5LIH0: Digital Integrated Circuit Design Project Report

Siddhant Rajashekar Honnalli (1328492), Stijn Kamp (0938450)

s.r.honnalli@student.tue.nl, s.l.kamp@student.tue.nl

I. Introduction

A full adder is the adder which adds 3 inputs and produces 2 outputs. The first 2 inputs are A and B and the third input is C-in (carry in). The output sum is designated as S and carry-out as C-out. To add N-bit numbers, a parallel adder is required that uses N number of full adder (FA). In this logic, the carry out of each FA is carry in of the next most significant FA. However, there is delay due to propagation of carry from initial stage (LSB) to final stage (MSB) and the delay increases if the number of parallel adders increases. To minimize the delay, a carry look ahead adder (CLA) is used. This adder works by creating 2 signals, i.e. carry propagator (P) and carry generator (G). The carry propagator is propagated to the next levels whereas carry generator is used to generate output carry despite input carry.

Kogge-stone adder is an example of CLA and it is the fastest adder which focuses on the design time. It reduces the delay of passing the carry through the lookahead stages with a fanout of 2 at each stage. This is main motivation behind using the Kogge-stone adder compared to other adders because of its low fanout per stage and low delay. The downside of the Kogge-stone adder is the large amount of wiring compared to other 16-bit adders.

II. Problem Formulation

The objective of this project is to design a 16-bit Kogge-Stone adder as shown in Fig.1. Two 16-bit numbers (A[15:0]) and (B[15:0]) and 1 bit carry-in signal (CIN) are the inputs of the 16-bit adder. A 16-bit sum (S[15:0]) and 1-bit carry out signal (COUT) are the outputs. The technology used is 45nm CMOS process, supply voltage is 1V and the adder drives a 40 fF capacitor load. The main goal is to design the adder with minimum delay and minimum area. The minimum area is achieved by sizing the transistors to minimum as possible which reduces the area of each cell which in turn reduces the overall area of adder. The minimum delay is achieved by varying the transistor sizes and including buffers at required stages. Equation (1) helps give an estimate of the tree delay.

$$t_{tree} = t_{pg} + [\log_2 N] * t_{AO} + t_{xor} \qquad (1)$$

where,

t_{pg} – delay of the 1-bit propagate/generate gates,
t_{AO} – delay of the AND-OR gate in the gray cell
t_{xor} – delay of the final sum XOR.

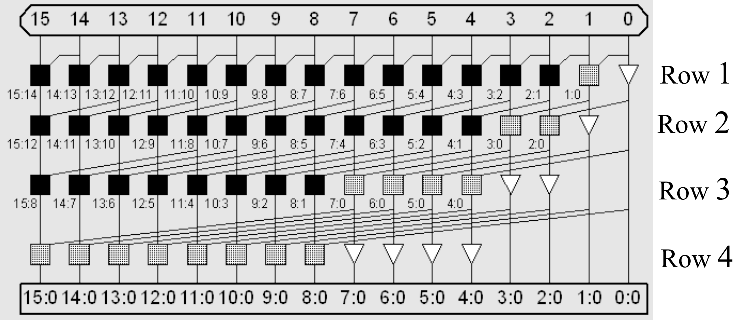

Fig. 1. The tree structure for a Kogge-stone adder

III. CMOS Circuit Design

For the CMOS circuit design, the adder is divided into separate blocks and a systematic approach is taken to make sure the final physical layout will be without refactoring the complete design repeatedly. First for each block in the tree structure a schematic was drawn, when each block was finished the blocks were put together and the connections between the blocks were defined. At last this schematic was translated to a physical layout. Each step in the process is further explained.

First the separate schematics for the different blocks, the same convention was used of a standard CMOS design, the focus was on making a logically correct schematic, using the least number of transistors.

