



## **PERFORMANCE AND ANALYSIS OF LOW POWER, AREA EFFICIENT AND HIGH SPEED CARRYFAST ADDER**

<sup>1\*</sup>.M.AntoBennet,<sup>2</sup>S.Sankaranarayanan<sup>3</sup>V.BanuPriya<sup>4</sup>PJayaPretheena<sup>5</sup>S.Yamini<sup>6</sup>S.Supriya

<sup>1</sup> Faculty of Electronics and Communication Department, vel tech, Chennai, India.

<sup>2</sup> UG Students of Electronics and Communication Department, vel tech , Chennai, India.

Email: [bennetmab@gmail.com](mailto:bennetmab@gmail.com)

---

**Submitted: May 27, 2017**

**Accepted: June 15, 2017**

**Published: Sep 1, 2017**

---

*Abstract- Carry Select Adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. From the structure of the CSLA, it is clear that there is scope for reducing the area and power consumption in the CSLA. This work uses a simple and efficient gate-level modification to significantly reduce the area and power of the CSLA. Based on this modification 8, 16,32,and 64-bit square-root CSLA (SQRT CSLA) architecture have been developed and compared with the regular SQRT CSLA architecture. The proposed design has reduced area and power as compared with the regular SQRT CSLA with only a slight increase in the delay. This work evaluates the performance of the proposed designs in terms of delay, area, power. Binary to Excess-1 Converter (BEC) instead of RCA with the regular CSLA to achieve lower area and power consumption. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder structure. The delay and area evaluation methodology of the basic adder blocks. The SQRT CSLA has been chosen for comparison with the proposed design as it has a more balanced delay, and requires lower power and area. Reducing the area and power consumption in the CSLA. Efficient gate-level modification to significantly reduce the area and power of the CSLA.*

**Index terms:** Carry Select Adder (CSLA), Binary to Excess-1 Converter (BEC), Carry-propagation adder (CPA)