8 BIT KOGGE STONE ADDER

EE 619 COURSE PROJECT

PROJECT REPORT

Submitted By:

Anurag Sindhu (Y5106)

Ashish Bhatia (Y5827121)

1. Abstract

KSA is a parallel prefix form carry look ahead adder. It generates carry in O (logn) time and is widely considered as the fastest adder and is widely used in the industry for high performance arithmetic circuits. In KSA, carries are computed fast by computing them in parallel at the cost of increased area.

2. Theory

The complete functioning of KSA can be easily comprehended by analyzing it in terms of three distinct parts :

1. Pre processing

   This step involves computation of generate and propagate signals corresponding too each pair of bits in A and B. These signals are given by the logic equations below:

$$p_i = A_i \; xor \; B_i$$

$$g_i = A_i \; and \; B_i$$

2. Carry look ahead network

   This block differentiates KSA from other adders and is the main force behind its high performance. This step involves computation of carries corresponding to each bit. It uses group propagate and generate as intermediate signals which are given by the logic equations below:

$$P_{i:j} = P_{i:k+1} \; and \; P_{k:j}$$

$$G_{i:j} = G_{i:k+1} \; or \; (P_{i:k+1} \; and \; G_{k:j})$$

3. Post processing

   This is the final step and is common to all adders of this family (carry look ahead). It involves computation of sum bits. Sum bits are computed by the logic given below:

$$S_i = p_i \text{ xor } C_{i-1}$$

3. Illustration

The working of KSA can be understood by the following Fig. 1 which corresponds to 4-bit KSA. 4-bit KSA is shown for simplicity.

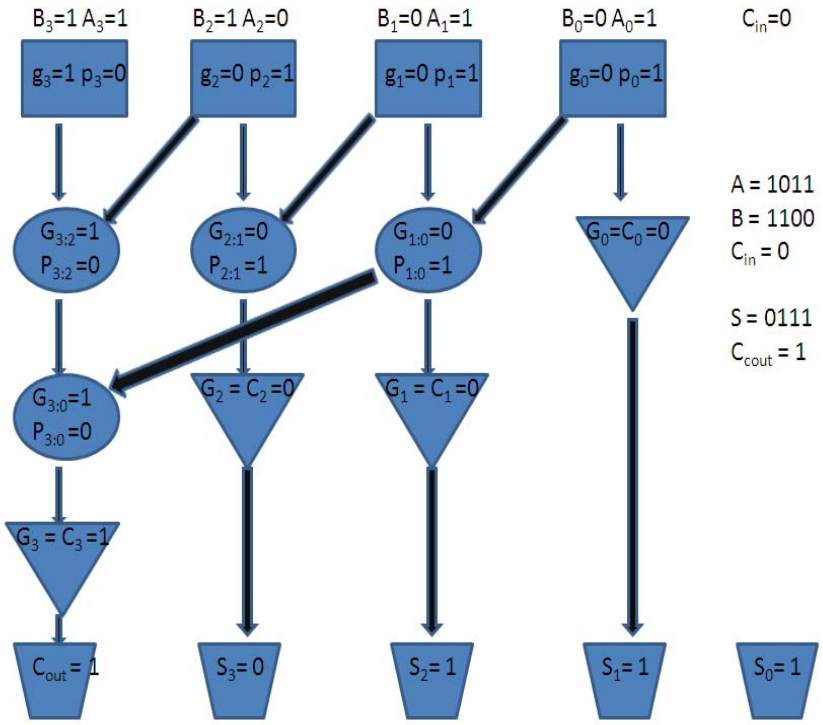

Fig. 1 Illustration of 4 bit KSA

4. Implementation

The schematic of KSA is implemented by using following building blocks :

