The Hong Kong University of Science and Technology

EESM5000 Group Project

Evaluation of Digital Design Styles using a 16-Bit Adder as a Benchmark

Authors:
Arthur Song, Yassir Al-Refaie, Shrey Malik

Professor:
Dr. Amine Bermak

A Project Report submitted for the completion of the EESM5000 Group Project

December 2013

Contents

List of Figures

List of Tables

Abbreviations

RCA	**R**ipple **C**arry **A**dder
CLA	**C**arry **L**ookahead **A**dder
LCU	**L**ookahead **C**arry **U**nit

Chapter 1

Introduction

1.1 Goals and Motivation

The purpose of this project is to give insight into some of the more popular digital design paradigms available to IC designers. In order for this to be a fair assessment the same digital circuit will be analysed in each case. The requirement is to design a 16-bit adder in four design styles;

1. Complementary Logic
2. Dynamic Logic
3. Ratioed Logic
4. Pass-Transistor Logic

In each case, the worst case carry delay and sum delay will be extracted from both schematic and post layout simulations to give theoretical and practical timing parameters. The case used will be,

A=`0xFFFF`

B=`0x0000`

$$C_{in}0 \xrightarrow[\text{at time, T}]{} 1$$

This will allow the final C_{OUT} and Sum_{15} to be a function of the initial C_{IN} changing from low to high. By comparing the point when C_{IN} raises to 0.5*VDD and C_{OUT} and Sum_{15} also reaches to that level, we can deduce the Carry and Sum delays.

The other parameter that will be extracted is the PMOS and NMOS counts, the Total MOS count and the Area of the final design. This will be extracted from the Virtuoso tool manually.

1.2 Adder Design

To compare the varying design styles, the adder implemented will be a hybrid between the CLA and RCA adder topologies. The idea is to strike a balance between both area and speed of the design rather than emphasising either. Hierarchacly the design will be segmented into 4-bit adder chunks, each with an LCU. Each 4-bit adder itself will be made of three 1-bit full adders and a single 1-bit partial adder, each generating P and G logic for the LCU. The LCU will not be a full LCU that generates C0:C3, but will only generate the final C3 in parallel with the the inputs becoming available. The purpose is to allow the subsequent 4-bit stages to have the carries ready for the sum generation.

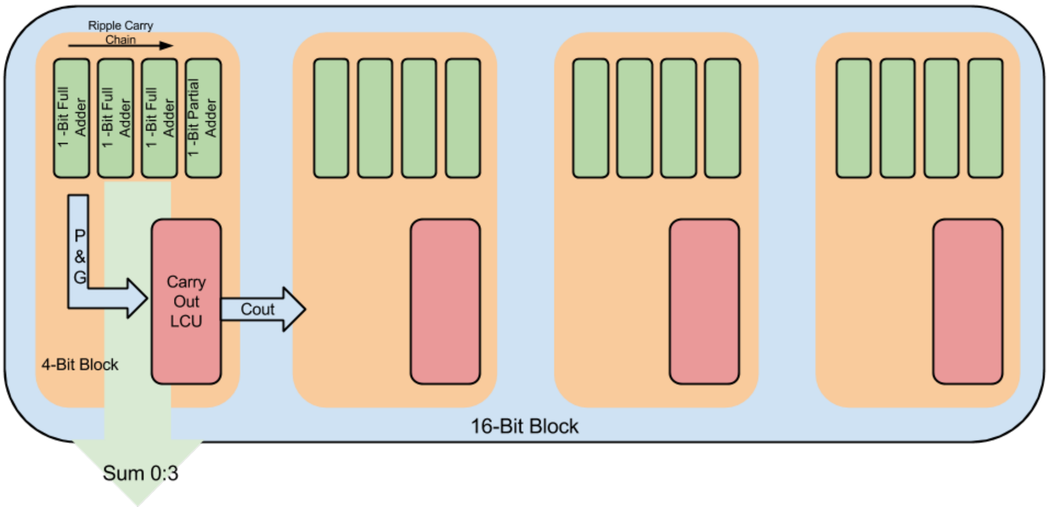

FIGURE 1.1: Block Diagram of Adder Design

This means that the overall delay of an N-bit adder, where N is any multiple of four no longer scales linearly as with the RCA. Instead the increase in delay should theoretically be equal to

$$T_{C_N} = (N-1) \times T_{LCU} + 3 \times T_{FA} + T_{PA} \tag{1.1}$$

The precise implementation of this topology varies with each design style in order to take advantage of the benefits and to cover up the weakness of the particular style. As such the exact implementation will be elaborated where appropriate.

Chapter 2

Complementary Logic

2.1 Concept

Complementary Logic is a design style from the static family. The idea is to form two networks, a PUN from PMOS transistors which provides a route for the output to VDD, and a PDN from NMOS transistors which provides the output with a path to ground. These two networks would be formed from complimentary logic so that at any time, only one may provide a path for the output. In that sense, because the output always has at any time a path to either VDD or GND and will not change unless the inputs vary, the logic is considered static.

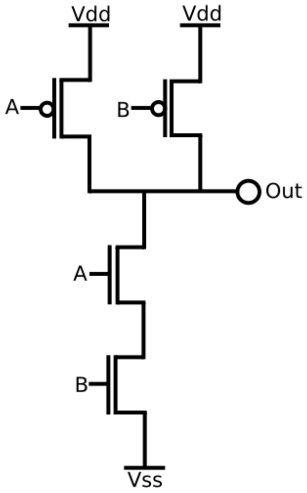

Figure 2.1: Schematic for Static CMOS NAND

CMOS Logic has some benefits that make it a very popular design paradigm:

1. Very low static power consumption
2. High noise immunity
3. Low waste heat
4. Easy to implement

but also has some drawbacks:

1. Large transistor count (2*N)
2. As a result, one stage at a minimum needs to drive two gates in the next stage (High input capacitance)
3. Consequently, this reduces the speed of critical paths

As a result of this many considerations where made to reduce the number of transistors in the overall design.

2.2 Schematic

2.2.1 Design

As mentioned, CMOS has the largest transistor count of the design styles as it requires a minimum of 2*N, where N is the number of inputs in a function. Owing to this, it was natural that the focus in this design style was in reducing the overall area as this is the limiting factor of the CMOS family of logic. Considerations were also made towards the Carry critical path to help further enhance the impact of the LCU on the carry path. Furthermore, due to the fact that the remaining design styles are based on the fundamental design defined here, the most detail will be presented here.

2.2.1.1 XOR Gate

The first change is in the nature of the XOR function implementation. Rather than use the function $\overline{A}.B+A.\overline{B}$ which requires an inverter for both inputs, it was implemented as a NOR that is passed through another function to generate A⊕B. Therefore,

$$A\oplus B=\overline{\overline{A+B}+A.B}$$

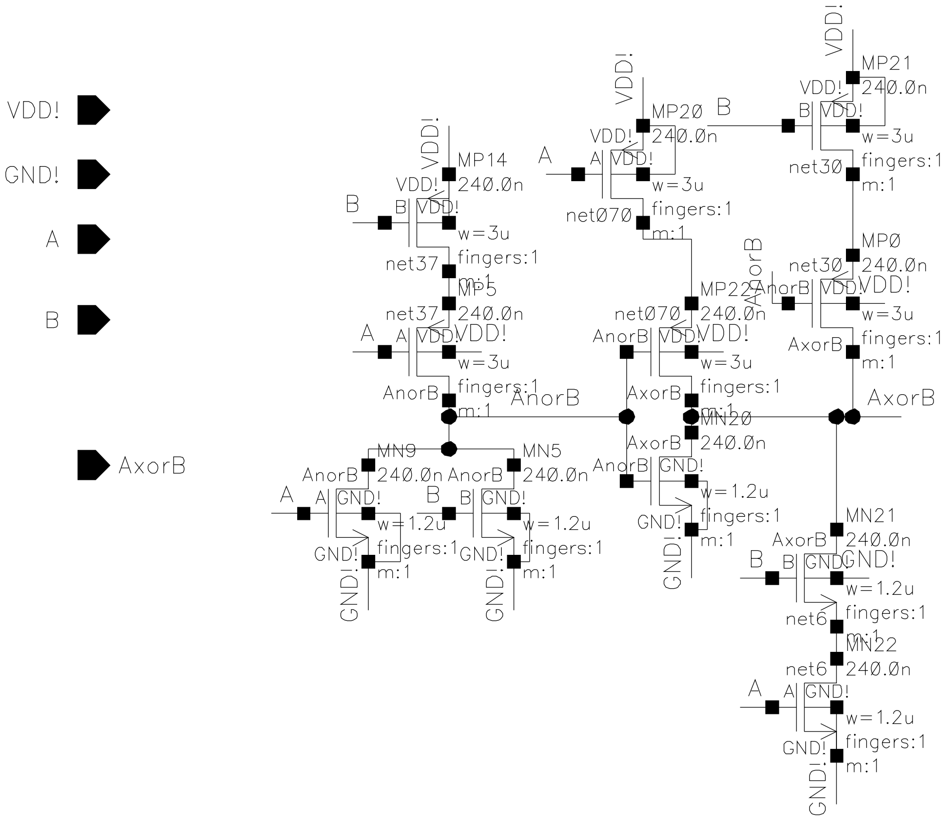

FIGURE 2.2: XOR Schematic with Redundant Gate added for Eulers Path

Another observation of the two input XOR gate is that the output is correct if both the inputs are inverted, therefore using inverted inputs allows the correct sum to be generated if we make sum a function of $\overline{\text{P}}$and $\overline{\text{Cin}}$. Owing to the design of the LCU which will be outlined in the next section, $\overline{\text{Cin}}$is present in all stages since we output $\overline{\text{Cout}}$ from the LCU, all that is required is the input Carry at the LSB to be inverted.

2.2.1.2 Four-Bit LCU

The first observation was to have the entire 16 bit chain operate in an inverted state. By using DEMORGAN'S LAW the LCU PDN and PUN were swapped, and all the inputs inverted. This means that only $\overline{\mathrm{P}}$ and $\overline{\mathrm{G}}$ are required, eliminating at least the inverter for generating G. The P inverter isn't saved since the output of the above function is A⊕B whilst we require $\overline{\mathrm{A}\oplus\mathrm{B}}$. The LCU therefore generates the fourth-bit Carry which we then invert to generate $\overline{\mathrm{Cout}}$ to feed into the input of the next 4-bit adder block.

