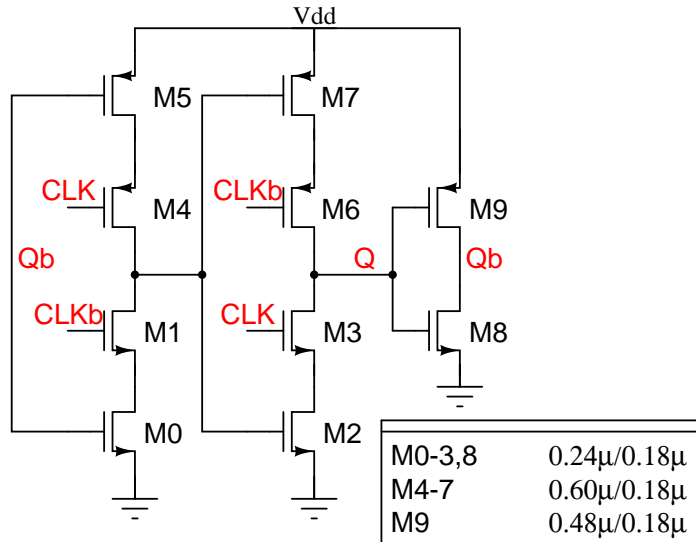


Figure 4.10: The C^2MOS based D flip flop used for frequency division

operated by the negation of the input. This leads to the output stage shown in 4.12. It can be switched with a pair of complementary CMOS clocks. Using a CMOS clocks to turn on an NMOS transistor between two levels other than the ground and supply nodes (say v_{low} and $v_{high}(>v_{low})$) would result in loss of overdrive and hence we employ clock boosting to reduce the switching transistor size and thereby reduce the power consumption in the driving stages preceding the converter. We therefore use a Nakagome charge pump to produce boosted clocks. Since the difference between the voltage levels is 300 mV , we boost the clocks to *ideally* run between v_{high} and $V_{dd} + v_{high}$ because the leakage current in the NMOS transistor connecting the output to the lower voltage level will be small for a gate-source voltage of 300 mV . Moreover, in practice, due to charge sharing at the output of the charge pump, the output voltage level will never reach $V_{dd} + v_{high}$.

Another concern in reducing the power consumption is the crowbar current drawn by the switching pair from the voltage levels while driving the comparator array. To reduce the crowbar current, we use non-overlapping clocks to drive the transistors and this is achieved by using the stages preceding the charge pump in figure 4.12. We now proceed to design the logic circuitry which generates the CMOS clocks to drive the CML to CMOS converter and the LRST signal.

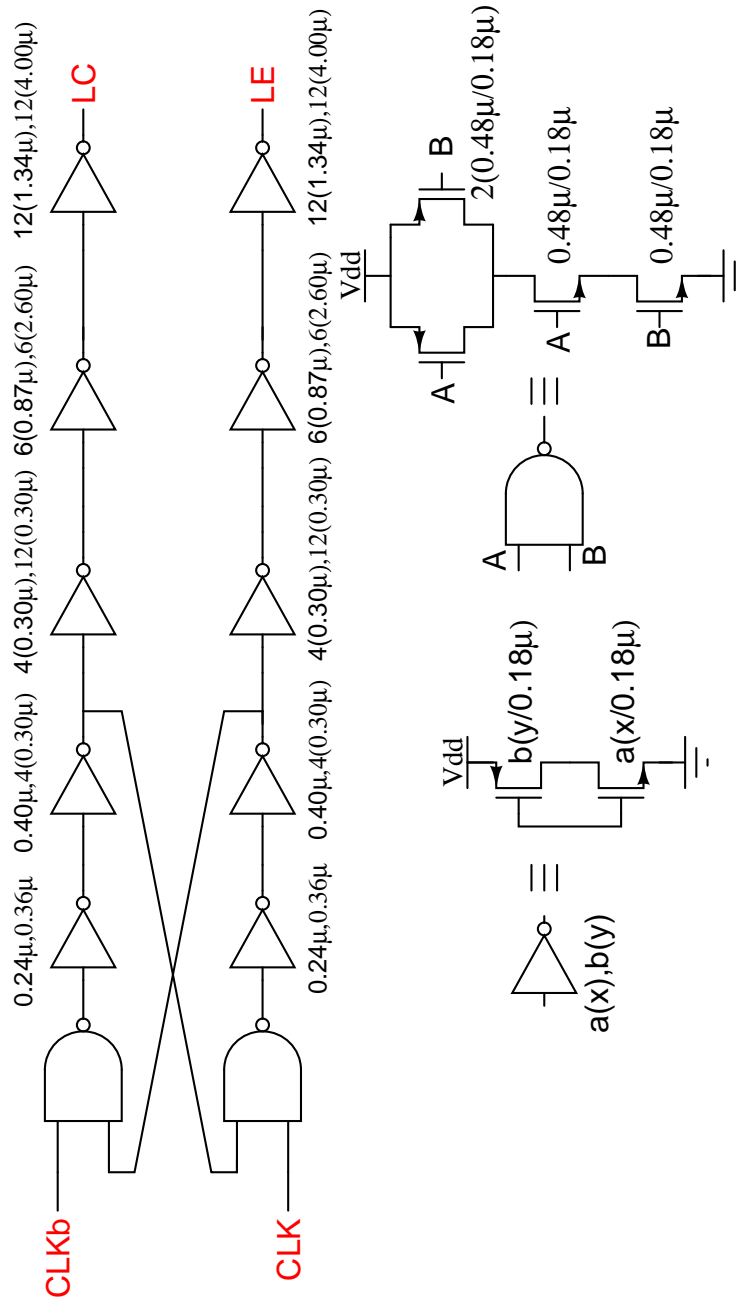


Figure 4.11: The inverter chain driver for the CMOS clocks for the input -reference subtraction stage of the comparator

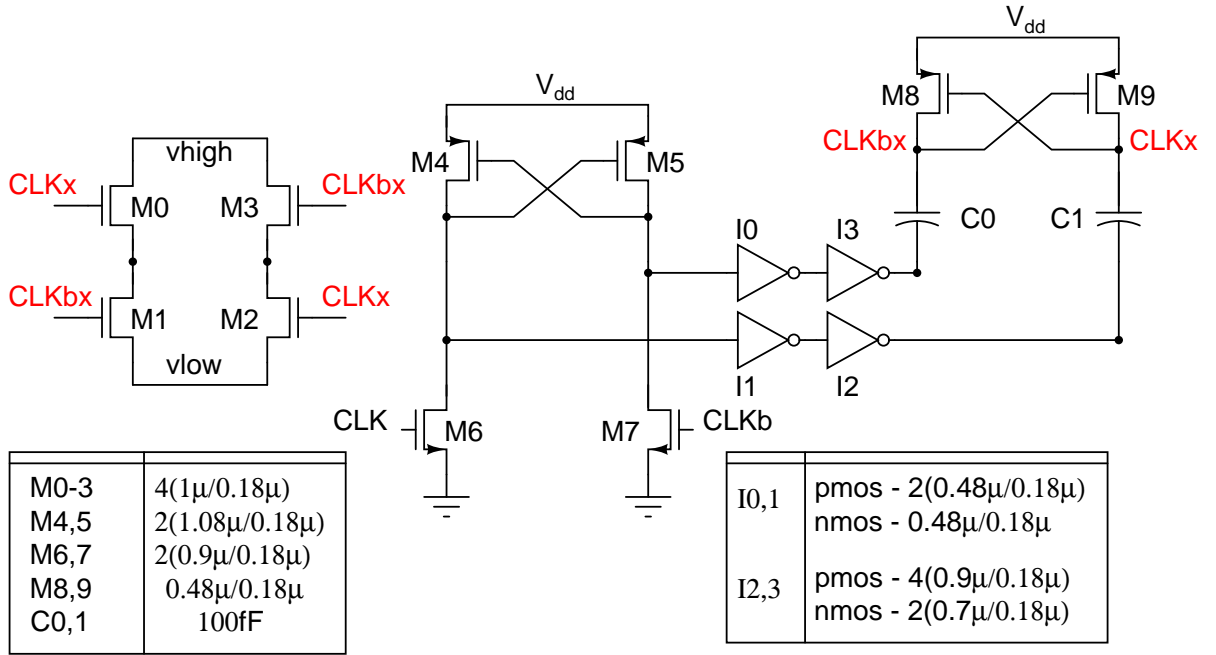


Figure 4.12: The CML clock driver which drives the comparator array taking in the CMOS equivalent of the clocks

Since we desire that the reset signal does not go low before the D_CLK signal goes low (second latch gets in to regeneration), it is essential that the signals are generated by closed loop means. To ensure that the LRST signal goes low only after D_CLK goes low, we generate LRST from D_CLK and LCL by passing LCLb and D_CLKb through a NAND gate as shown in figure 4.14. However, there still remains the generation of the CMOS equivalent of the D_CLK signal. This unfortunately cannot be generated by open loop means because the desired width of the D_CLK pulse is 200 ps and the variation across process corners is at least of the order of 100 ps. The only alternative therefore is to use some kind of a negative feedback loop to regulate the width of the D_CLK pulse. To do this, we require a tunable delay element. One kind of a delay element is a inverter with an extra PMOS between the inverter PMOS and the output thereby controlling when the rising edge of the output pulse with the gate voltage of the new PMOS. This however will require a larger size for the PMOS than a normal inverter. Instead we use, a delay element with a NMOS whose gate is controlled and is placed between a inverter's NMOS transistor and output of the inverter. This obviously can delay only the falling edge of the output and hence is used as shown in the figure 4.14. A clock is fed to the input of the tunable delay element just discussed and the output

Since we desire that the value of on time is at least 200 ps in the slowest possible corner, we fix x to be 0.3 and obtain an on time of 200 ps. The total time consumed for the pulse to go low is therefore $x \times 1\text{ ns} + 100\text{ ps}$. When there is a change of process corner and the rise/fall time goes below 100 ps, the on time only increases and the total time consumed for the pulse to go low reduces. The performance therefore only improves due to change in process corner if x is fixed to 0.3 in the slowest process corner.