1. Bit propagate and generate

   This block implements the following logic:

$$G_i = A_i \textbf{ AND } B_i$$
$$P_i = A_i \textbf{ XOR } B_i$$

   Schematic for this block is shown in Fig. 2

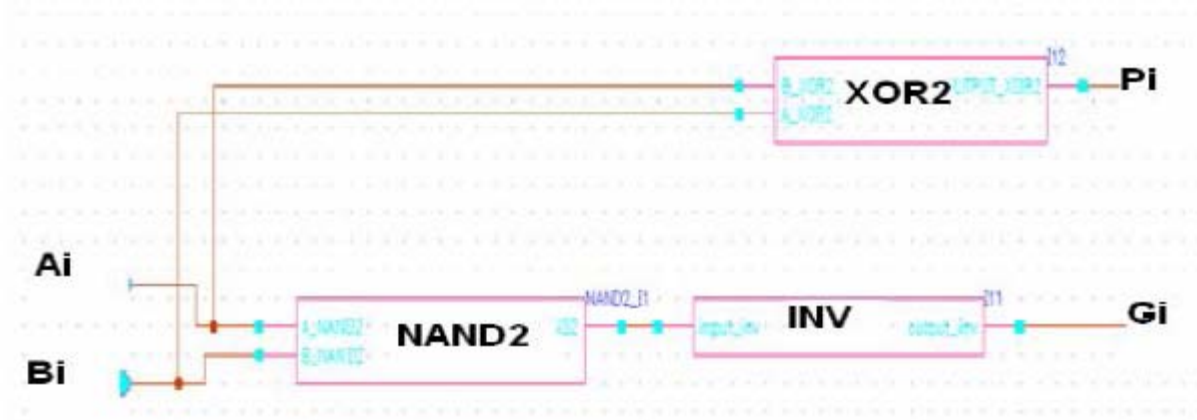

Fig 2. Schematic of Bit Propagate and Generate Block

2. Group propagate and generate

   This block implements the following logic:

$$G_2 = G_1 \textbf{ OR } (G_0 \textbf{ AND } P_1)$$
$$P_2 = P_1 \textbf{ AND } P_0$$

   Schematic for this block is shown in Fig. 3

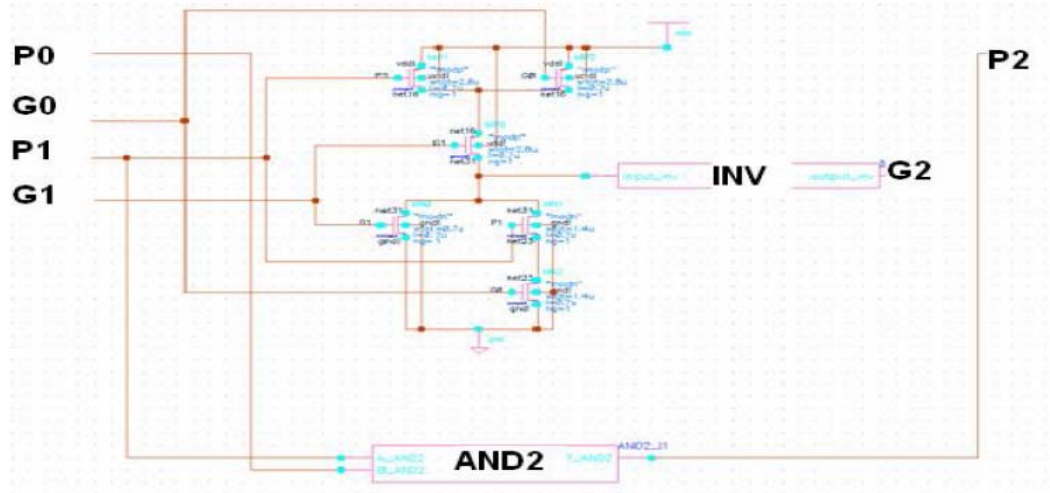

Fig 3. Schematic of Group Propagate and Generate Block

3. **Group Generate**

   This block implements the following logic:

$$G_2 = G_1 \textbf{ OR } (G_0 \textbf{ AND } P_1)$$

Schematic for this block is shown in Fig. 4

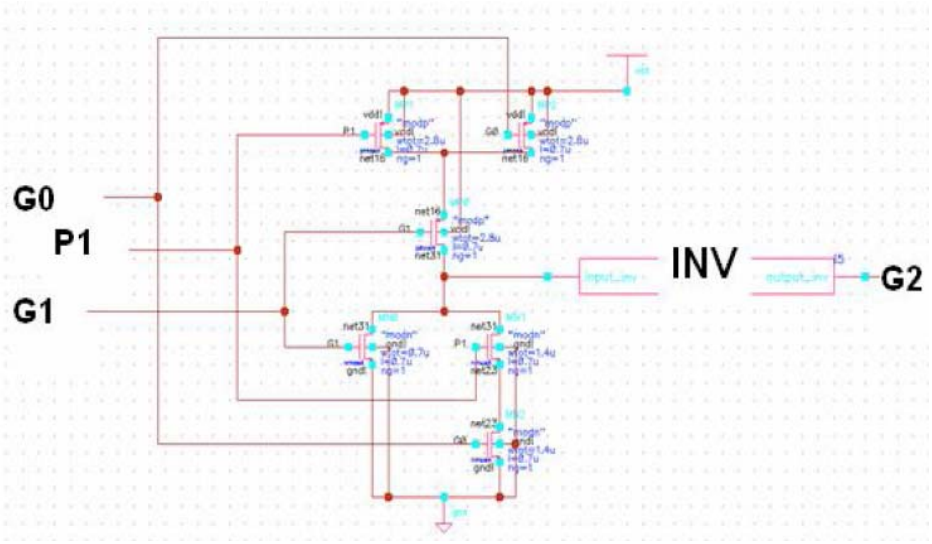

Fig 4. Schematic of Group Generate Block

Fig. 5 shows the complete schematic

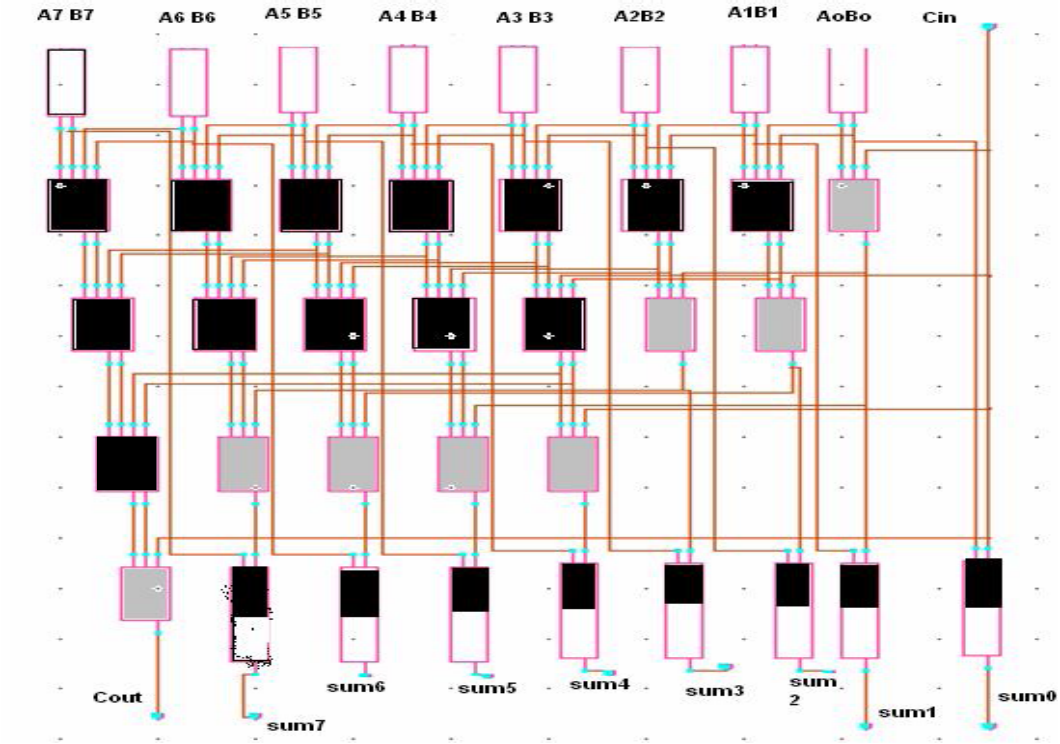

Fig 5. Complete schematic of 8-bit KSA

Following is the color coding for the blocks:
White: Bit propagates and generate
Black: Group propagate and generate
Grey: Group generate
Half White Half Black: XOR

5. Layout

The complete layout is shown in Fig. 6. Technology used was AMS 0.35um c35b4. Layout was done using Cadence Virtuoso Layout Editor. The DRC and LVS runs were successfully done.