To generate the P and G signals, the ¬P signal was generated using the equation $\neg P = \neg(A \cdot B)$. Next, an extra inverter was used to make P, and ¬P was used also generate to ¬G, using the following equation $\neg G = \neg(\neg P \cdot (A + B))$. Then another inverter was used to generate G. 7 NMOS and 7 PMOS were needed to generate P and G, as shown in Fig. 2. For the gray cell a combination is made between an AND and OR gate, as shown in Fig. 3. Since, the aim was to minimize the number of transistors, a truth table has been made to check whether the implementation is correct. Then the design is generated by first designing the CMOS-side and the make the PMOS-side. Because we are using the CMOS side, $G_{i:j}$ is calculated using $\neg G_{i:j} = \neg (G_{i:k} + P_{i:k} \cdot G_{k-1:j})$, and using then this output is inverted to go back to $G_{i:j}$. A total of 4 NMOS and 4 PMOS transistors, see Fig. 4. This approach has also been used for the black cell, which is using the following equations: $\neg G_{i:j} = \neg (G_{i:k} + P_{i:k} \cdot G_{k-1:j})$ and $\neg P_{i:j} = \neg (P_{i:k} \cdot P_{k-1:j})$, then 2 inverters are used to get the correct output, see Fig. 5 for the result. At last S is calculated using the XOR, which is a simple basic XOR with 6 NMOS and 6 PMOS, see Fig 6. Regarding the sizing of the transistors we have kept the equivalent of using a minimum sized inverter, W_{NMOS} = 90nm and W_{PMOS} = 180nm. So, when we have to NMOS transistors in series they would be both 180nm. To still deal with capacitance load we have increased the size of the final buffer, where the W_{NMOS} = 360nm and W_{PMOS} = 720nm, so 4 times larger than the minimum sized NMOS/PMOS. Furthermore, the length of all transistors has been set to 50nm. For each schematic a DRC check has been done before continuing to the next step.