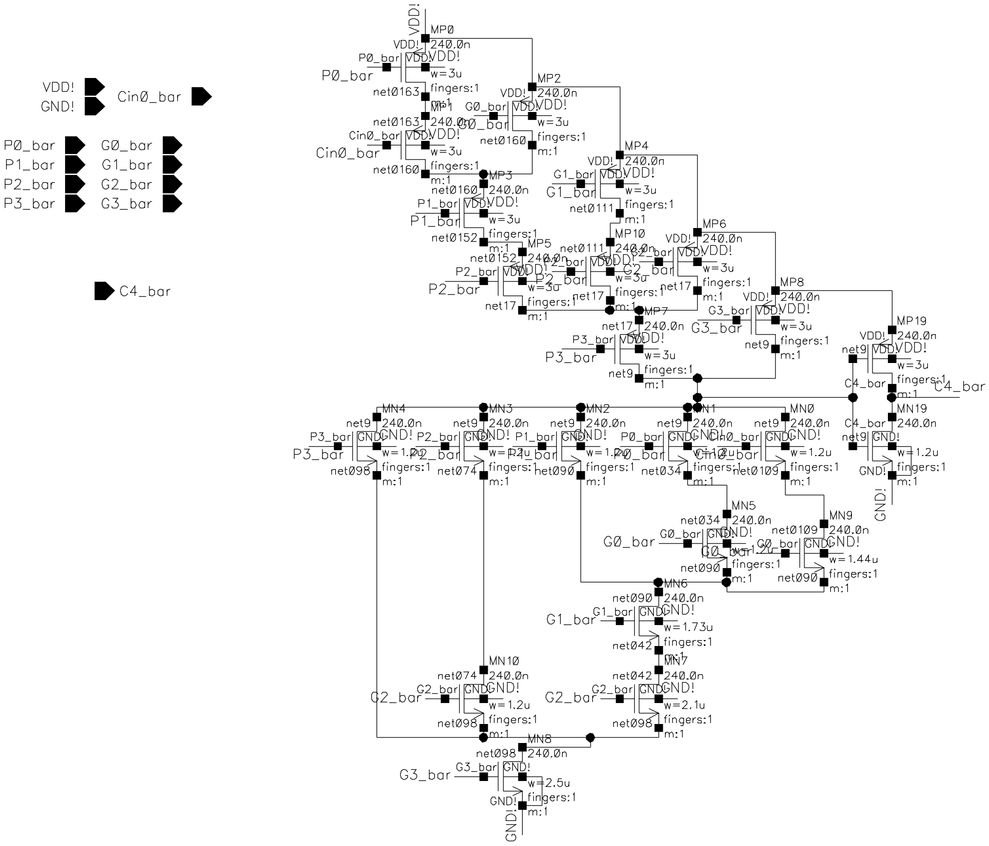

FIGURE 2.3: LCU for Generating the Fourth-Bit Carry, with Swapped PDN and PUN, and Inverted Inputs.

This allowed the carry generation to become very quick with only two inverters added to the input Carry and the output Carry.

2.2.1.3 4-Bit Adder Block

To achieve a design that was manageable and simple to verify, the design was constructed hierarchacly as a P cell, a G cell, the Sum generation, Ripple carry generation and the LCU. From this, two blocks were defined. The FULL ADDER which generates all the signals, and the PARTIAL ADDER which generates all save for the Cout signal since the fourth adder block does not need to propogate a carry out.

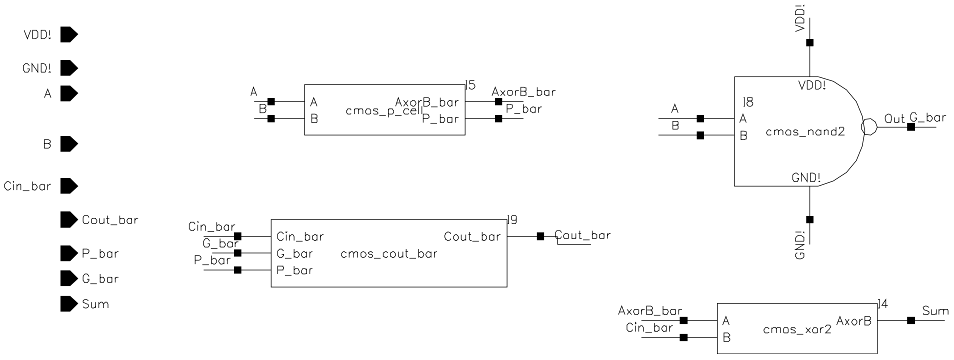

Figure 2.4: 1-Bit CMOS Full Adder Block

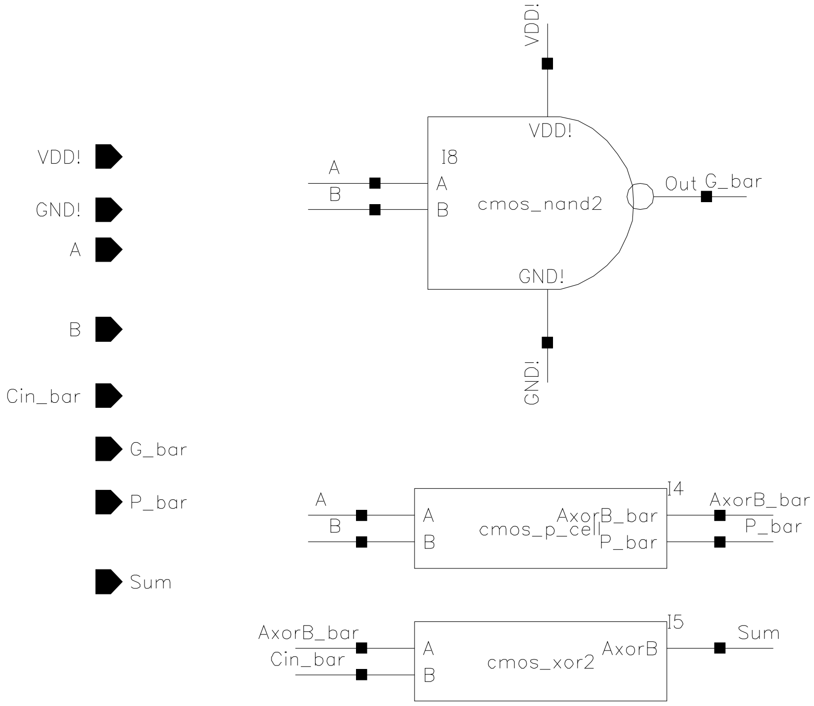

Figure 2.5: 1-Bit CMOS Partial Adder Block

These were combined in a 4-bit slice, with the LCU to create the fundamental block, that would be iterated four times to create a 16-bit system.

2.2.1.4 16-Bit Adder Block

The 16-bit adder is simply four iterations of the four bit block, with an inverter to the input Carry and another to the output carry. The initial idea was to use ripple carry chain in the final block to save the LCU since we need to wait for the final sum in any case. However, at layout, it apeared that removing the LCU left empty space and so it was more effective to utilise that die area to generate the final carry out before the

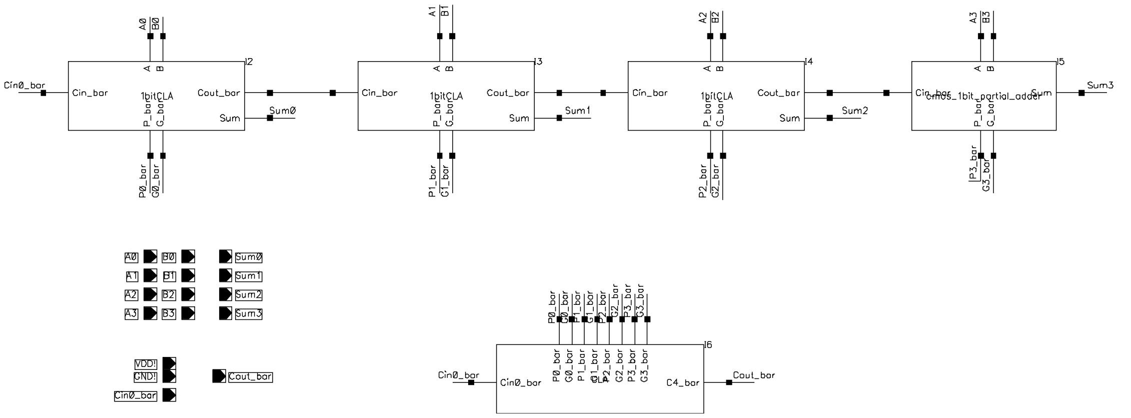

FIGURE 2.6: 4-Bit CMOS Adder Block with LCU

Sum is ready, the justification is that it allows systems to detect overflow even before the sum is ready and so can respond promptly.

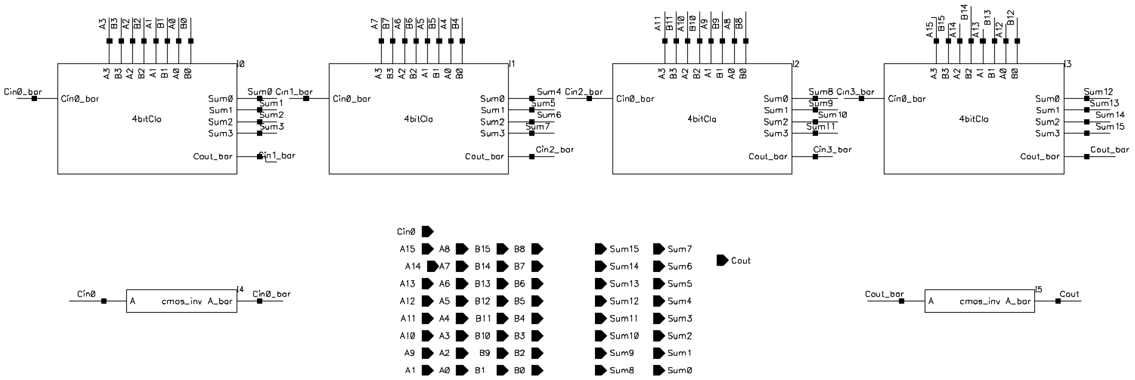

FIGURE 2.7: 16-Bit CMOS Adder

2.3 Layout

Hierarchicaly the layout was constructed in the same way as the schematic. This was to ensure that the LVS would be manageable and thus improve verification speeds. Considerations where made so that constructing different cells, and using separate diffusions in a single cell had little difference in impact to the area.

One trick to reduce the effective cell size was to reduce the VDD! and GND! extension beyound the Diffusions. Usually, as described in the lab, the VDD is takes to the edge of the N Well and so the GND must extend to this distance, however this is unnecessary and the VDD and GND can be brought to be flush with the diffusions as seen in Figure 2.8

so that when combining two cells together, the N wells will overlap, whilst the Oxides have the necessary 10λ separation.