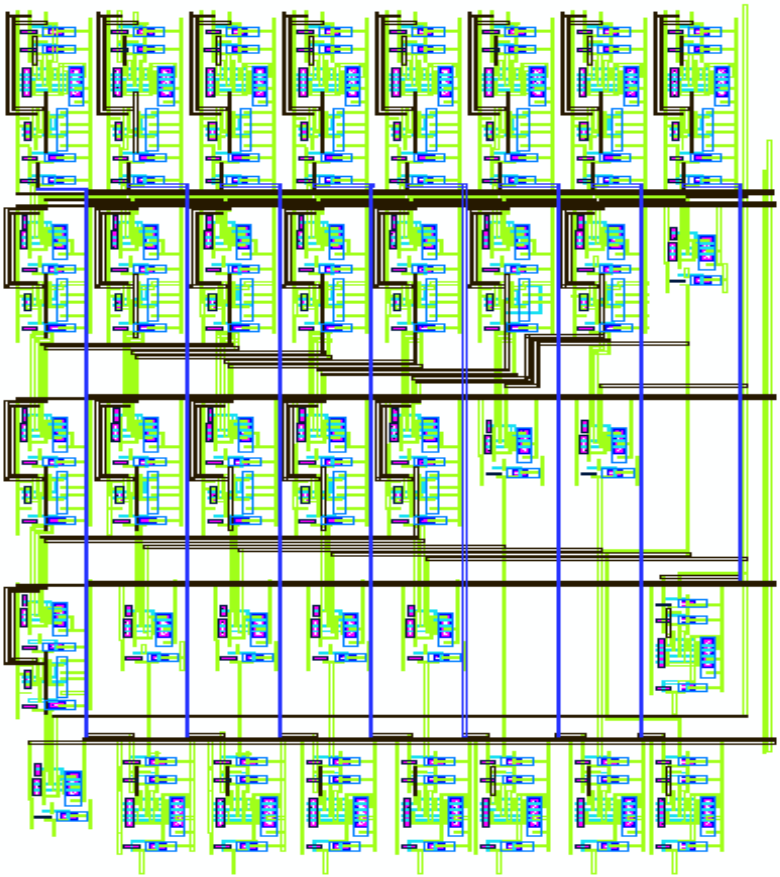

Fig 6. Complete Layout of 8-bit KSA

6. Simulation Results

Following specifications were achieved:

- Max frequency = 374.94 MHz
- Area = 440 µm X 300 µm = 0.132 mm^2
- Power = 460 uW