To sync the fall of LCL pulse and the rise of D_CLK pulse, we pass CLKb through the same logic as CLK (which was used to generate D_CLK) but connect the other input of the NAND gate to V_{dd} to get a 50% duty cycle pulse for LCL. To reduce power consumption, we generate the inverted pulse for the CML to CMOS converter by passing through an inverter. This however introduces a delay between the inputs to the CML to CMOS converter. This delay between the inputs to the CMOS to CML converter can cause problems in the following stage by changing the width of the pulses. To put it more clearly, if we had connected M0's gate to ground, the output of this stage will be two pulses with pulses with 50% duty cycle but the output of the CMOS to CML converter will not be of 50% duty cycle. This comes because of the generation of non-overlapping clocks as shown in figure 4.12. However, if we had not generated non-overlapping clocks, in addition to the increase in crowbar current of the converter, the clock boosting is imperfect. To fix this delay, we introduce a transmission gate between the CMOS equivalent of the D_CLK pulse and the input to the CML to CMOS stage. This reduces the delay between the CMOS inputs to the CML to CMOS converter and hence improves performance to a certain extent.

4.3 CML level generation

The CML voltage levels are to be generated with a common mode from the voltage obtained from the circuit shown in figure 4.5 with a voltage difference of 300 mV. This is achieved by using a two stage opamp in feedback as shown in figure 4.16. The opamp used is a two stage opamp, the first stage being a cascode NMOS

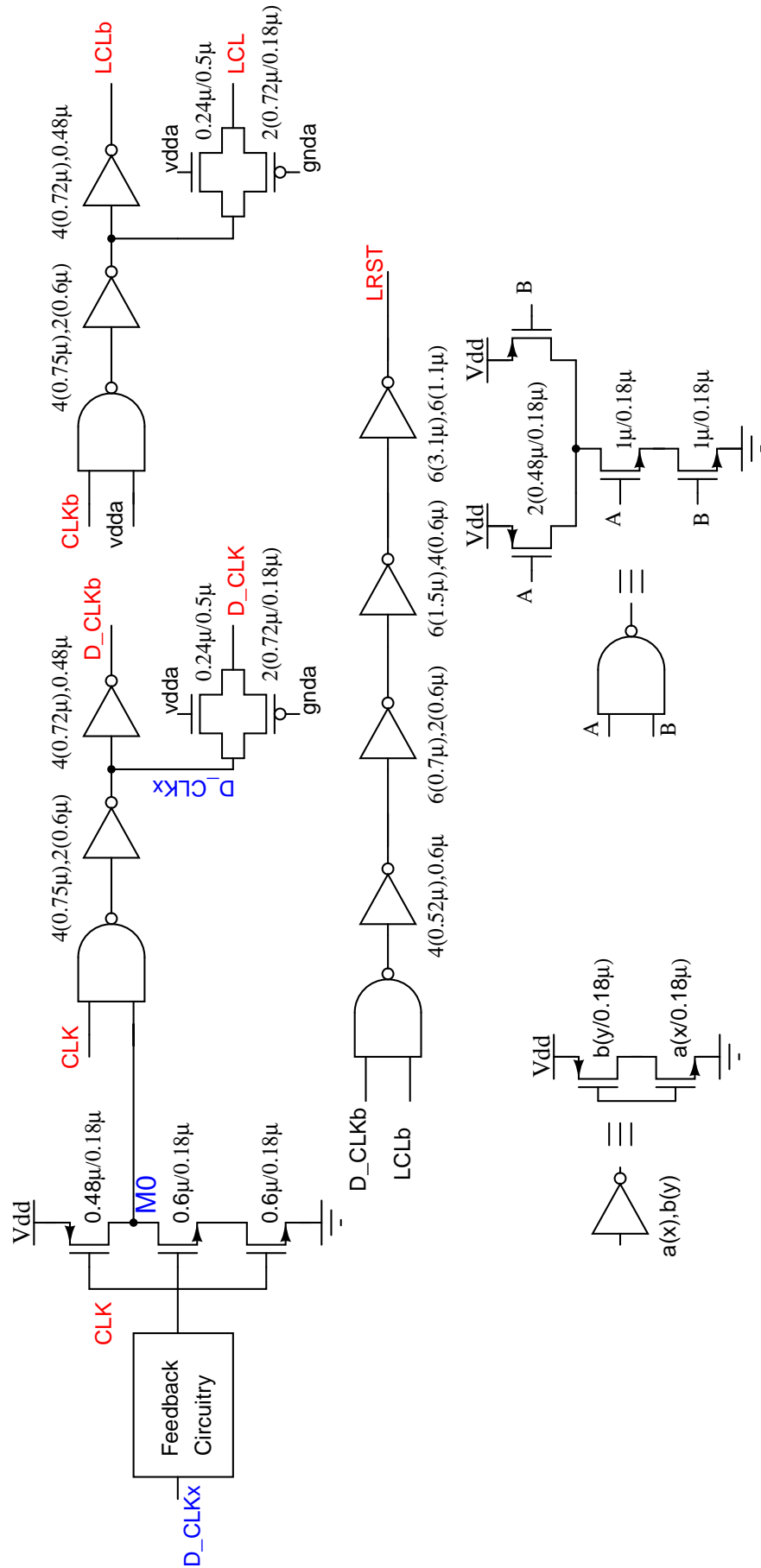


Figure 4.14: The logic which generates the LRST clock to be given to the comparator array and the CMOS

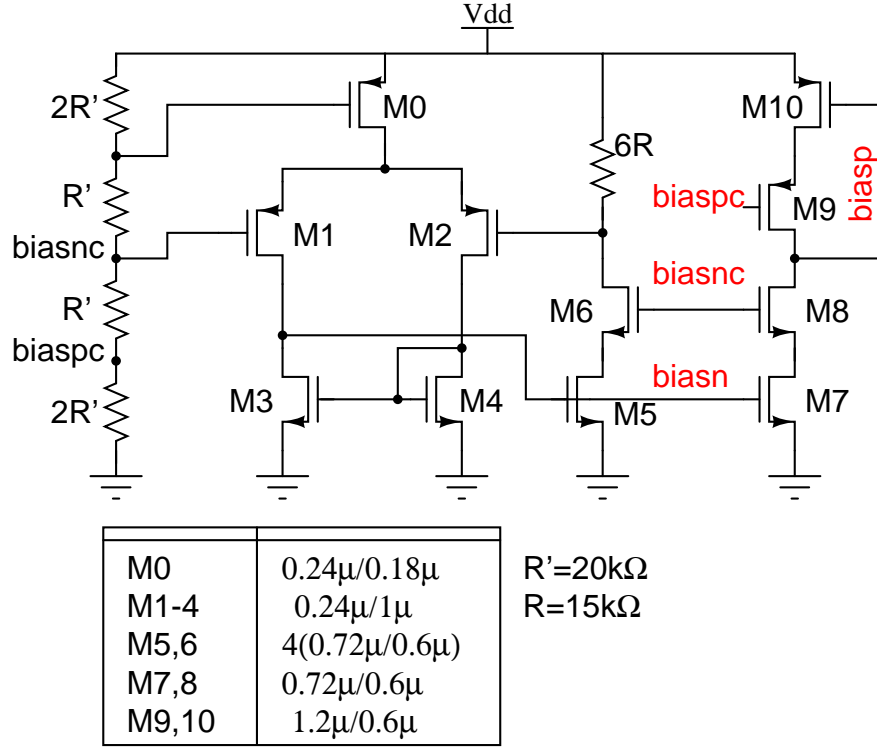


Figure 4.15: Circuit which generates a $\frac{V_{ref}}{R}$ reference current to produce voltage levels for the CML clocks

differential pair and a class AB second stage to supply the load current drawn by the CML to CMOS converter (The total load current drawn is approximately $140 \mu A$). A current I_{ref} is pushed in to the opamp's input and is passed through the feedback resistor R to produce a voltage drop of $150 mV$. The current I_{ref} is generated by fixing the drop across a resistor, $6R$, to $\frac{V_{dd}}{2}$. The current I_{ref} so obtained equals $\frac{V_{dd}}{12R}$. The circuit which accomplishes this task is shown in figure 4.15.