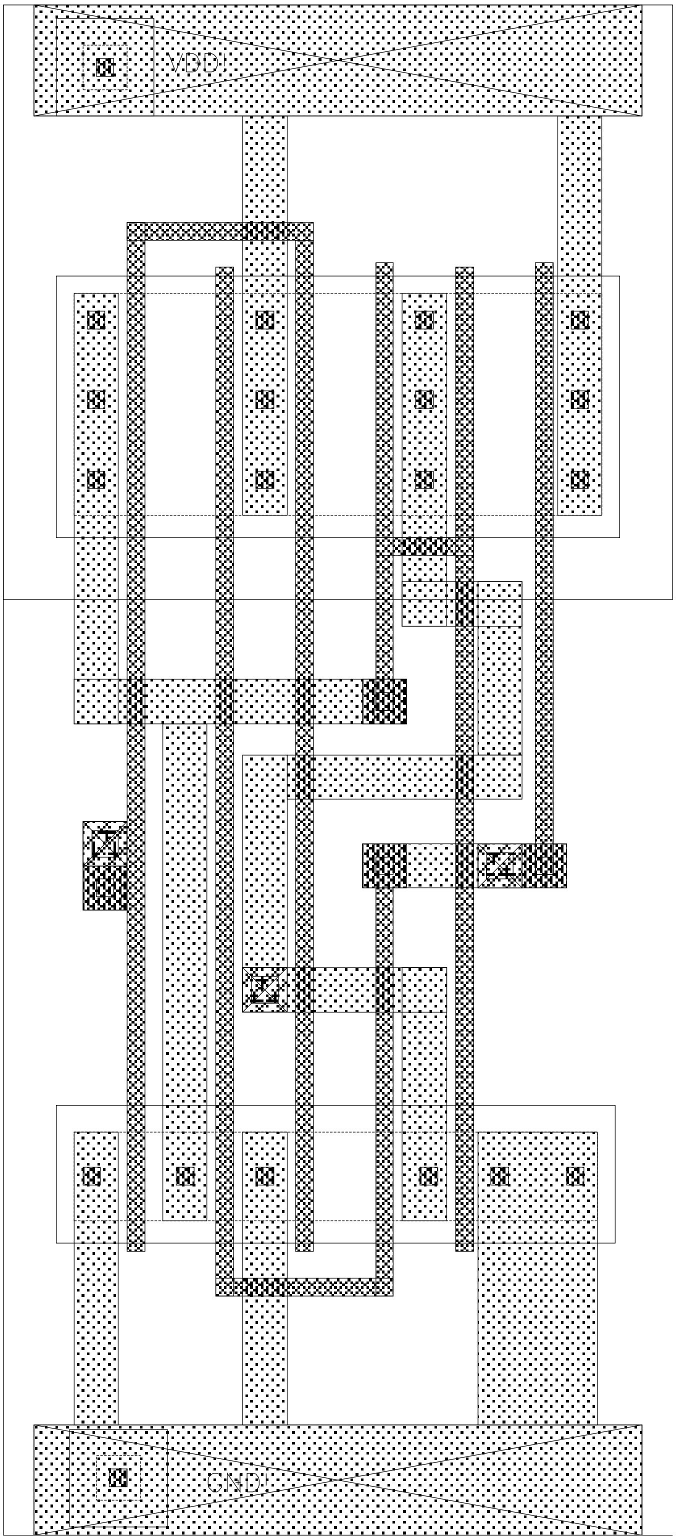

Figure 2.8: Layout of CMOS 2-Input XOR demonstrating the VDD and GND Rail Underextension

2.3.1 Design

For the design of efficient layouts, there were certain tricks which were devised to allow single diffusions to be used even when a shared Eulers Path could not be found, furthermore efforts where made to ensure the design was as close to a regular square as possible to maximise the die area used and used as few metal layers as could be achieved to reduce the costs. In the case of the LCU, as can be seen in the shematic there is not a single path for both the PDN and the PUN.

Therefore for the PUN, the best route that could be found, required the P2 to be expanded so that a single loop could be created for the PUN. The route found was **Inv-G3-P3-G2-G2-P2-P2-P1-G0-P0-Cin**. For the PUN, a similar strategy was used where G2 and G0 where expanded out to provide a single loop, **Inv-G3-P3-P2-G2-G2-G1-P1-P0-G0-G0-Cin**.

PUN	Inv	G3	P3	G2	G1	P2	P2	P1	G0	P0		C0
PDN	Inv	G3	P3	P2	G2	G2	G1	P1	P0	G0	G0	C0

From this we can see that there are five transistors which share the same gate, however, using this route, the mismatch between the two networks is solveable by crossing the gates to achieve what is effectively two matching routes. Due to it's complexity and the fact that it was something that needed to be visually verified first, a stick diagram was derived.

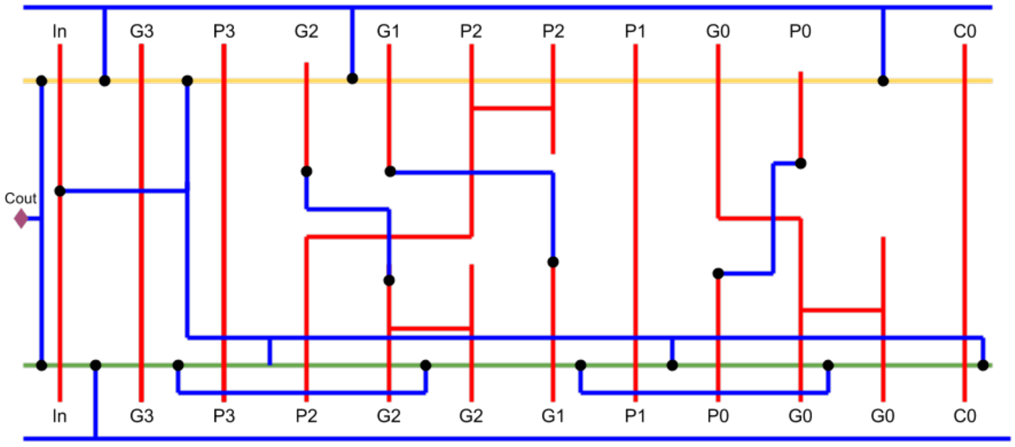

Figure 2.9: Stick Diagram for CMOS LCU, demonstrating the crossing of Gates

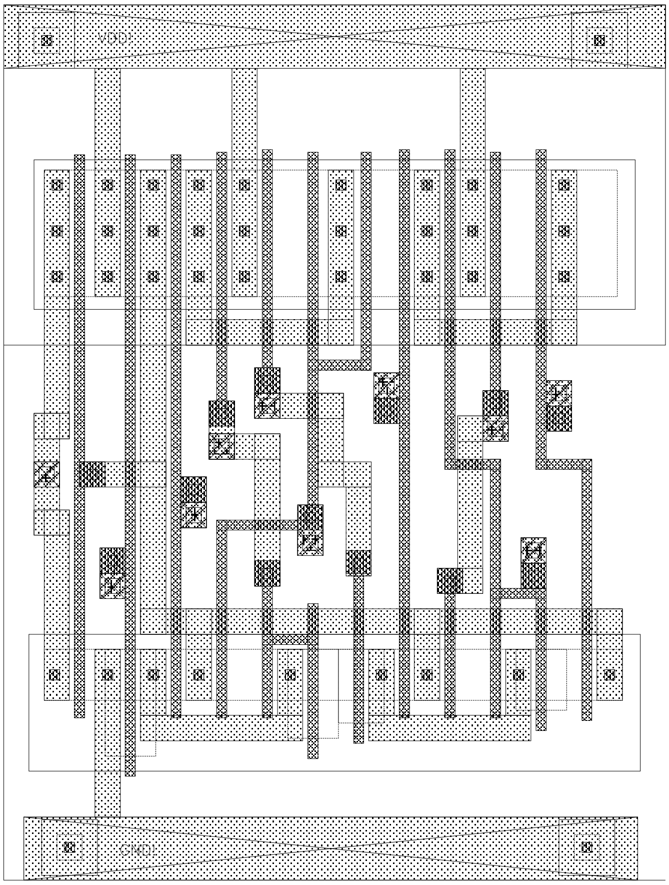

Figure 2.10: The Final Layout for the CMOS LCU

Furthermore, with regards to the LCU some sizing was considered for the worst case paths to ensure the output node can discharge quick enough to allow the rise time of the final Carry to be as quick as possible.

2.3.1.1 16-Bit Adder Block

The 4-bit adder block was devised with the LCU sitting in the center of two bits above and below it. The idea was to utilise the space created from the extra length that the 2-bit adder extends over the LCU for routing and to create a more regular shape as opposed to a long rectangular array.

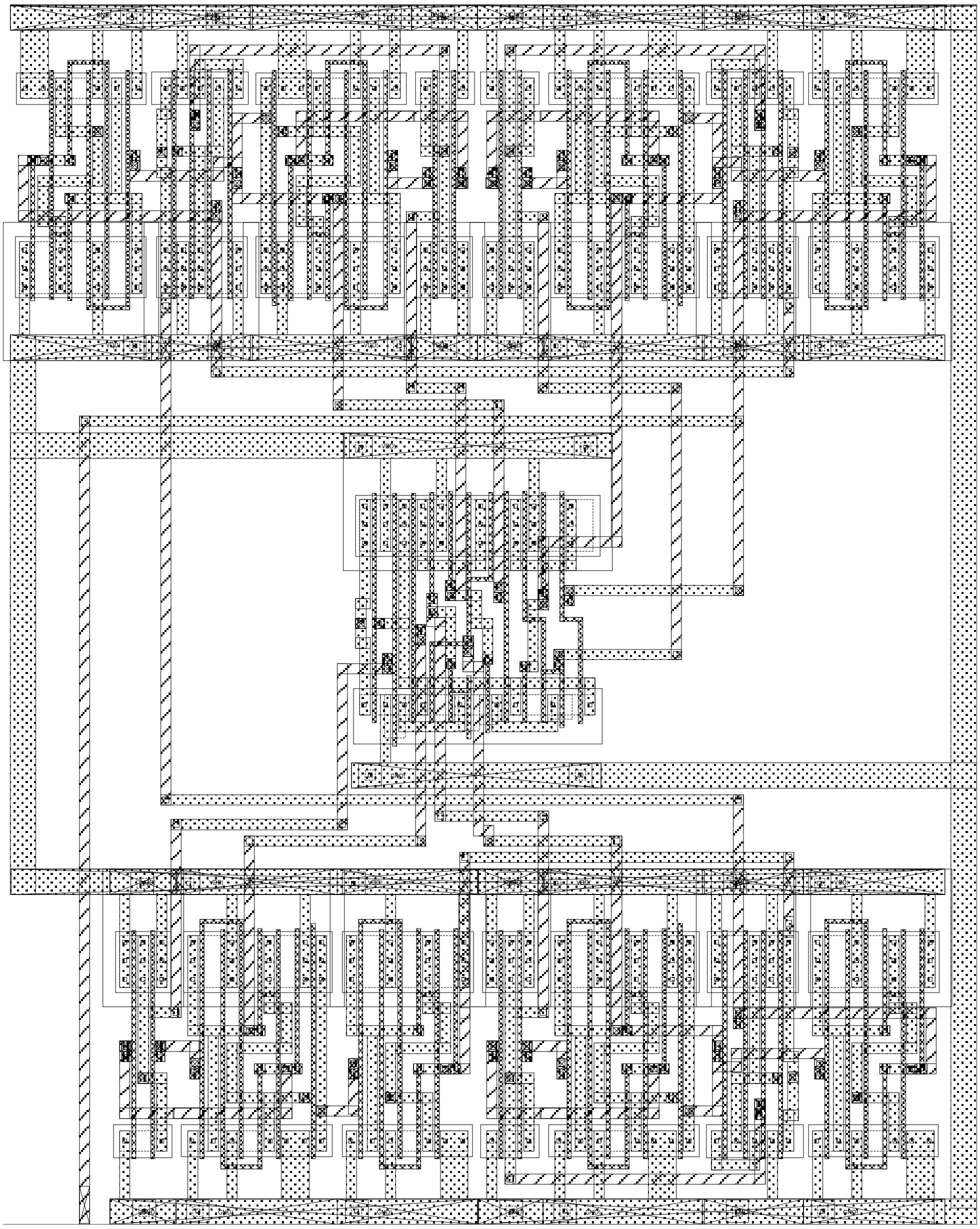

Figure 2.11: CMOS 4-bit Layout with LCU in the Center

This was then combined in a 2×2 array of the 4bit slice above with the two Cin and Cout Inverters embedded into the space available in the 4-bit block.