Worst case delay is shown in Fig. 7

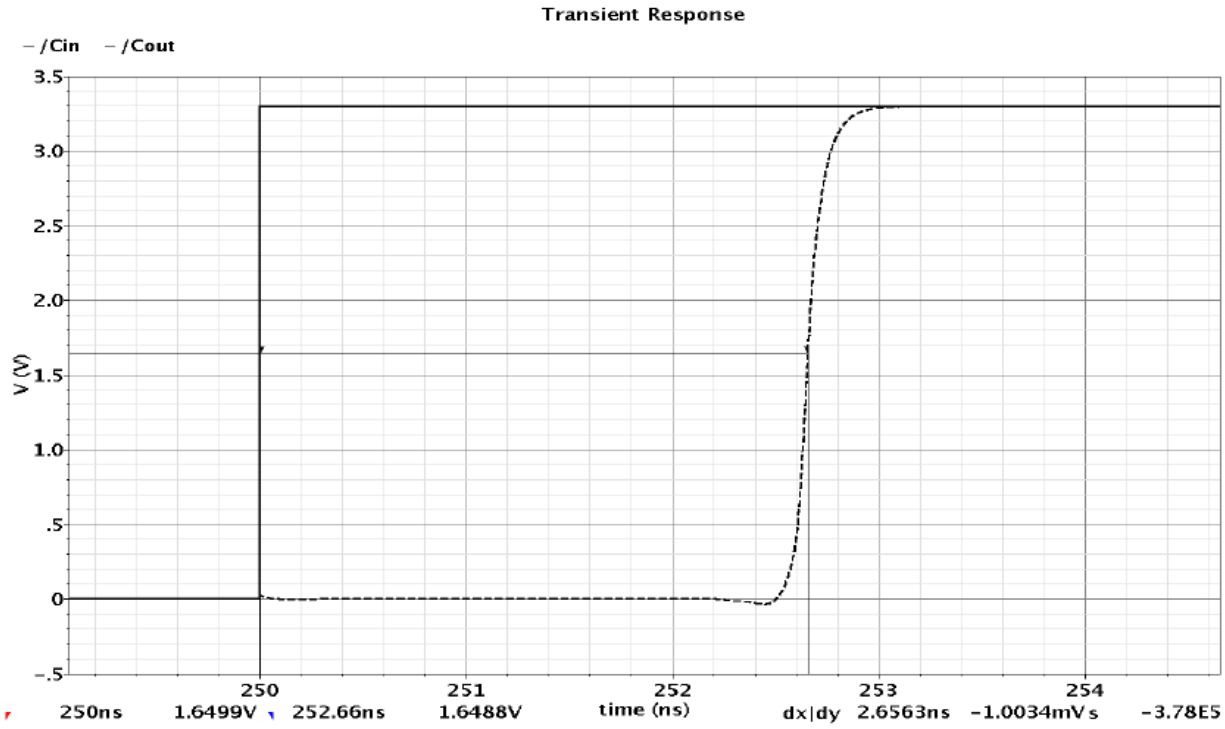

Fig 7. Worst Case Delay of 8-bit KSA

Power consumption is shown in Fig. 8

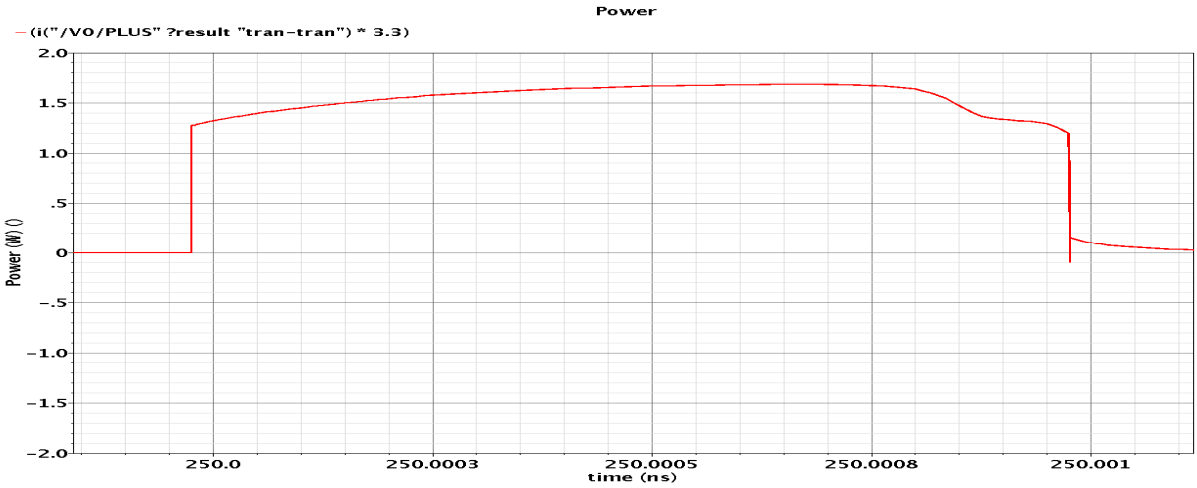

Fig 8. Power Consumption of 8-bit KSA

7. Conclusion

We designed and implemented 8 bit Kogge-Stone Tree Adder that operates at 375 MHz(f_{max}) and complete layout takes an area of 440 X 300 um^2.

References

- Swaroop Ghosh, Patrick Ndai, Kaushik Roy. "A Novel Low Overhead Fault Tolerant Kogge-Stone Adder Using Adaptive Clocking". DATE 2008.
- J. Rabaey, "Digital Integrated Circuits: A Design Perspective", Prentice Hall, 1996.