4.4 Flash Reference Level generation

The 4-bit flash ADC designed in this work operates with a full scale of $3V_{ppd}$. This implies that the flash operates with reference voltage levels as linear divisions of voltages $1.5 V$ apart and centrally located in the available voltage range $0-1.8 V$. In other words, the reference voltages for the comparator array V_{refp} and V_{refm} are $1.65 V$ and $0.15 V$. These voltage references should be able to supply a current $\frac{1.5}{16R_{ladder}}$ where $R_{ladder} = 1 k\Omega$ form the equivalent ladder resistance between to

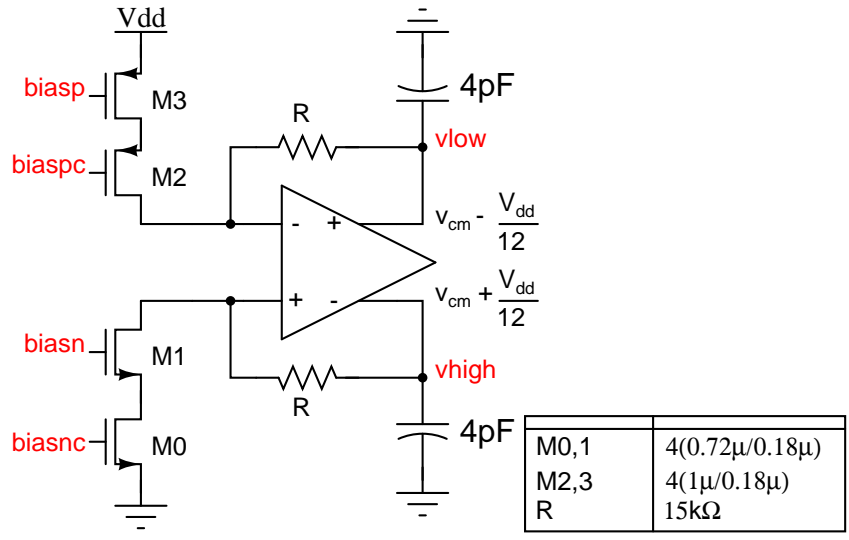


Figure 4.16: Circuit which generates the CML clock levels from the reference current generated using the circuit shown in figure 4.15

consecutive reference levels. These voltages are therefore generated in a method similar to the CML voltage levels by producing a drop of 0.75 V across the feedback resistor and fixing the output common mode to 0.9 V. This is shown in figure 4.17.

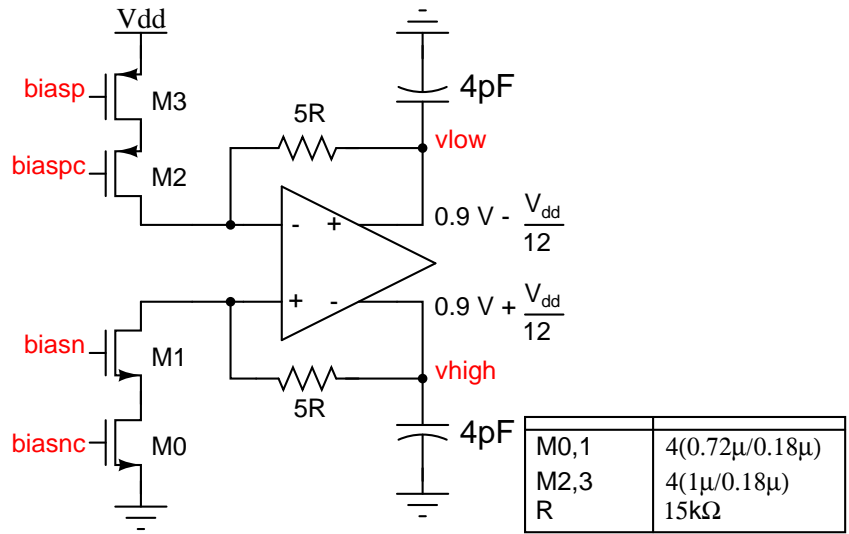


Figure 4.17: Circuit which generates the flash reference levels from the reference current generated using the circuit shown in figure 4.15

CHAPTER 5

Binary Code generation and Deserialisation

The following chapter discusses the processing of the differential thermometer code output of the flash to obtain binary coded data. The output of the flash is a thermometer code corresponding to a 4-bit data. Since the streams are at 1 GHz, for convenience, they need to be deserialised to 250 MHz streams to be brought out of the chip. However, deserialising 15 streams of data at 1 GHz to produce 250 MHz streams would result in 60 streams which is quite a huge number to manage. Moreover, these streams have to be processed to convert them to binary data. In view of implementation convenience, therefore, the differential thermometer code at 1 GHz is first converted to CMOS streams. This conversion takes the largest share of power, as would be seen in the discussion that is to follow. However, once this conversion is complete, digital processing becomes a lot easier when compared to processing in CML logic.

5.1 System level choice of thermometer to binary conversion logic

A third order $\Delta - \Sigma$ modulator is simulated with random offsets using a modified form of delsig toolbox (3) to understand the effect of different types of thermometer code to binary code conversion schemes. A third order DSM with an OBG of 3 and a 16 level quantizer (mid-rise, 4-bit) is chosen for simulation purposes. A peak SNR of 111 dB is expected of the DSM in absence of any offsets. Three schemes of conversion are employed for the thermometer code to binary conversion - summing the bits of the thermometer code, transition detect in thermometer code followed by a 4-bit encoder, 3-bit majority bubble correction followed by transition detect and a 4-bit encoder. 100 comparators with random offsets are simulated

for each σ_{offset} which is varied from 0 to 0.4LSB. (In other words, a Monte-Carlo simulation is performed with σ_{offset} varying between 0 to 0.4LSB. The output SNR is calculated by using 1024 point FFTs averaged by 100 periodograms. The resultant SNR distribution and the mean SNR is plotted in figure 5.1 below. It can be seen that the summing method works fine cleanly till 0.4LSB while the output SNR of the second scheme (no bubble correction) starts plummeting after 0.2LSB. Bubble correction using majority code ameliorates the situation by sustaining up to about 0.35LSB.

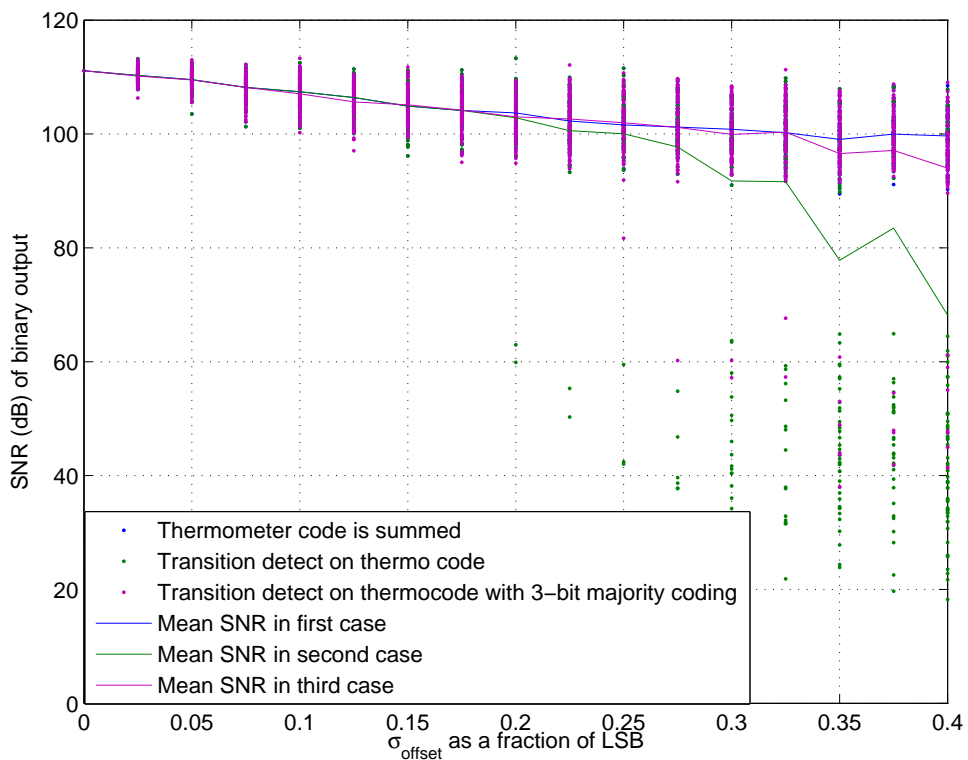


Figure 5.1: Effect of random offsets in the flash’s comparators on the output SNR of $\Delta - \Sigma$ loop

In terms of implementation complexity, summing the thermometer code is the one which has the greatest complexity. It is followed by transition detect using bubble correction and the transition detect is the simplest. Since the simulation results predict that transition detect without bubble correction can potentially be dangerous beyond a random offset of standard deviation 0.2LSB, we stick to using bubble correction logic followed by transition detect as a modest trade off between complexity and reliability.