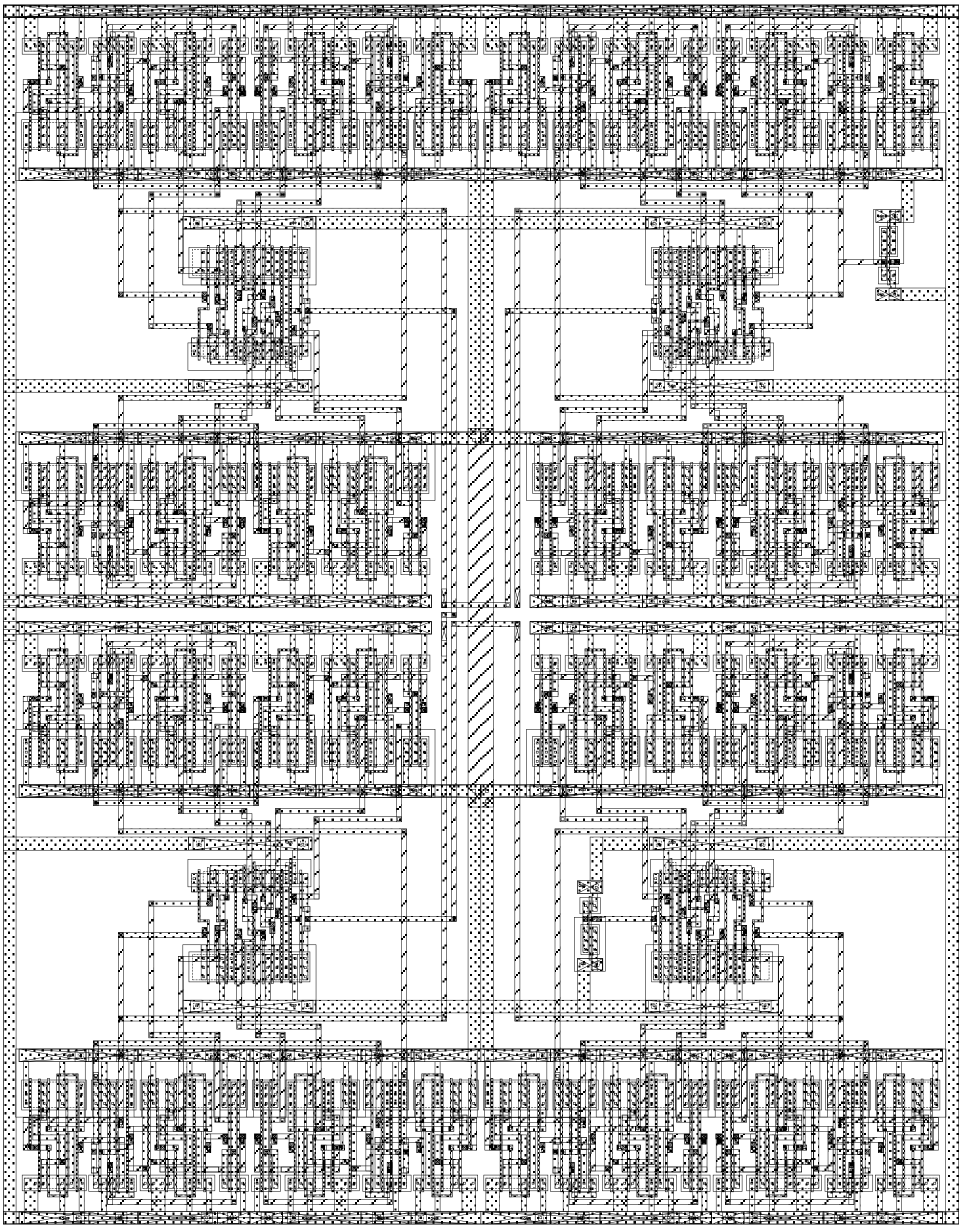

FIGURE 2.12: CMOS 16-bit Layout with Inverters

From this layout, it is clear why the LCU is kept in the final block. Since when the the final LCU is removed, we are left with a lot of empty space, which whilst reduces the transistor count, considering that the area is the length and height of the cell, there would be no difference in the apparant area. Therefore the performance improvement was kept and the LCU retained inside the final cell.

2.3.2 Simulation

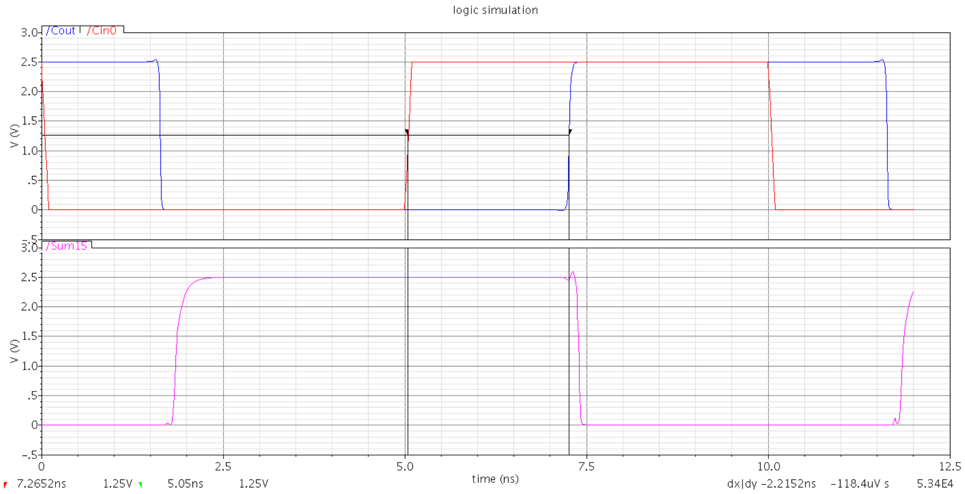

FIGURE 2.13: CMOS 16-bit Worst Case Simulation

From the waveform we can see a delay of 2.21ns which is reasonable considering the topology should only consider a 4-bit Full Adder delay, plus the delay of four LCU blocks.

2.4 Results and Conclusion

Cout Delay (ns)	2.21
Length (um)	143.58
Width (um)	112.44
Area (m^2)	$11.61*10^{-8}$
Power	1.89mW
NMOS Transistors	306
PMOS Transistors	334
Total Transistors	640

From the results above, it can be seen that from schematic to layout, there was approximately a doubling in the delay. However, this is still very good in terms of performance since the CMOS is the slowest of the design styles due to the large gate capacitance that must be driven between stages. With regards to area, if we compare to one of the more efficient CMOS Designs which is a 28 Transistor, 1-bit Ripple Carry Adder which has 448 transistors across 16-bits, the transistor count is roughly 50% larger than the ripple carry which is acceptable considering the performance.

Chapter 3

Ratioed NMOS Logic

3.1 Concept

Ratioed Logic is a static logic style that aims to reduce the numbers of transistors that are used by eliminating either the PDN or the PUN from the complementary logic, and replacing the network with a sized NMOS or PMOS that is always driving the output. The idea is that when both the single transistor and the network is on, a voltage division occurs with the output voltage being the amount of voltage dropped across the network. In the case of this implementation, the PDN is used with a sized PMOS transistor.

The benefits of Ratioed Logic are:

- Low transistor count relative to CMOS
- Only has to drive a single capacitance for stage.
- Easy to implement

Whilst the drawbacks are:

- Large static power consumption due to a path from VDD to GND when the PDN is on.
- Low noise immunity

- The output swing and consequently the noise margin is affected by the PMOS size - thus logic is Ratioed.
- Rise time is affected by the PMOS size.

As we can see there is a conflicting parameter requirement. The swing is enhanced by reducing the size of the PMOS pull up, whilst the rise time is improved by increasing the size. As such it is important to strike a balance between swing and speed.

3.2 Schematic

Owing to the conflicting requirements of Ratioed logic, it was necessary to consider whether the swing or the delay was more important. In this case, it was decided that since the topology itself is faster than a conventional ripple carry, then the swing would be optimised to make sure the noise margin was still high.

3.2.1 Design

In the case of this design, schematically they are replicas of the CMOS logic but with the PUN replaced by a single PMOS with its gate connected to VDD and sized to 1.2u to increase the resistance of the pull-up, therefore allowing the voltage to reach closer to GND.

3.2.1.1 1-Bit Adder Block

For the Ratioed logic design style, since the layout is significantly more simple due to the reduced transistor count, it was decided to integrate the entire design into single cell as opposed to being composed of sub-cells.

