**HIGH-SPEED AND ENERGY-EFﬁCIENT CARRY SKIP ADDER OPERATING UNDER A WIDE RANGE OF SUPPLY VOLTAGE LEVELS**

**Abstract:**

In this paper, we present a carry skip adder (CSKA) structure that has a higher speed yet lower energy consumption compared with the conventional one. The speed enhancement is achieved by applying concatenation and incrementation schemes to improve the efﬁciency of the conventional CSKA (Conv-CSKA) structure. In addition, instead of utilizing multiplexer logic, the proposed structure makes use of AND-OR-Invert (AOI) and OR-AND-Invert (OAI) compound gates for the skip logic. The structure may be realized with both ﬁxed stage size and variable stage size styles, wherein the latter further improves the speed and energy parameters of the adder. Finally, a hybrid variable latency extension of the proposed structure, which lowers the power consumption without considerably impacting the speed, is presented. This extension utilizes a modiﬁed parallel structure for increasing the slack time, and hence, enabling further voltage reduction. The proposed structures are assessed by comparing their speed, power, and energy parameters with those of other adders using a 45-nm static CMOS technology for a wide range of supply voltages. The results that are obtained using HSPICE simulations reveal, on average, 44% and 38% improvements in the delay and energy, respectively, compared with those of the Conv-CSKA. In addition, the power–delay product was the lowest among the structures considered in this paper, while its energy–delay product was almost the same as that of the Kogge–Stone parallel preﬁx adder with considerably smaller area and power consumption. Simulations on the proposed hybrid variable latency CSKA reveal reduction in the power consumption compared with the latest works in this ﬁeld while having a reasonably high speed.

**Index Terms—**Carry skip adder (CSKA), energy efﬁcient, high performance, hybrid variable latency adders, voltage scaling

**TOOLS:**

1. **XilinxISE 14.7**

**LANGUAGE:**

1. **VerilogHDL**