The CML data streams available at the output of the flash are situated around a common mode of 1.6 V. This value is quite high for the DAC which is expected to follow the flash in $\Delta - \Sigma$ modulator loop. Moreover, the DAC's input pair would be atleast a few microns in width and minimum length, $0.18 \mu m$. This can be too much for the flash to drive given that it has to drive the thermometer to binary converter section. The output of the flash is therefore followed by a nmos source follower which steps down the common mode from around 1.6 V to close to 0.9 V. Body effect in the nmos source follower is bound to affect the swing at the output of the source follower to a moderate extent. The input swing to the source follower is made large enough to be able to switch the DAC even with the reduced swing from the output of the source follower. The following section, therefore, was designed with the assumption that the source follower which follows the flash output increases the driving capacity of the flash to a substantial extent.

5.2 Implementation

5.2.1 Conversion of CML data streams to CMOS data streams

The output of the source follower which follows the flash was found to have a swing close to 350 mV (differential) after a reduction in swing due to body effect. To convert this signal to CMOS, a clocked regeneration scheme is used so that the resultant output waveforms are all synchronized. This helps to prevent unnecessary glitches in the digital logic which follows. The regeneration is done by use of a CMOS latch which is interfaced with the input through a differential pair. The circuit is shown in figure 5.2.

As seen from the figure, the CMOS latch takes one clock CLK, regenerates during it's on phase and resets during the off phase. During the off phase, since the latch is disabled, when it is reset, the output waveforms both reach 0.9 V (due to symmetry, the charge on the parasitic capacitors is equally shared amongst the two nodes when they are joined together). In order to minimized dynamic offset,

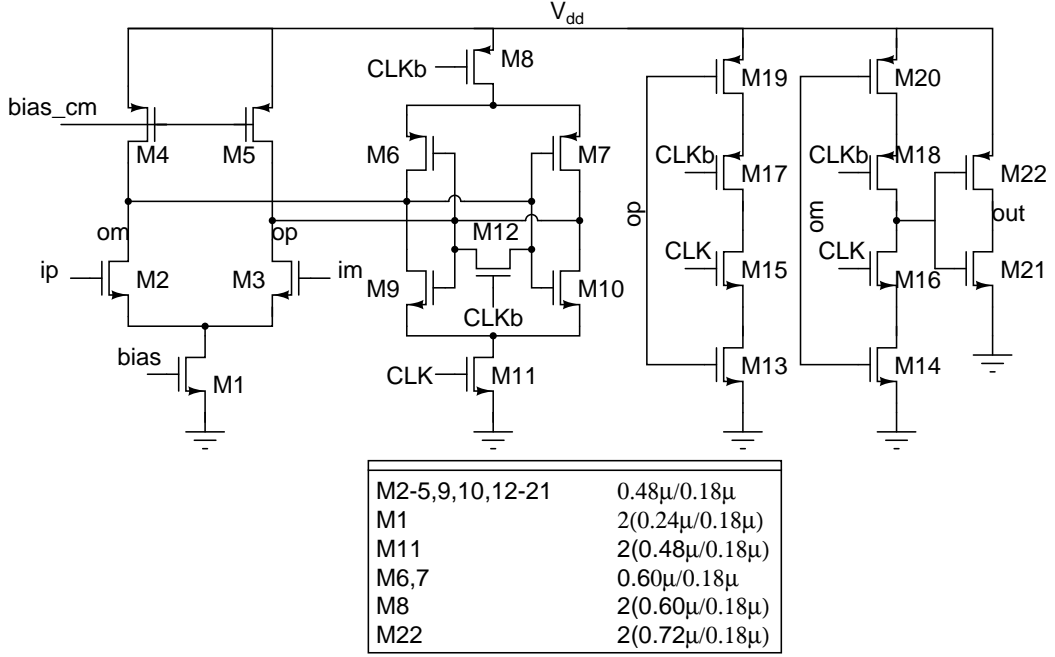


Figure 5.2: Circuit which converts the input CML streams to CMOS streams

the CML differential pair which takes the input is tuned to have a output common mode of 0.9 V by use of negative feedback in fixing the bias voltage of the active pmos load. The output of the latch is *tracked* by the C^2MOS inverter during the regenerating phase of the latch and is sampled before the latch is reset. The output of the C^2MOS inverter is fed to a CMOS inverter to be able to drive the section which follows.

5.2.2 Bubble correction

The bubble correction involves 3-bit majority coding of the CMOS version of the input data stream. The three bit majority coding implements the logic

$$out[i] = in[i - 1].in[i] + in[i - 1].in[i + 1] + in[i].in[i + 1]$$

The first and the last bits of the thermometer code $in[14]$ and $in[0]$ go through the logic to maintain the same delay. However, for these two bits, all the inputs to the logic are the same as $in[14]$ and $in[0]$ respectively. The circuit which implements this logic is shown in figure 5.3

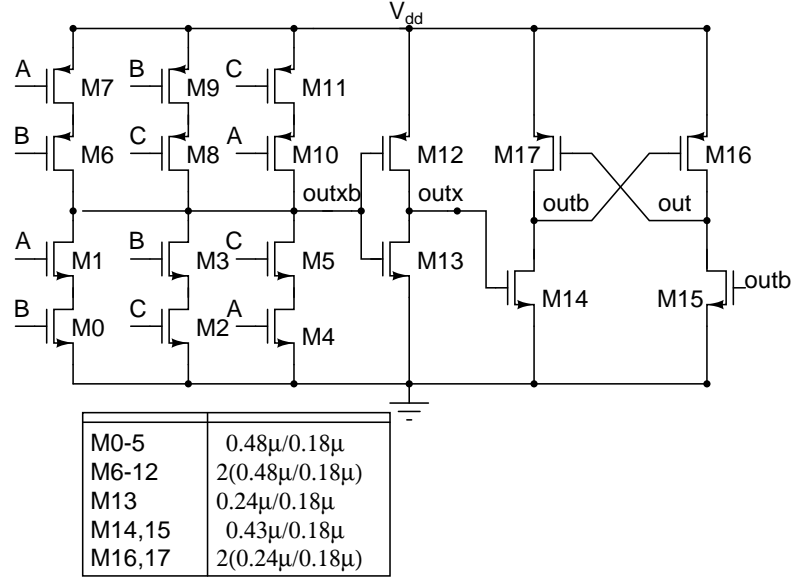


Figure 5.3: 3-bit majority coding based bubble correction logic

5.2.3 Transition detect and Conversion to binary

The bubble correction logic implemented in the previous subsection will be able to correct upto 1 bubble in the input thermometer code. We can therefore assume that the output of the bubble correction logic consists of string of 1's followed by a string of 0's. The binary logic level which represents all 1's is 1111 and the logic level which represents all 0's is 0000. Therefore, if there k 1's in the (corrected) thermometer code, the output binary code is the binary equivalent of k. To evaluate k, a transition detect is performed on the thermometer code using the logic

$$out[i] = in[i].\overline{in[i+1]}$$

This logic is implemented by means of a NAND gate and the output has a 0 at the code position which has the last 1 of the string of 1's. The last bit (the comparator with the highest reference voltage) goes through the logic with next bit as 1 to keep all the waveforms synchronized. The transition detect thus obtained is converted to binary logic using a pre-charge logic. The pre-charge logic functions as follows. In one phase of a clock CLKb, the four output nodes are charged to ground by turning on a NMOS transistor. In the other phase, the NMOS transistors are turned off. The transition detected code is fed to PMOS transistors connected

to the output nodes. For each code position k in the transition detect code, a PMOS transistor is connected to bit position $b[i]$ if $b[i] = 0$ in the binary equivalent of k . The output obtained in this clock cycle is sampled off using a C^2 MOS kind of logic. While implementing the C^2 MOS logic, the fact that the output of the pre-charge logic goes to 0 when CLK goes low is taken into account and the intermediate NMOS transistor is eliminated in the C^2 MOS implementation. The output of this C^2 MOS stage is passed through an inverter to protect the saved waveform before passing through the subsequent stage. The transition detect and the pre-charge logic are shown in figure 5.4.

The clock fed to the pre-charge logic is obtained by passing the sampling clock of the CML to CMOS data converter through the bubble correction logic and the transition detect logic to approximately sync the clock with the data transitions. The delay seen in the CML to CMOS data converter to the output transition approximates the delay in the pre-charge logic's response to the output. As a result, the data streams and the clock fed to the pre charge logic remain approximately synced.