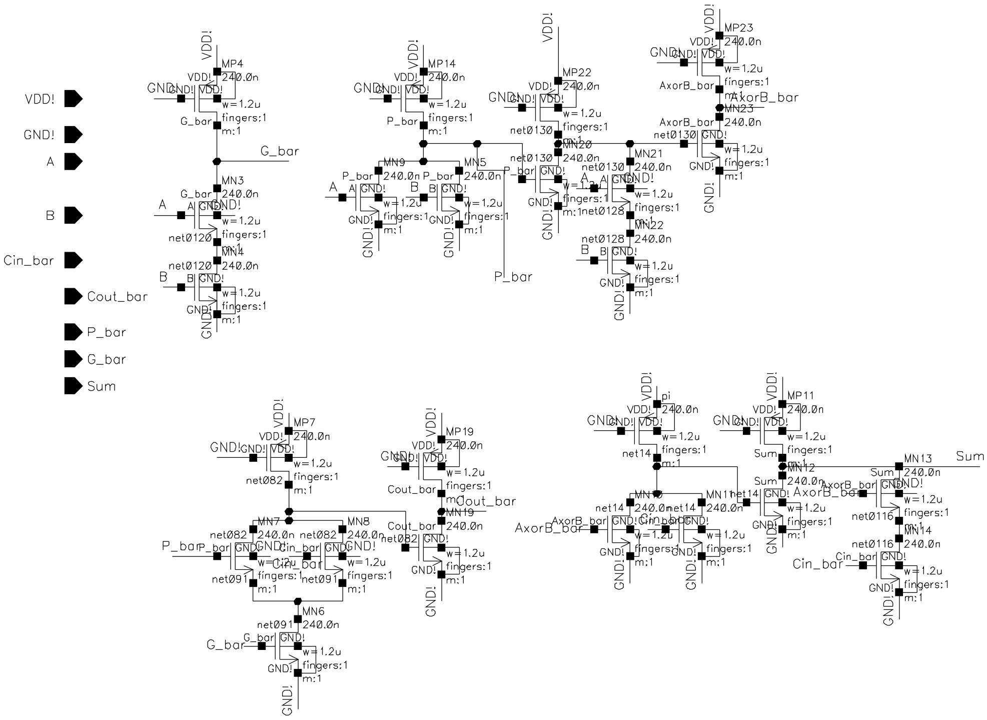

Figure 3.1: Ratioed NMOS 1-bit Adder Cell with the Sized PMOS Pull-Up networks

3.2.1.2 LCU

For the LCU the final output had to be buffered after post-layout simulations were run since it appeared that the swing was weak on the LCU output and so subsequent cells after the first 4-bit block were receiving a low signal regardless of the logic. As such buffering was included to restore the swing between the LCU blocks. Whilst this added delays, it provided correct behaviour.

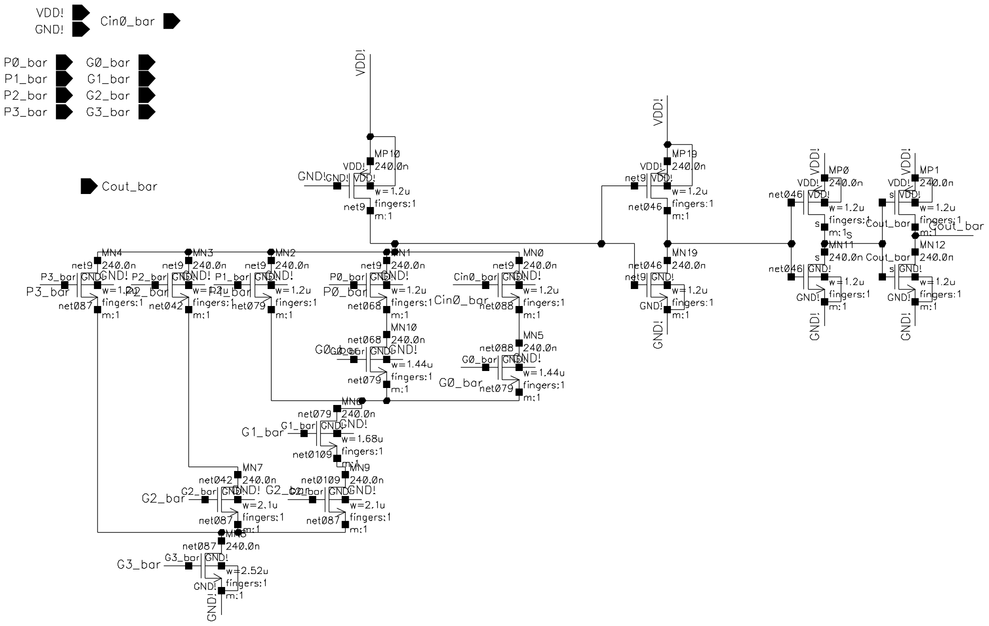

Figure 3.2: Ratioed LCU with the Two Inverters Buffering the Output

3.2.1.3 16-bit Adder Block

The composition of the 16-bit adder cell was the same as with regards to the CMOS, and essentially is a 2×2 array of the 1-bit adder cell, with an LCU placed in the center. This cell was arranged in a 2×2 array to provide the complete 16-bit addition. Since the appearance is identical to the CMOS at this level, the schematics will not be provided for the sake of succinctness.

3.3 Layout

For the layout, the same hierarchy was maintained as the schematic as this allowed LVS to be conducted in a manageable way. Because of the nature of ratioed simply being

CMOS without the PUN, the final product for the 4-bit and 16-bit layout was almost a replica of the CMOS but with a smaller cell height.

3.3.1 Design

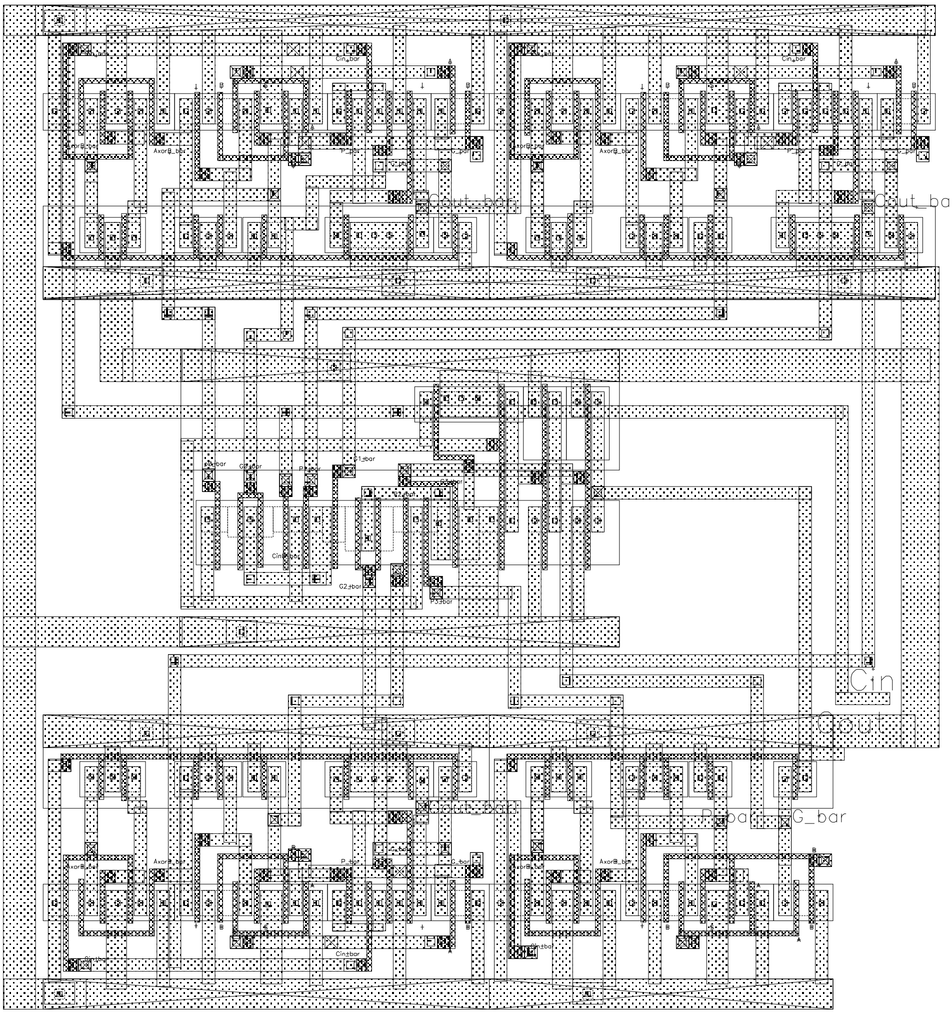

Figure 3.3: Ratioed 4-bit Adder Cell with LCU in the Center

As can be seen, there is a striking similarity and this is expected. The same can be seen with the 16-bit,