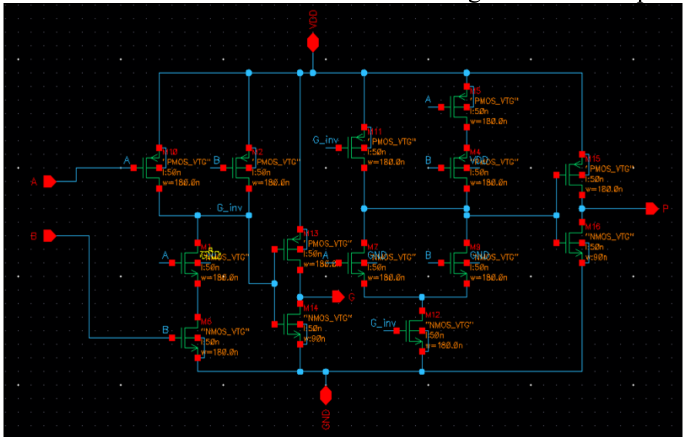

Fig. 2. The schematic of the PG generation cell

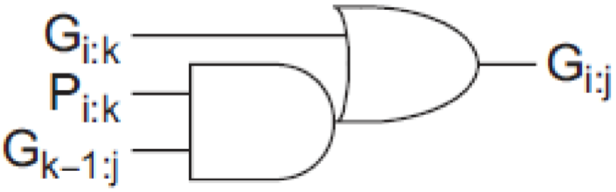

Fig. 3. Overview of the gray cell

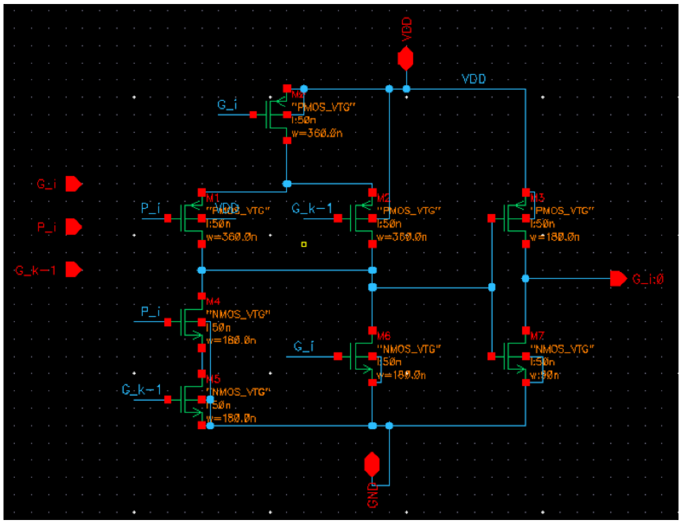

Fig. 4. Schematic of the gray cell

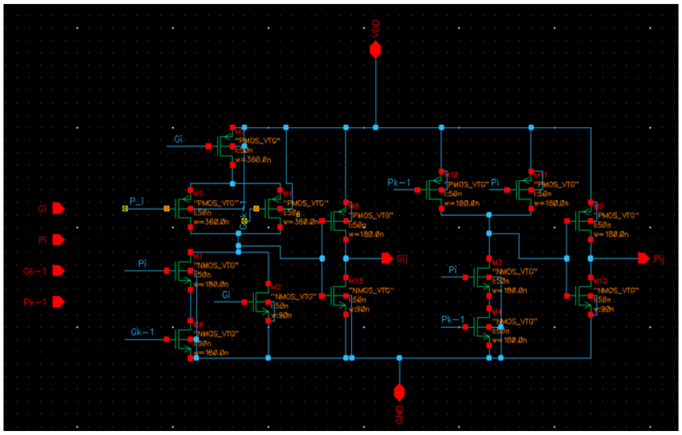

Fig. 5. Schematic of the black cell

When all schematics of the different cells were correct and working, the total schematic has been made using the given tree-structure. Here the same orientation is maintained as the tree-structure in order to keep a clear overview. Furthermore, it is made sure that the outputs of the symbols of the blocks were in such a way that the least amount of overlapping wires have been realized. The final schematic is shown in Fig. 6. Also, for the final schematic a DRC check has been done. When the final schematic did not have any errors, a test file with toggling inputs A and B for with fixed period was used to test the correctness of schematic. Next to the correctness, the delay of the total circuit was evaluated and the differences between the rise and fall times of the different bits and how it did influence the delay. In the next section the results of the delay will be discussed.

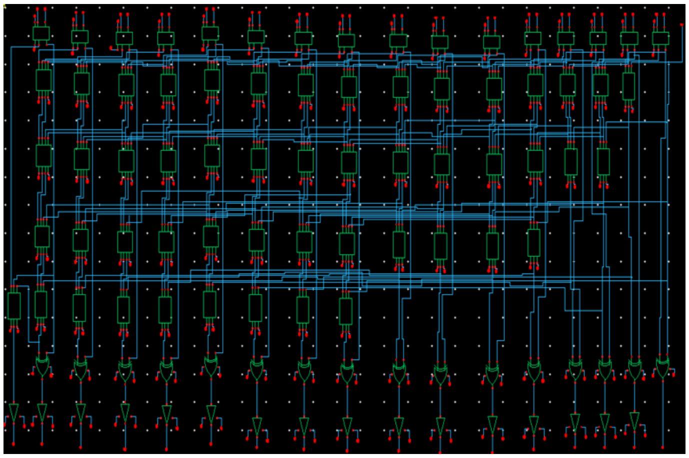

Fig. 6. Final schematic of the Kogge-Stone Adder

After the thorough analyzation of the correctness of the schematic, the start on the layout has been done. First a standard convention of height and width of the cells have been determined in order to make the final layout more structured and easier to implement. Each cell will have the same height, namely 1.52 µm. Next to that the width of the cells are also the same, so the cells can be placed next to each other. Next, the amount of contacts has been maximized to decrease the resistance as much as possible. Furthermore, there has been effort in reducing the metal wire lengths as much as possible and to avoid the metal 2 layer as much as possible. When the block was finished a DRC and LVS had been done to check the correctness of the layout, for each cell separately. One of the layouts is shown in Fig. 7, which is representing the black cell. Before working on the final layout, wiring plan of the total layout was decided. Since it is known that the Kogge-Stone Adder has a lot of wires and the wiring of a layout can be complex because of the strict design rules that are given, a decision was taken to export the total layout placement without any inter-wiring, and first draw the wires in Adobe Illustrator in order to play around with the placing of the wires, and to be able to adapt the wire placement were needed. Part of this approach result will be shown in Fig. 8, where the black wires illustrate the metal layer 3, and the blue lines illustrate the layer 4 metals. When it was figured out what was the most efficient looking alignment of the wires and how much space is needed, wiring was done on the final layout. Initially an assumption was made to place all the cells next to each other, but after using the Illustrator approach it was identified that the wires would not fit in the layout, and thereby move the cells further apart from each other. One other trick that was applied is to flip the horizontal uneven rows so we could overlap the GND of both cells and the VDD on the other side. Since the same convention of the contacts was followed on the GND and VDD side they were perfectly aligned and therefore would not give any DRC errors about overlapping wells and contacts.

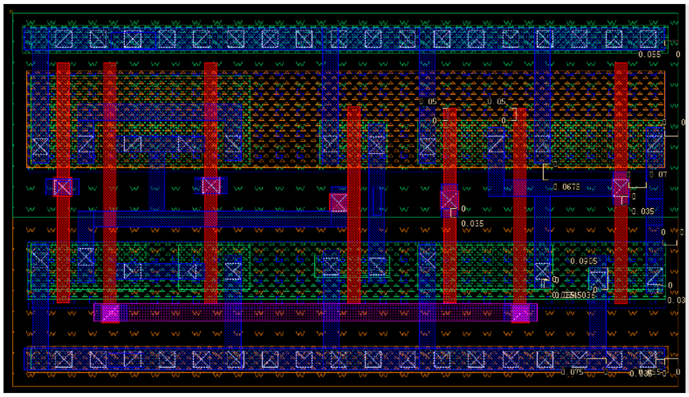

Fig. 7. Final layout of the black cell

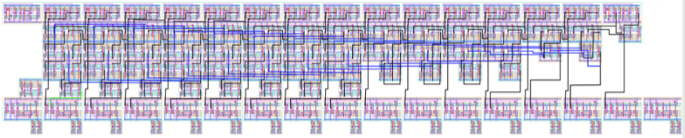

Fig. 8. Illustrator sketch for wiring of the final layout.