5.3 Deserialisation

The following section discusses the deserialisation of the output streams of the thermometer to binary converter which switch at 1 GHz. The outputs of the deserialising stage are 16 streams of data each running at 250 MHz. Deserialising the streams at 1 GHz requires 4 CMOS clocks each running at 250 MHz and spaced 1 ns apart to sample successive samples of the 1 GHz streams. These are realized by using a commonly used CMOS D flip flop with its negated output tied back to its input. The resulting block is a divide by 2 clock divider. Two such stages are cascaded to achieve a divide by 4 clock division and the two internal nodes of the clock divider (which are the outputs of the positive and the negative latches in the D flip flop) produce the required 4 clocks. The divide by 2 clock division circuit is shown in figure 5.5. The block shown in figure 5.5 is used in figure 5.6 to produce the 4 clocks required to deserialise the streams. The clocks generated

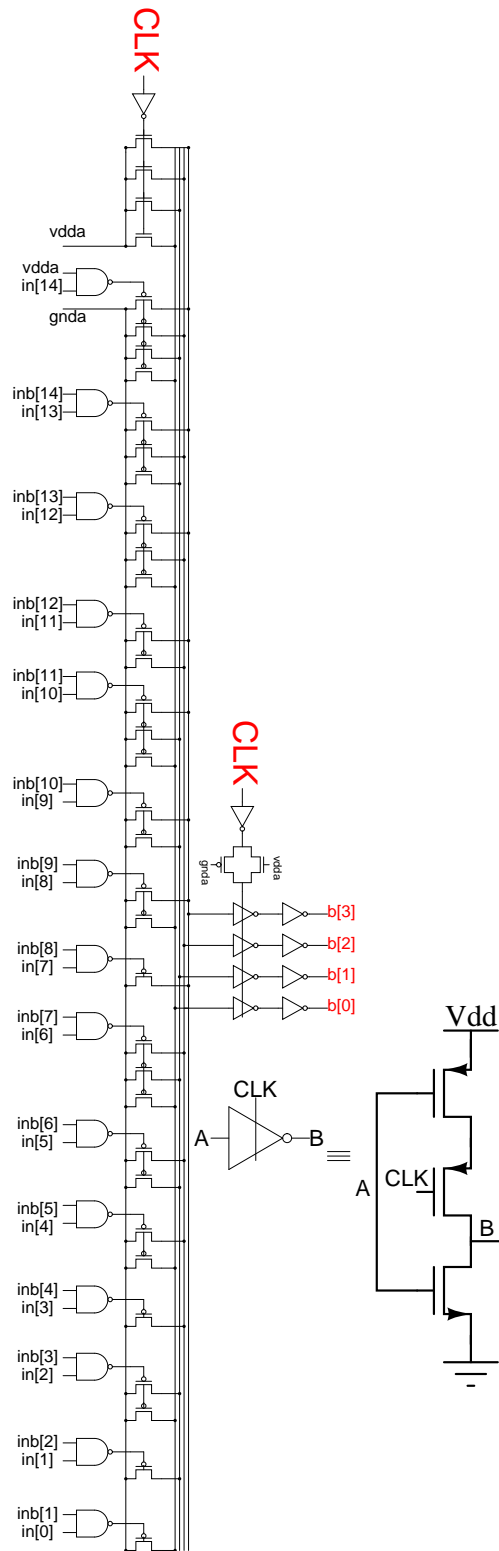


Figure 5.4: Transition detect and pre-charge logic

as shown in figure 5.6 are fed to the C²MOS clocking stages shown in figure 5.7 to produced the 250 MHz streams.

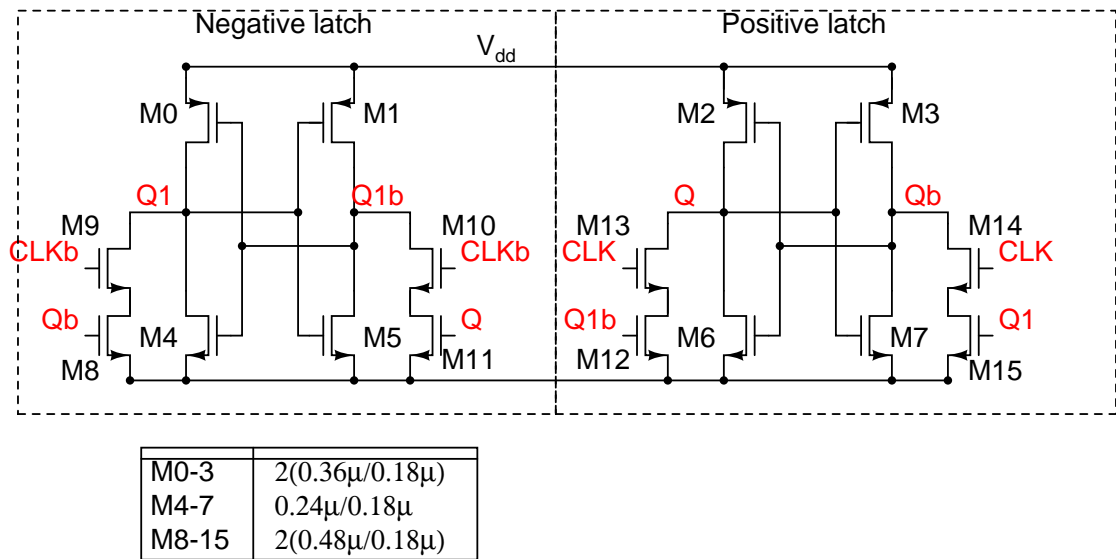


Figure 5.5: The divide by 2 clock divider used in generating the sampling clocks for the deserialiser

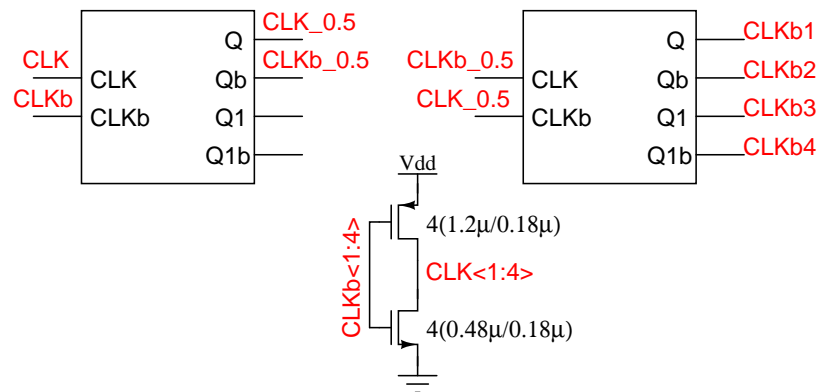


Figure 5.6: The divide by 4 clock divider realized using the divide by 2 clock divider shown in figure 5.5

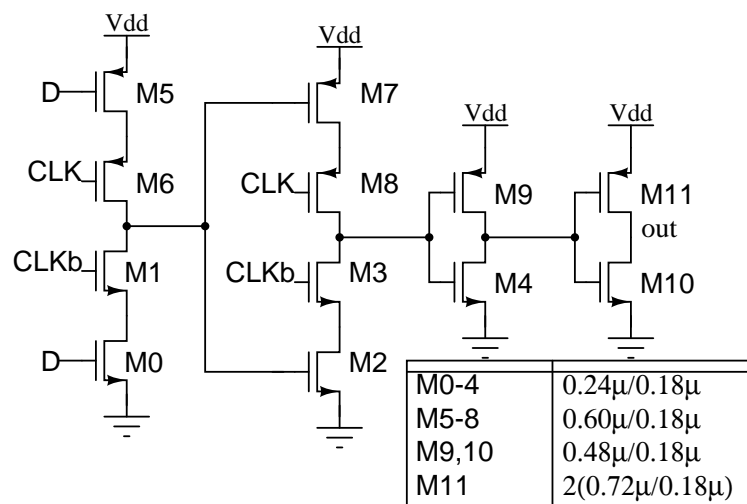


Figure 5.7: The c^2 MOS sampling stages to sample off the 1 GHz streams to produce the 250 MHz streams to the output

CHAPTER 6

Layout and Simulation results

All the previously described sections have been laid out and extracted for parasitic capacitances. The laid out comparator array occupies an area of $470 \mu m \times 220 \mu m$. All the simulation results mentioned in this section correspond to the worst performing corner i.e., Slow NMOS, Slow PMOS, $75^\circ C$, minimum resistor and minimum MIM capacitor. As already mentioned, each comparator consumes an approximate current of $90 \mu A$. This gives a power consumption of $1.53 mA(2.75 mW)$ in the comparator array. The clock generation and the bias generation stages consume $4.45 mW$ of power while thermometer to binary conversion stage consumes close to $3.3 mW$ and the de-serialiser consumes $810 \mu W$. The figures 6.1 and 6.2 show the waveforms of the subtraction section clocks and latch's clocks produced by the laid out clock generator when driving the laid out comparator array.