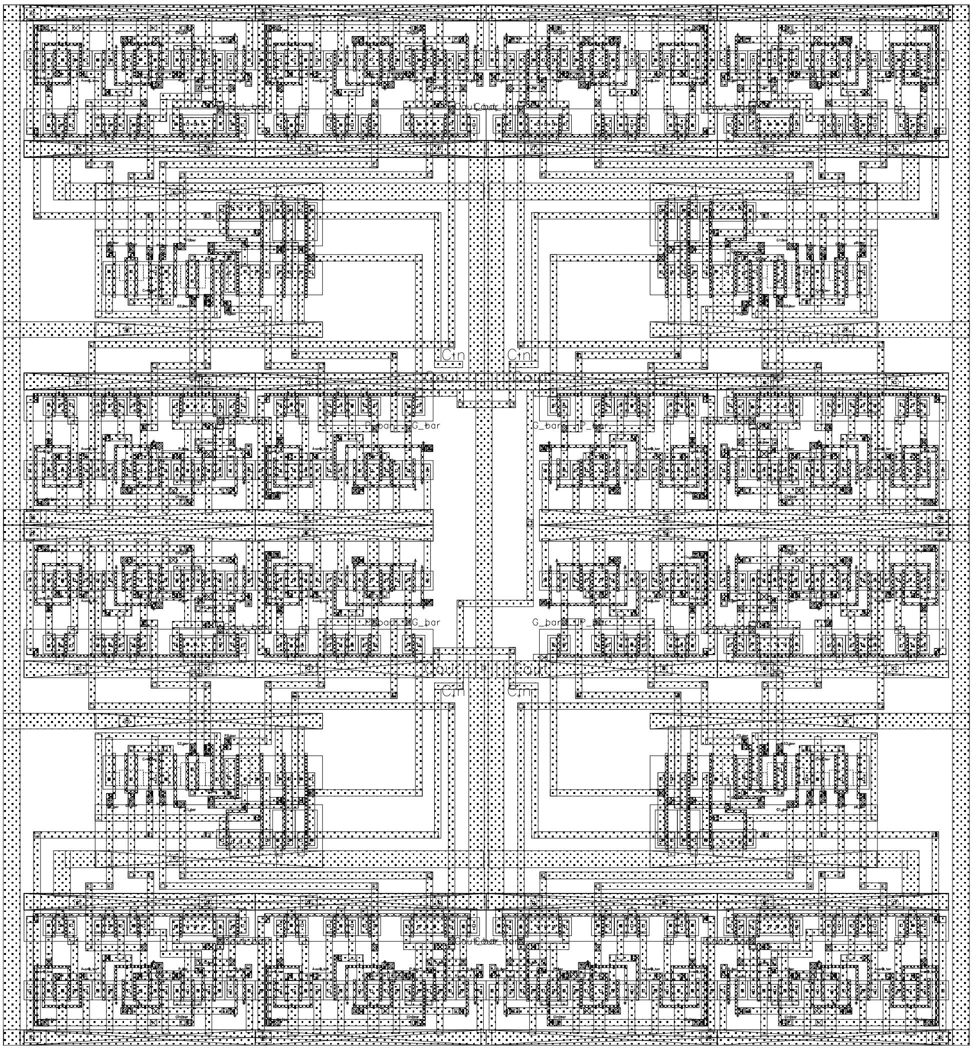

Figure 3.4: Ratioed 16-bit Adder

3.3.2 Simulation

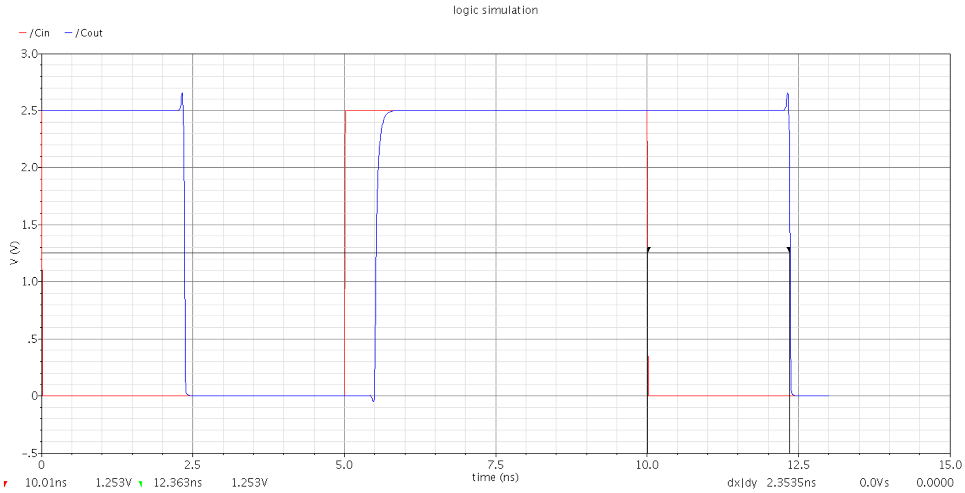

Figure 3.5: Ratioed 16-bit Adder Worst Case

From the above wave, the delay can be seen to be 2.35ns.

3.4 Results and Conclusion

Cout Delay (ns)	2.35
Length (um)	88.56
Width (um)	95.4
Area (m^2)	$8.45*10^{-9}$
Power	57mW
NMOS Transistors	312
PMOS Transistors	136
Total Transistors	448

With regards to the performance, these values are satisfactory, save for the power, which seems fairly high. However this is one of the disadvantages of the Ratioed logic since there is static current due to the path from VDD to GND.

Chapter 4

Dynamic Logic

4.1 Concept

Dynamic logic is different to the static family in that it depends on capacitance on the output to store charge. It works much like ratioed but instaed of an always on, sized network, it instead uses a clocked network to precharge the node to either GND or VDD. In this case, it is a PMOS that precharges the output node to VDD, with a PDN that pulls it down to GND if its condition is satisfied. It requires a periodic clock to refresh the value of the output node so that the inherent leakage of capacitors is mitigated and the value is valid.

Dynamic logic has some benefits:

1. Small transistor count due to the removal of the PUN
2. Only has to drive one capacitance on the next stage
3. Consequently, it isfast

The drawbacks however are;

1. Prone to the effects of charge sharing
2. Complex in larger designs due to clock routing and clock skew.
3. Cannot directly drive the next stage due to metastability.

4.2 Schematic

4.2.1 Design

One big consideration was to take advantage of the abilit to precharge the carry nodes in the LCU and therefore be able to generate all the carries quicker than with the the previous design styles. As such, it was necessary to revert back from the inverted logic blocks back to the non-inverting logic. This also helps with dynamic since the simplest mechanism for chaining stages is by using a static inverter to provide a stable output for the next stage that isolates the previous stages precharge phase. This is the idea behind Domino Logic.

As such, the LCU was redesigned to allow the nodes to be accessible and sized so that the delay is reduced for the worst case.

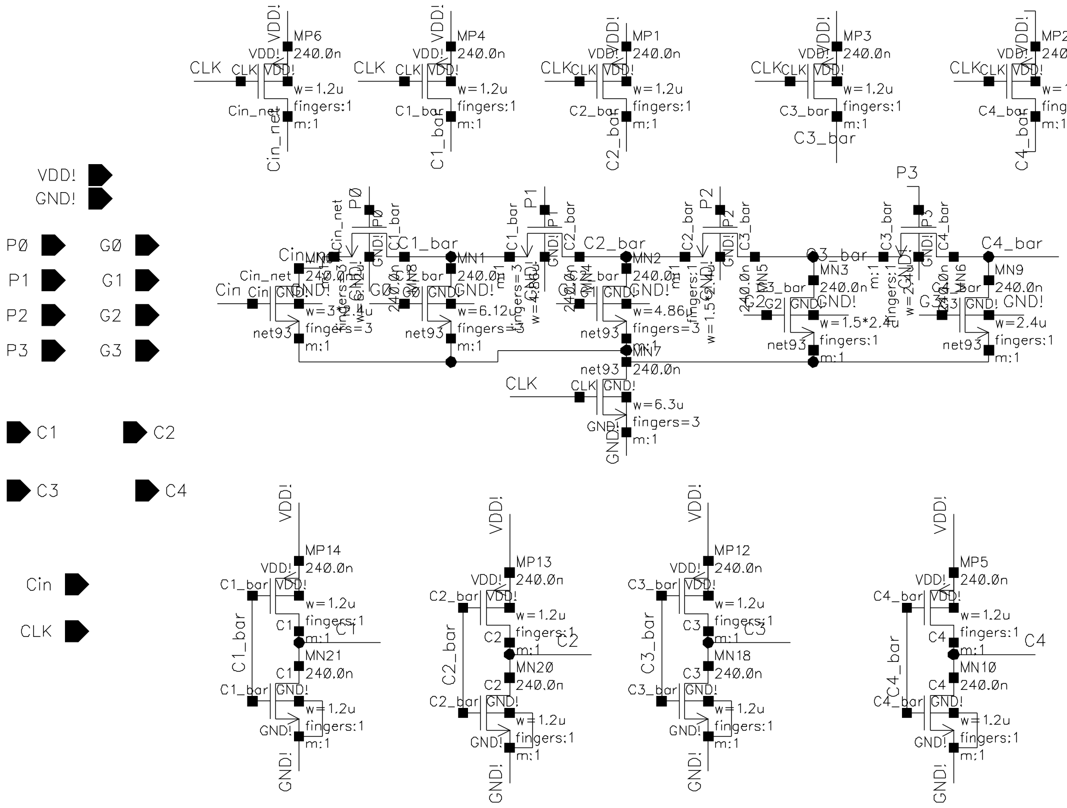

Figure 4.1: Dynamic LCU with Precharging of Nodes

Another point that was deduced was that, for dynamic it was better that P be generated from the function A+B rather than A⊕B since generating sum as a function of P created a delay that was unacceptable at each stage. Instead, a three input XOR gate was created to allow the sum to be created as soon as Cin was valid, with the function A⊕B⊕Cin. Whilst this added some transistors, it meant the performance was improved.

The final 1-bit has a split clock, with the main clock driving P and G and the shifted clock driving Sum (to give time for Cin to become valid). The shifted clock is also output to the next stage as clk_2. The main clock is also used to drive all of the LCU blocks across the entire design.

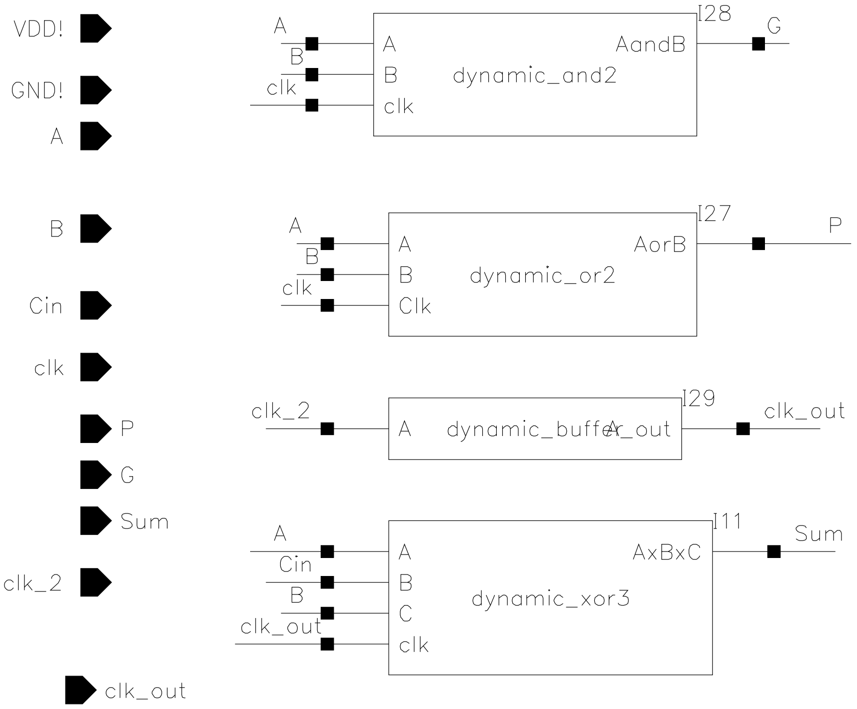

FIGURE 4.2: Dynamic 1-Bit Cell with Split Clock

The Four bit therefore is the same as CMOS but with the only consideration being the propogation of the delayed clock