It is known that the wiring of the Kogge-Stone adder is challenging, especially between row 1, 2, 3 and 4 of Fig. 1, which is row 3, 4, 5 and 6 in the final layout including the PG-generator and the XOR gate. After all wires had been drawn inside the illustrator file, the outcomes combined with the design rules given, the spacing between row 3,4,5 and 6 needed to increase in order to fit in the wiring that was drawn inside the illustrator file. The width of the final layout was not a bottle neck for the wiring, so minimum width with black cells next to each other was applied. The result is shown in Fig. 9.

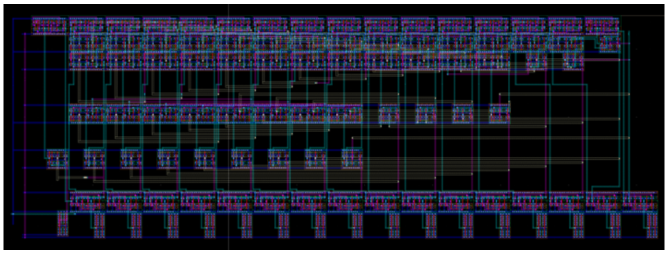

Fig. 9. Final layout of the 16-bit Kogge-Stone adder.

IV. Results and Analysis

To analyze the results of the schematic and layout, different simulations have been done,where the outputs of S and C_{out} are compared to the inputs of A, B and C_{in} to check the delay for the schematic, and also to simulate the energy consumption of the layout. To analyze the delays, a test file has been used for both the schematic and layout, so also the two could be compared. The bottleneck case for the delay, where the delay is expected to be the highest, is when the output is dependent on the C_{out}, in that case all outputs should wait for the C_{in} of that column. One of the cases where this is happening, is when all A's are going from zero to one, B's stays 0, and C_{in} goes from 0 to 1. Then first the outputs will go to 1, but after C_{in} has arrived, the output will go back to 0 except for C_{out}, which will go to 1. For the delay of the design the midpoint (at 500mV) of the last output that is flipping is picked. Furthermore, the simulation duration is set to 5ns, and the 1 input fields (A0), and all output fields are being plotted. The first input field to see when the A and Cin are going to 1, and all the output fields, because it is not certain that S15 will be the slowest of all outputs. When looking at the results of the schematics, C_{out} is the slowest, and has a delay of 385ps, see Tab. 1. Next, also the layout has been analyzed to see the differences and check what the delay would be, for this specific layout design. When looking at the layout, the overall delay was caused by S15, which was 762ps, from the moment A and C_{in} rises, see Tab. 1. At last also the energy and area of the layout has been analyzed. The area of the total layout has been calculated by using the ruler tool to get the width and height of the layout, which was in total 46*16.4 = 754µm^2, and the energy the discussed transition is 965fJ, as shown in Tab. 1.

	Schematic	Layout
Delay of S15	382ps	762ps
Delay of C_{out}	385ps	715ps
Overall delay	385ps	762ps
Energy		965fJ
Area		754 µm^2

Table 1. Analysis results of schematic and layout

V. Conclusions

In this project a fully functional 16-bit Kogge-stone adder layout and schematic was achieved with the above-mentioned delay and area. One of the major difficulties involved the wiring, as the adder demands lot of wiring. During the design of schematic, the approach for design keeping in mind the wires was not considered. Also, the floor plan of schematic and layout was the same which led increase in area to accommodate the wiring. Since we were not experienced enough to work with the tool/software, debugging the errors was difficult initially.

Our work could be improved by redesigning the individual cells such that they occupy minimum area as possible and the complete adder layout with minimum area. In the current work, a lot trade-off was made with area to accommodate wiring. Also, resizing the transistors in a custom way for each cell would improve the delay of carry network.

Declaration

Task	**Person Responsible**
Schematic floorplan	Siddhant and Stijn
Transistor floorplan for individual cells	Stijn
Layout design of cells	Siddhant
Layout floorplan	Stijn
Design of wiring for layout	Siddhant and Stijn
Placement of wires on layout	Siddhant
Debugging of layout	Siddhant and Stijn