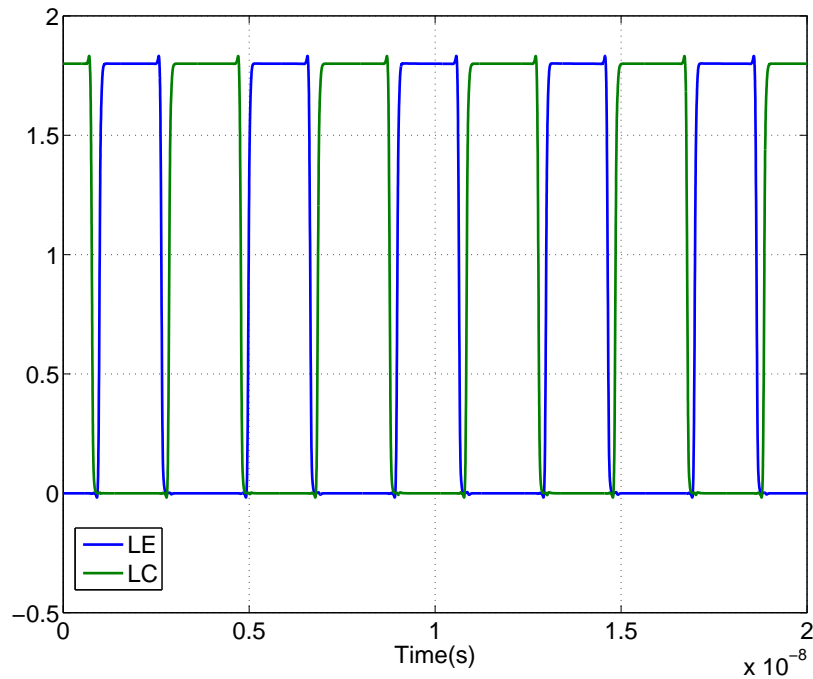


Figure 6.1: Clocks generated by the laid out clock generator driving the input-reference subtraction stage of the laid out flash

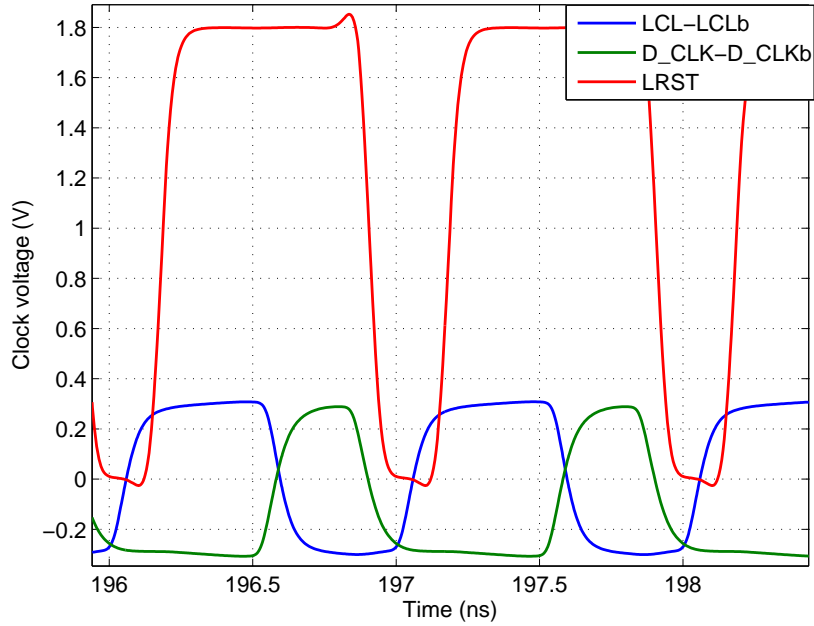


Figure 6.2: Clocks generated by the laid out clock generator driving the latches in the laid out flash

The laid out comparator is excited with a 6LSB peak to peak differential ($562.5 mV_{ppd}$) sinusoid at $\frac{31}{64} GHz$ superposed on a $3 V_{ppd}$ sinusoid at $\frac{5}{256} GHz$ scaled to a $3 V_{ppd}$ full scale. The output of the comparator is passed through a source follower and is then fed to an ideal DAC cell to get a glimpse of the switched waveform at the output of the DAC. The waveforms shown in figure 6.5 give an overall view of the output of the comparator. Figure 6.3 shows a closer look at the waveforms at the output of the comparator and the output of a ideal DAC cell. It can be seen from the figure 6.3 that the delay in the comparator is of the order of 400 ps. The SQNR obtained from the output of a ideal single point sampling DAC from this flash is 98.8 dB. The spectrum at the output of the flash when excited by a $2.4 V_{ppd}$ tone at $\frac{5}{512} GHz$ is shown in the figure 6.4. However, due to some unknown issues with the thermometer to binary converter, its output shows a spectrum which is not even noise shaped. This is yet to be resolved.

The figures 6.6 and 6.7 show the laid out versions of the flash (with it's comparator array and clock generator) and the thermometer to binary converter (along with the deserialiser) with the individual sections labeled in the figures.