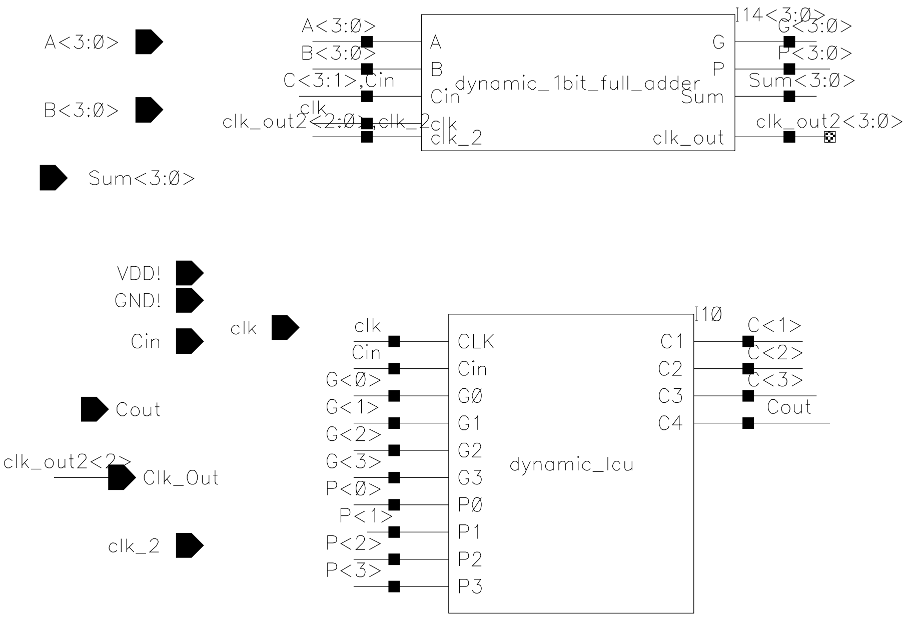

Figure 4.3: Dynamic 4-Bit Cell with Delayed Clock Distribution

And consequently for 16-bit,

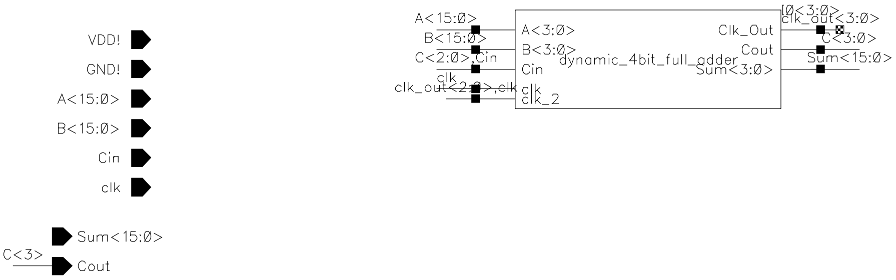

Figure 4.4: Dynamic 16-Bit Cell with Delayed Clock Distribution

4.3 Layout

4.3.1 Design

The layout for dynamic had some interesting strategies to maintain a small area. First of all, owing to the reduction in PMOS transistors, there was a lot of space in the top half of the cell, epsecially for the XOR3 cell. As a result, it was decided to try and embed the three input inverters into this area. This required an N-Well that was shaped to only include the PMOS transistors. The final result was successful and showed fair performance in final simulation.

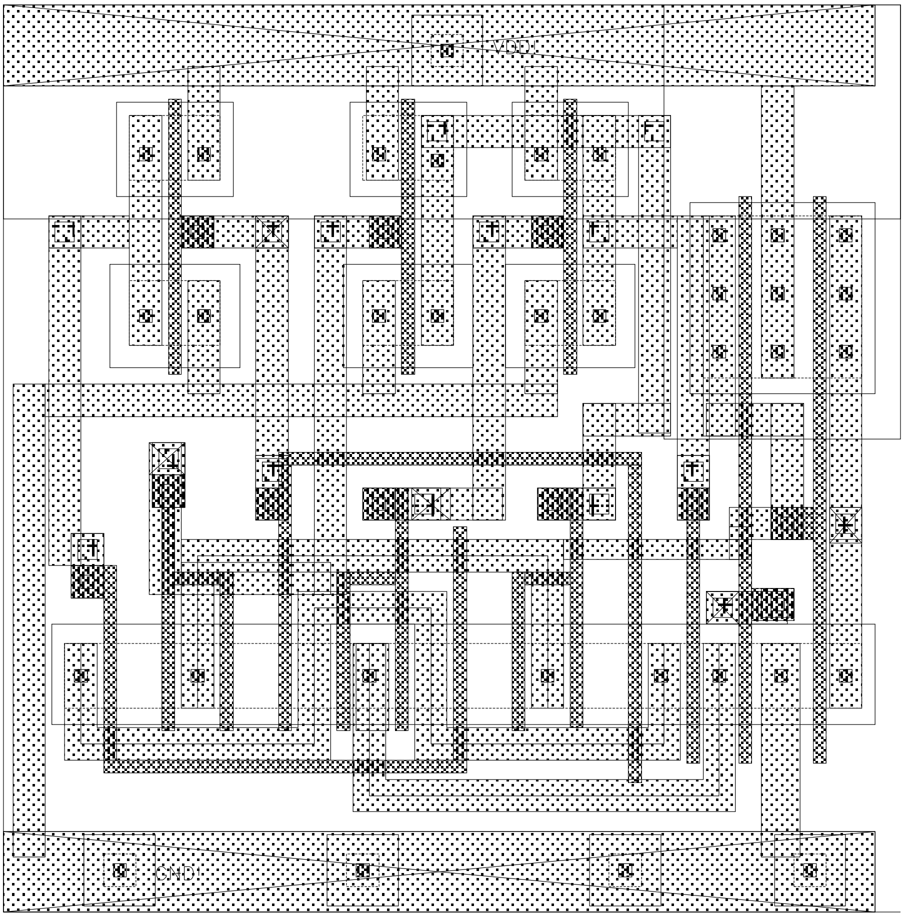

Figure 4.5: Dynamic Three-Input XOR Layout with Three Embedded Inverters

The other area of interest is the LCU and the combination of the four inverters on either end of the LCU structure, with the transistor sizing in the NMOS chain.

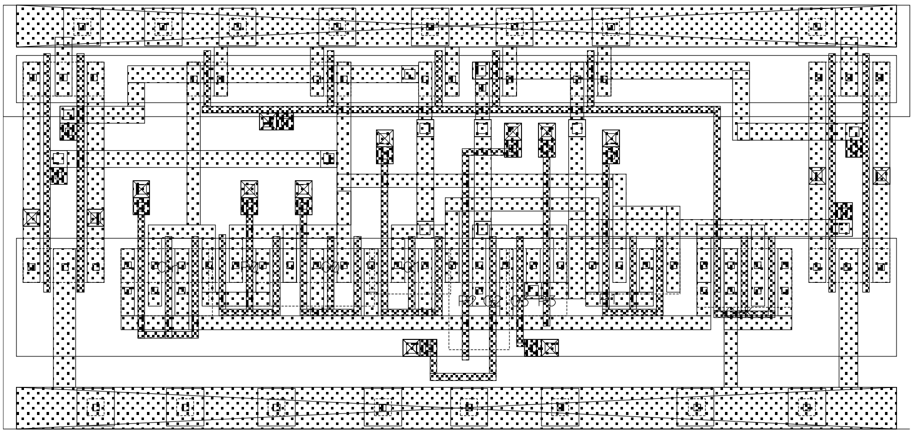

Figure 4.6: Dynamic LCU with Four Prechargers, Four Inverters and Sized Transistor Chain

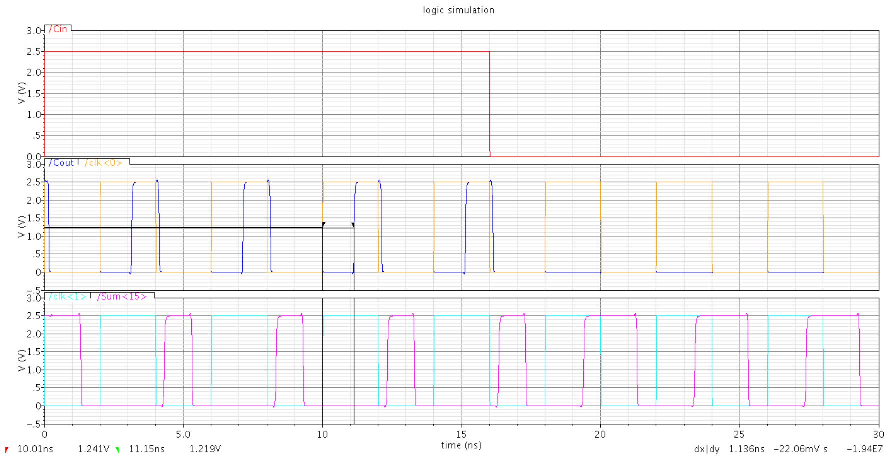

FIGURE 4.7: Dynamic 16-bit Worst Case Simulation

4.3.2 Simulation

The carry delay from the above simulation is 1.136ns, whilst there are many glitches due to the clock becoming too delayed in the later stages, they are only present on the low phase of the clock and so will not be sampled by the system.

4.4 Results and Conclusion

Cout Delay (ns)	1.136
Length (um)	128.28
Width (um)	124.5
Area (m^2)	$1.60*10^{-8}$
Power	4.68mW at 250MHz
NMOS Transistors	936
PMOS Transistors	388
Total Transistors	1324

Another consideration is the operating frequency. For this simulation, the system was operating at a clock frequency of 250MHz. Overall the performance is satisfactory considering the capacitance that is being driven when trying to generate the all the carries from a single LCU network. The NMOS transistor count also considers transistor folding to be equivalent to N transistors in parallel and so a 3 finger NMOS is considered as 3 NMOS transistors in the table. Whilst the transistor count is larger than the CMOS the overall area is reduced due to the smaller size of the PMOS transistors.

Chapter 5

Mirror Logic

5.1 Concept

Mirror Logic is a form of Static Logic that aims to exploit the symmetry that is present in the logic for a Full Adder.

A	B	Ci	S	Co	Carry Status
0	0	0	0	0	Delete
0	0	1	1	0	Delete
0	1	0	1	0	Propagate
0	1	1	0	1	Propagate
1	0	0	1	0	Propagate
1	0	1	0	1	Propagate
1	1	0	0	1	Generate
1	1	1	1	1	Generate

From this we can see that Co follows Ci as long as the function A⊕B is true, otherwise if $\overline{A}.\overline{B}.$ then the Carry is killed and Co is always low, else, when A.B, Cout is created and Cout is always high.

From this, the schematic becomes symmetrical across the X-Axis as opposed to being complementary.