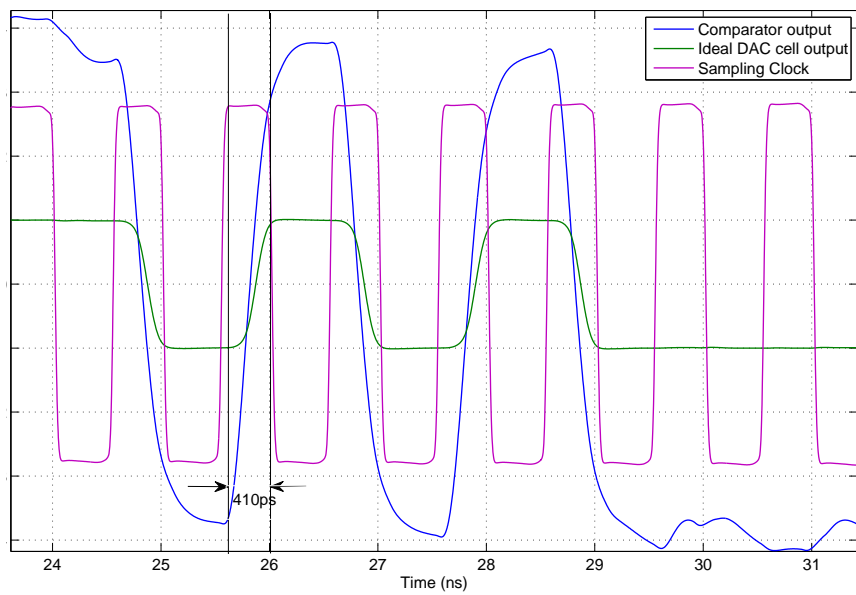


Figure 6.3: A closer look at the output of comparator and ideal DAC cell

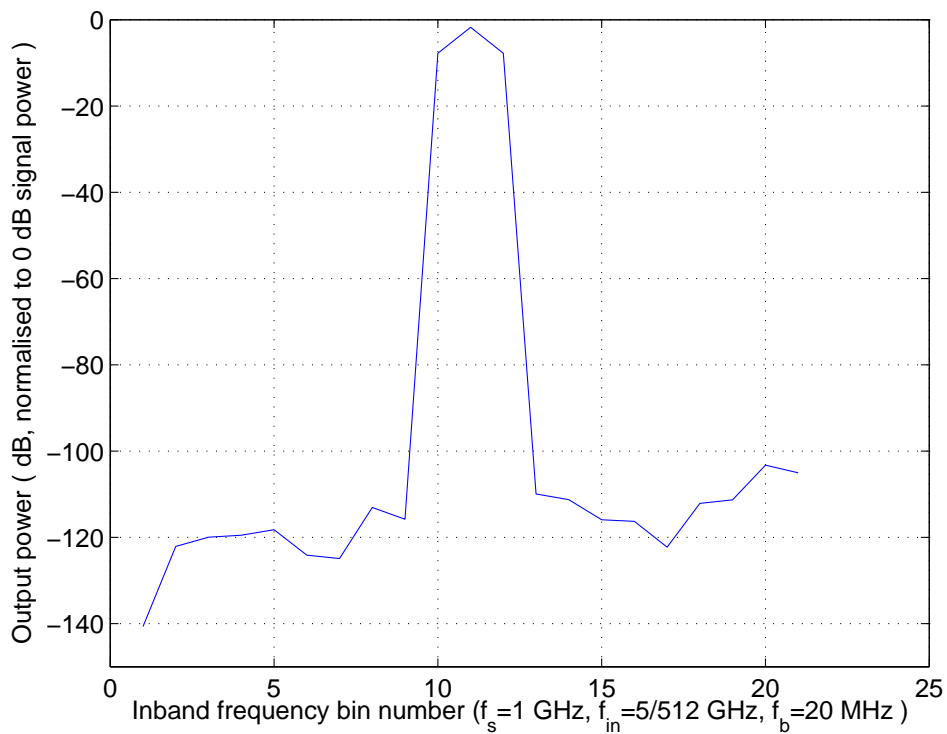


Figure 6.4: Spectrum at the output of a single point sampling DAC in the loop

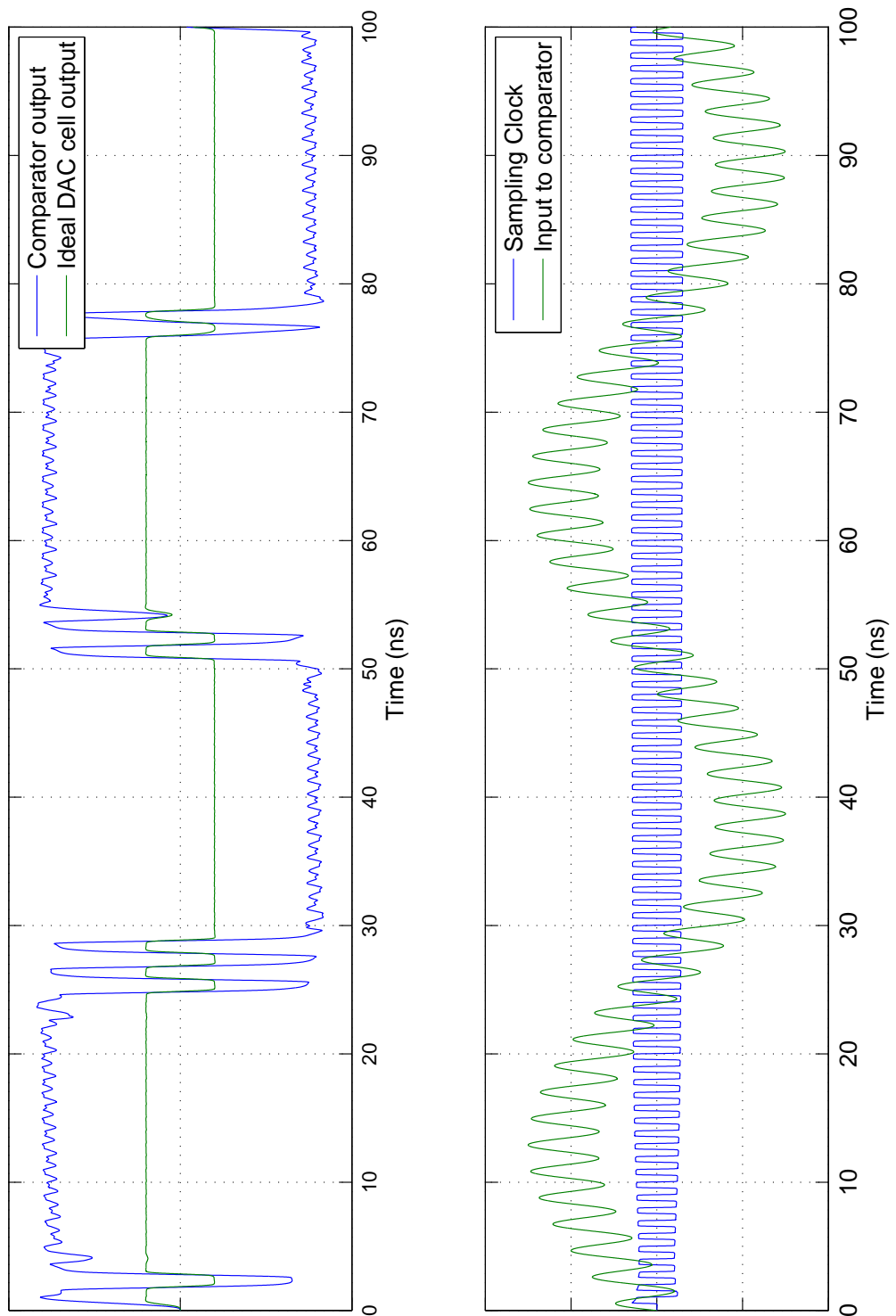


Figure 6.5: Waveforms at the output of the comparator and the ideal DAC cell

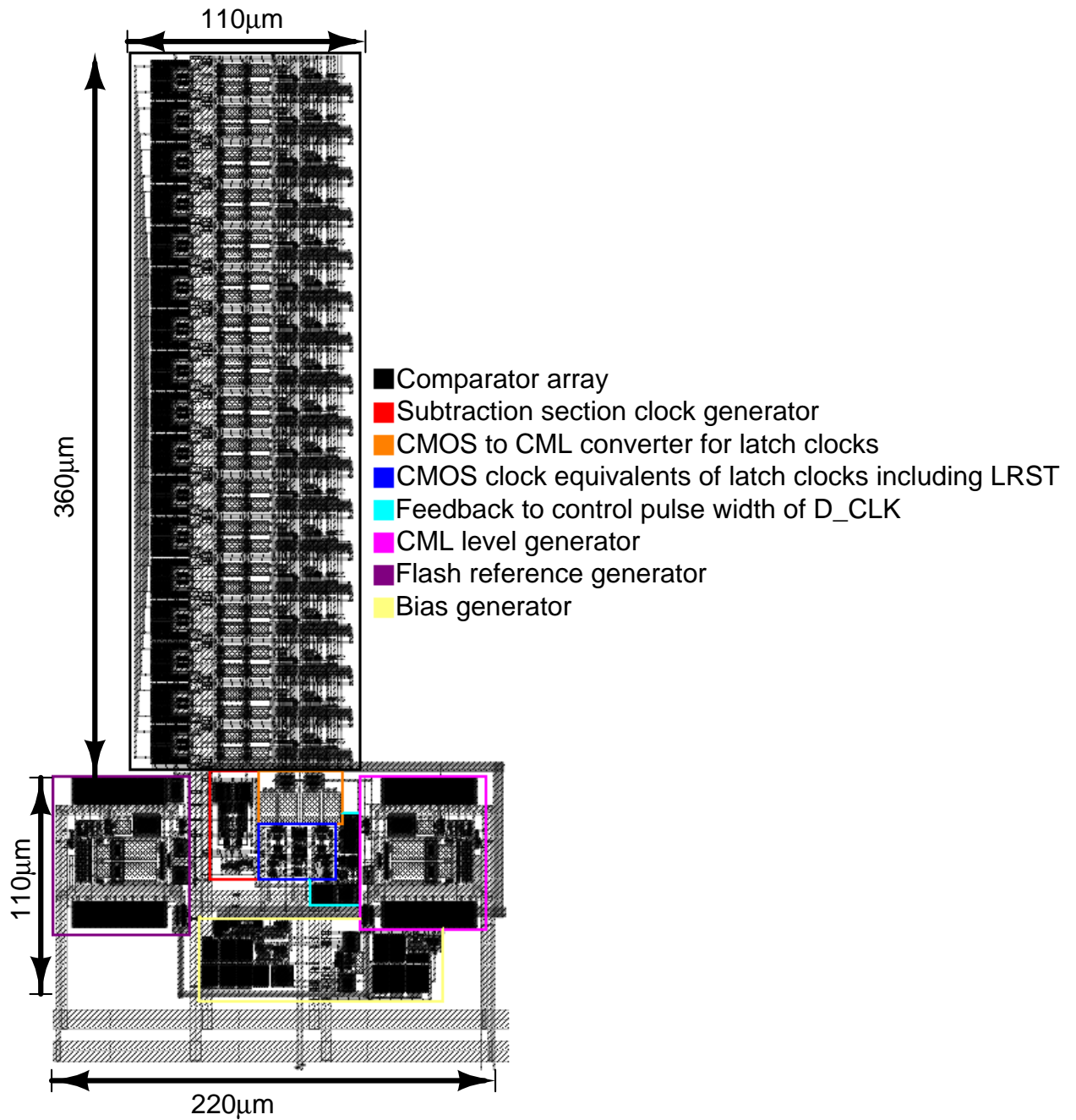


Figure 6.6: Layout of the flash with its clock generator

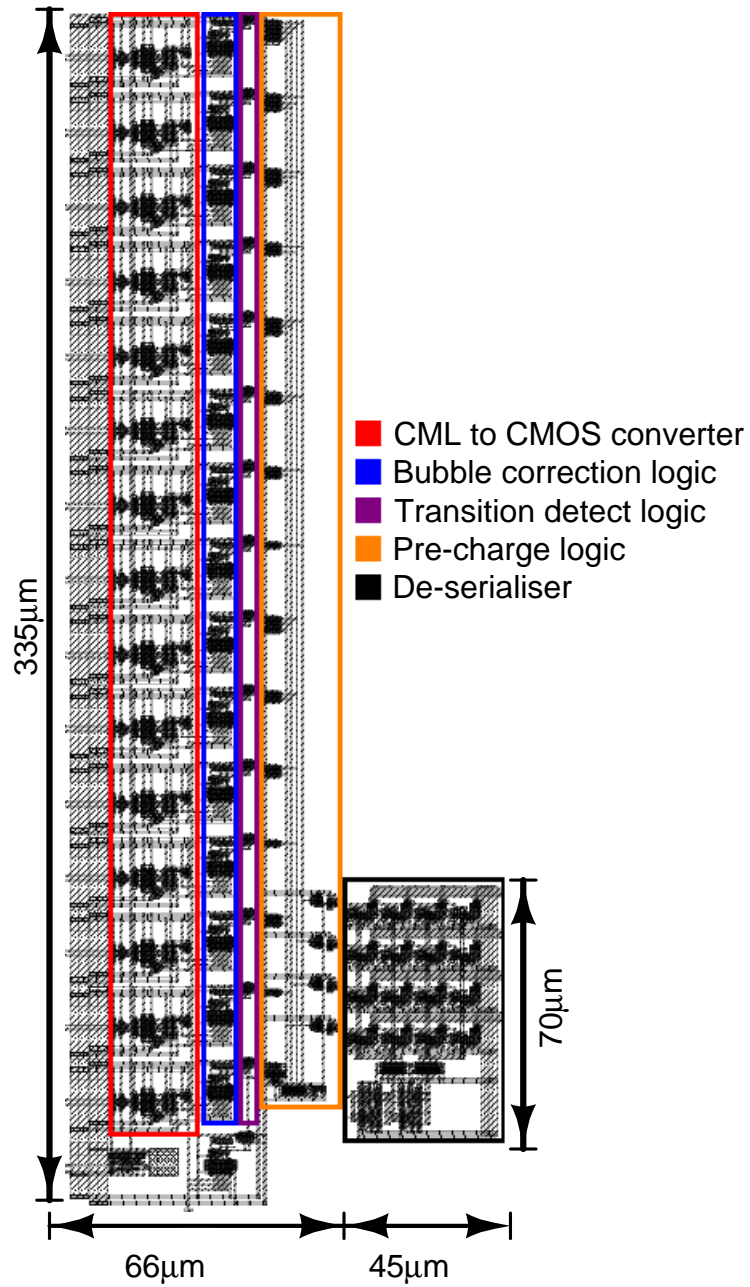


Figure 6.7: Layout of the thermometer to binary converter and the de-serialiser

CHAPTER 7

Conclusions

A Flash ADC clocked at 1 GHz for use in a $\Delta - \Sigma$ modulator was designed and verified for performance on a $0.18 \mu m$ UMC process. It was found that it consumes a power of 2.75 mW in the comparator array with its references, 4.45 mW in the bias and clock generation and 4.11 mW in the thermometer to binary conversion and deserialising the 1 GHz streams to 250 MHz streams. It has a delay of 400-500 ps depending on the input it samples. The flash output is passed through a source follower to level shift the output common mode and to improve driving capability. The output of the source followers was fed to a differential pair terminated with ideal resistors and the time elapsed between the sampling clock's sampling instant and the switching time of the output of the differential pair was used to measure the above mentioned delay. It produces an in-band SNR of 98.8 dB with ideal blocks for the loop filter and the DAC in the $\Delta - \Sigma$ modulator. However, there seems to be some problem with the thermometer to binary conversion and the output of this stage shows sudden glitches.

7.1 Things to be completed

Improving the rise time of DAC output

The current design relies on the gain provided by the CML stages to a substantial extent due to the extremely small time available for regeneration. As a result, when the input is close to the reference the rise time one obtains at the switched output of the DAC can be large. In order to reduce this, one can add more gain stages in the signal path before feeding it to the DAC. This adds to the delay in the loop as expected. It has been suggested by Vikas that this can be done by adding

CMOS inverter style gain stages in the signal path to accomplish this purpose. In this case, one can get rid of the CML buffers at the end of the comparator because the purpose of the CML buffer is precisely the same. This reduces the power consumption in the comparator array as well. Another added advantage is that one does not require a CML to CMOS converter for the data streams in the thermometer to binary converter. This cuts down the major contributor to power consumption in the thermometer to binary converter.

Glitches in the output re-constructed from binary code

Though the output of the ideal single point sampling DAC produces a SQNR of 98.8 dB, the output reconstructed from binary code shows glitches as if the thermometer code has more than one bubble (The thermometer to binary converter has a bubble correction logic which can correct upto 1 bubble in the thermometer code). This needs to be fixed.

CHAPTER 8

Appendix A

Description of Cadence Cell Hierarchy

Cell name	Description
TEST_CTDSM_DSER	Test bench to determine output SNR of single point sampling ideal DAC and thermometer to binary converter
TEST_CTDSM_IDEAL_CLOCKS	Test bench to determine output SNR of single point sampling ideal DAC with ideal clock generator
TEST_VB_BIAS_STAB	Test bench to determine stability of feedback loops used to fix swing of latches and CML buffer in comparator
TEST_VB_CLOCKGEN	Test bench to test the clock generator when loaded by the laid out comparator array
TEST_VB_COMPARATOR	Test bench to test a single comparator cell with real clock and bias generator
TEST_VB_LATCHES_CLOCKGEN	Test bench to test the feedback loop used to control the D_CLK pulse width
VB_1OFN_BIN	Pre-charge logic circuitry to convert 1 of N encoded data to binary data
VB_BIAS_CASC_GEN	Bias generator to fix currents in comparator and input common mode to first latch
VB_BIAS_COMP_SWING	Negative feedback based bias generator to fix swing of CML buffer in comparator
VB_BIAS_GEN	Bias generator for the entire flash.
VB_BIAS_LATCHA_SWING	Negative feedback based bias generator to fix swing of first CML latch in comparator
VB_BIAS_LATCHB_SWING	Negative feedback based bias generator to fix swing of first CML latch in comparator
VB_BIAS_LATCHES_CLOCKGEN	Negative Feedback circuitry to control D_CLK pulse width

VB_BIAS_T2B	Bias generator for the thermometer to binary converter
VB_BUBBLE_CORRECT	Bubble correction logic
VB_C2MOS_DFF_DSER	C^2MOS sampler for De-serialiser
VB_CLKDIV_DSER	Clock divider for De-serialiser
VB_CLKDIV_SUB_CLOCKS	Clock divider (by 2) for subtraction section of Comparator
VB_CLK_BY4_SUB_CLOCKS	Cascade of Divide by 2 clock divider
VB_CLOCKGEN	Clock generator for the flash
VB_CLOCKGEN_IDEAL	Ideal Clock generator for the flash (veriloga)
VB_CML_TO_CMOS_DATA	CML to CMOS converter for data input to thermometer to binary conversion
VB_CMOS_NAND_LCLOCKGEN	NAND gate used in clock generation for latches
VB_CMOS_NAND_SCLOCKGEN	NAND gate used in clock generation for input-reference subtraction section of comparators
VB_CMOS_TO_CML	Driver which converts the CMOS clock input to CML logic
VB_COMPARATOR	Unit comparator cell
VB_COMP_BUFFER	CML buffer used in comparator
VB_DAC_IDEAL	Ideal single point sampling DAC to test the flash
VB_DSERIALISER	De-serialiser which takes in the binary code from the thermometer to binary converter.
VB_DSER_CLOCKGEN	Clock generator to produce clocks for De-serialiser (4 clocks at $\frac{f_s}{4}$ separated by $\frac{1}{f_s}$)
VB_FLASH	Schematic of the comparator array with the reference ladder
VB_FLASH_CLK	Complete flash which includes the comparator array and the clock and bias generator
VB_FLASH_CLK_BIN	Cellview with the entire flash, thermometer to binary converter and de-serialiser
VB_FLASH_OUT_LEV_SHIFTER	Level shifter to reduce the output common mode of flash and increase it's driving capacity
VB_FLASH_REF_GEN	Reference level generator for flash.
VB_IREF_GEN	$\frac{V_{ref}}{R}$ type reference current generator to generate CML levels and flash reference levels
VB_LATCHA	First latch which is reset in the comparator
VB_LATCHB	Second latch in comparator which stores output of VB_LATCHA

VB_OPAMP	Two stage opamp used to generate the CML levels and flash reference levels
VB_SIG_GEN_IDEAL	Signal generator for preliminary testing. It generates test inputs for flash and clocks using veriloga blocks.
VB_SUBTRACTOR	Input reference subtraction stage for the comparator
VB_SUB_CLOCKGEN	Clock generator which generates clocks for input reference subtraction section.
VB_TDETECT	Transition detect logic to detect transition in bubble corrected thermometer code.
VB_TGATE	Transmission gate switches used in input-reference subtraction stage.
VB_THERM_TO_BIN	Thermometer to binary converter.
sp_ctdsm_ideal_all_delay	CTDSM stage with ideal loop filter and ideal single point sampling DAC and real flash
sp_dac_ideal	Single point sampling DAC used in testing the flash closed loop
sp_loopfilter_ideal_delay	Ideal dependent source based loop filter used in testing the flash closed loop.
sp_opamp	Ideal dependent source based opamp for the loop filter.

CHAPTER 9

Appendix B

Tips to reduce simulation time

In this appendix, we describe the steps we adopt to reduce the simulation time which is spent in achieving steady state solution for the clock generator. The testbenches named TEST_VB_CLOCKGEN and TEST_VB_LATCHES_CLOCKGEN are used for this purpose. Under any given process corner and instance (schematic or av_extracted), we first run the testbench TEST_VB_LATCHES_CLOCKGEN to obtain the steady state values for the nodes I1, I2 and bias in the circuit shown in figure 4.13. This testbench does not contain the laid out flash and hence takes much less time than the entire test bench to reach steady state. Following this, we use the testbench TEST_VB_CLOCKGEN to determine the steady state values for the node voltages of the first stage outputs of the opamp used in generating the CML clock voltage levels (figure 4.16). These change when the clock generator is loaded by the comparator array due to the average current drawn by the CML clock generator shown in the figure 4.12. In addition to this, the common mode feedback voltages of the two stages of the opamp are obtained from this testbench. This testbench does not contain the laid out thermometer to binary converter and deserialiser and hence is faster than the final flash testbench, TEST_CTDSM_DSER. These steady state voltages determined are fed in as initial conditions for the final testbench to reduce the time spent by the flash in the final testbench, TEST_CTDSM_DSER to reach the steady state.

REFERENCES

- [1] P. Ramalingam, “Design of a 16-bit continuous time delta-sigma modulator for digital audio,” Master’s thesis, IIT Madras, 2006.
- [2] R. Karthikeyan, “Analysis of clock jitter in delta-sigma modulators,” Master’s thesis, IIT Madras, 2007.
- [3] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*. John Wiley and Sons Inc.