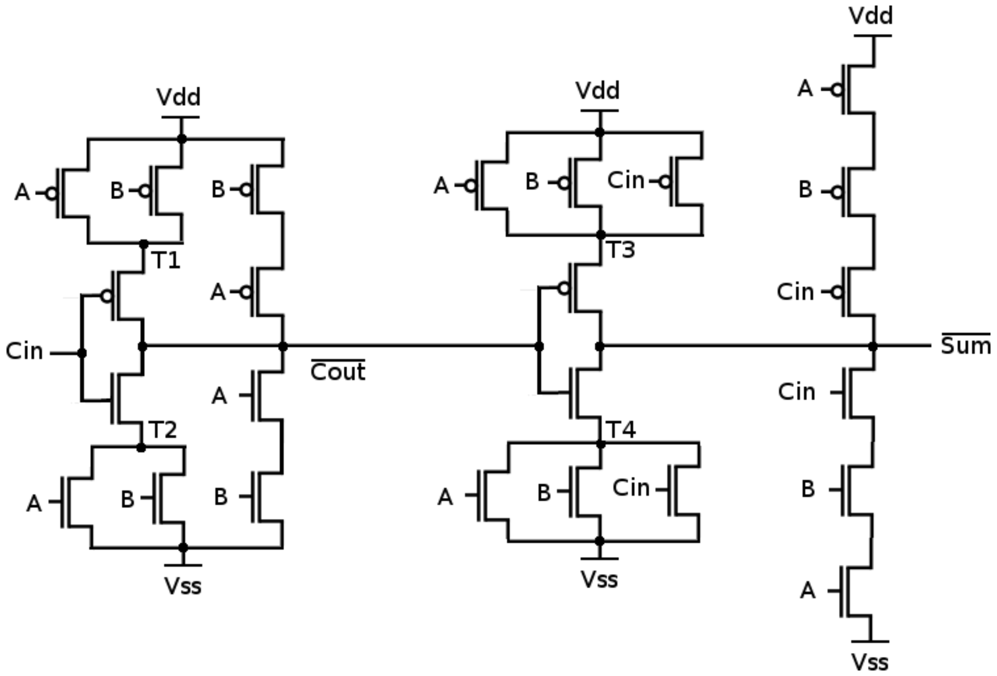

FIGURE 5.1: Mirror Adder showing the Reflected Logic across the X-Axis

Advantages of Mirror logic:

1. Carry Generation has no more than two transistors in series.
2. Easy to implement
3. Static, so low power consumption

Disadvantages however are;

1. Boolean expression of S and Co more difficult to identify in circuitry.
2. As with CMOS, connecting stages together means large capacitance due to having 2N inputs to drive.

The purpose of having Mirror Logic is to provide a performance and area comparison for the topology employed for the remaining design styles, in particular with CMOS since it shares the same advantages and disadvantages.

5.2 Schematic

With regards to the schematic, there is little to discuss since the captured schematic is equivalent to the arrangement in Figure 5.1, except for the addition of Inverters to give valid outputs. The only point to make is that due to the repeated nature of the Mirror Adder, it was only necessary to create a 1-bit slice, which was then iterated into a 4×4 array to give the full 16-bits.

5.3 Layout

5.3.1 Design

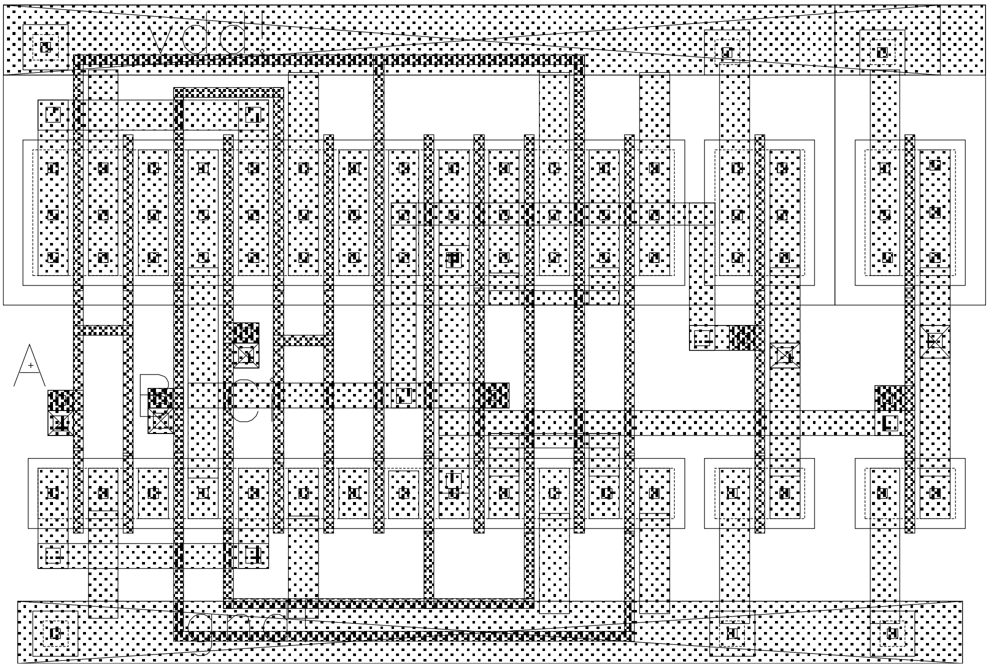

Figure 5.2: 1-Bit Mirror Adder

As can be seen, the layout for the 1-bit cell is fairly straight forward since all the functions are containted within the cell, with no interdependancies other than the carry chain.

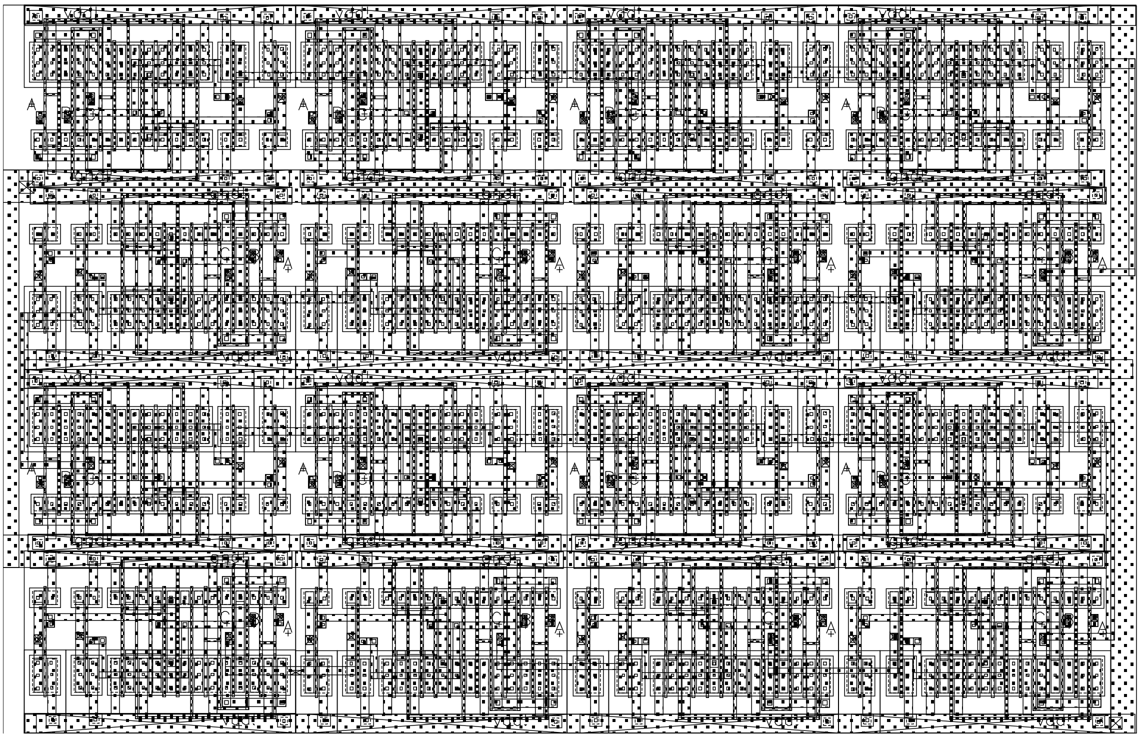

Figure 5.3: 16-Bit Mirror Adder

5.3.2 Simulation

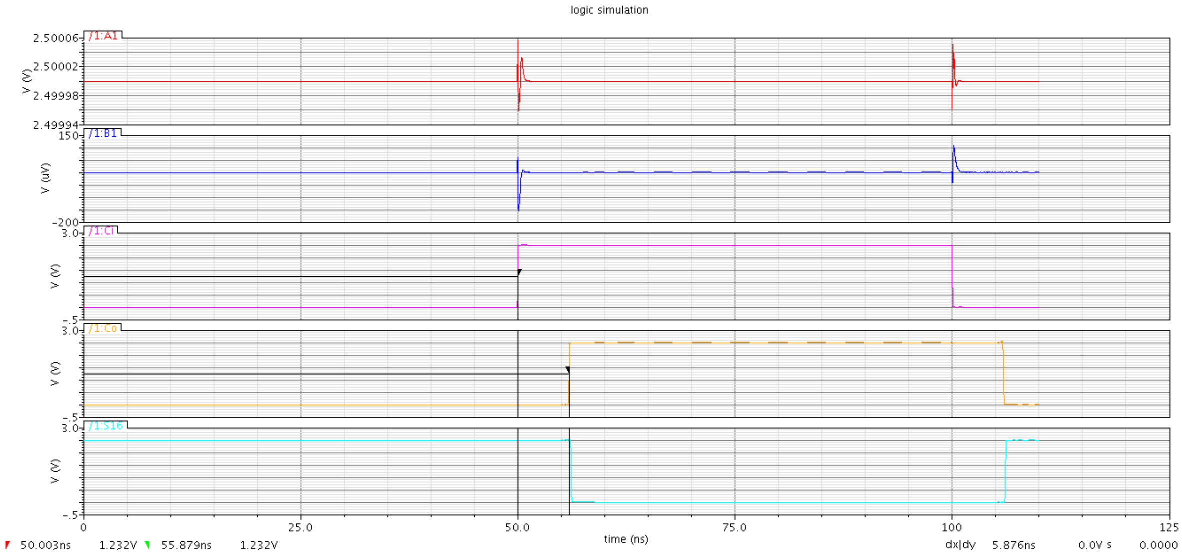

FIGURE 5.4: 16-Bit Mirror Adder Worst Case Simulation

From the simulation it can be seen that the carry delay is 5.86ns which is what is expected form this logic style owing to the large capacitances that each stage loads the previous stage with.

5.4 Results and Conclusion

Cout Delay (ns)	5.86
Length (um)	63.12
Width (um)	98.34
Area (m^2)	6.21*10^{-9}
NMOS Transistors	224
PMOS Transistors	224
Total Transistors	448

The results for this design is satisfactory in that, it shows the improvements the proposed topology in Figure 1.1 gives to the performance, with only a 50% increase in transistor count.

Chapter 6

Conclusion

	CMOS	Mirror	Dynamic	Ratioed
Cout Delay (ns)	2.21	5.8	1.16	1.38
Area (m2)	$1.61*10^{-8}$	$6.21*10^{-9}$	$1.60*10^{-8}$	$8.45*10^{-9}$
Power	1.89mW	N/A	4.68mW	57mW

Unfortunately the Mirror designed developed a fault within the tool that meant simulations wouldnt run on it again and so the power could not be extracted. However, we can see that as with any design, there is always a tradeoff. Dynamic and Ratioed exhibited similar performance, but the difference in Power was tenfold due to the static power consumption of Ratioed. With CMS showing with this topology the best balance between power and performance. What is interesting is the difference between the delays in CMOS and Mirror being more than double whilst the area is less than double. This is due to the hybrid between CLA and Ripple Carry adding a smaller area to the overall Ripple Carry design. It was calculated that the difference between CMOS and Mirror with regards to transistor count was only 50

Overall the design was a success and with a little more time could be tweaked to reduce the power consumption further.