JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DESIGN AND ANALYSIS OF BASIC ADDER WITH 8-BIT PARALLEL PREFIX ADDER

Jyotsna Rani Nelli M.Tech Student
Department of Electronics and Communication Engineering
International School of Technology & Sciences
(Approved by AICTE & Affiliated to JNTUK Kakinada)
Rajamahendravaram - 533294

Dr R Prasad Associate Professor (Guide)
Department of Electronics and Communication Engineering
International School of Technology & Sciences
(Approved by AICTE & Affiliated to JNTUK Kakinada)
Rajamahendravaram - 533294

Abstract

Adders are the important and essential block of the digital circuit. It performs the addition in a digital circuit and so the overall performance of the digital circuit depends on the selection of a suitable adder. The proper selection of adder is needed to reduce the extra laborious work and complexity of a digital system. In this paper, we are comparing the different adders that have been synthesized using the Xilinx synthesis tool and simulated using the Verilog hardware description language and Xilinx simulation tool. The output of the simulation data helps in observing the different properties of adder. These properties get on the difference in operation and execution of the adders. Prefix Adder (PA), Kogge Stone Adder (KSA), Carry Select Adder (CSLA), Carry Skip Adder (CSKA), Carry Look Ahead Adder (CLA) and Ripple Carry Adder (RCA) have been compared with help of three basic aspects namely, device utilization i.e., number of slices, delay and power.

Keywords*— *HCA, KSA, CSLA, CSKA, CLA, RCA.

I. Introduction

Adder circuit is an important block of the digital system. In digital circuits, Adder plays a key function in any arithmetic operation of a computer. It is not used only in computers, but also applies in processors for different functions where increment of program counter is one of the examples. Adder circuit is also employed in digital signal processing (DSP) application, wireless communication, ALU, design of multiplier, DFT, IIR and DCT filters in VLSI system [1]. It has nearly become a necessity of the cognitive programs and also evaluated as pliant. The major aspect of ALU is the binary adder. A competent adder is one of those which enhances the presentation of the adder circuit and due to this number of transistors are reduced. However, adders play a critical role, based on device utilization, speed and Power, the selection of adder adjusts from one program to another program [2]. The important specifications of any digital circuits are:

- Less delay [3]
- Less transistors count i.e., low area
- reducing power dissipation
- reducing power consumption

Since all of the above referred properties cannot be demonstrated by a single adder, several adders take over from others based on the user's specification. In order to make the selection of adders suitable and now not an onerous one, the evaluation amongst different 4-bit adders has been completed extensively. The work in [6] looks similar to our proposed work, but in [6], the authors have made the comparison of different adders using VHDL, while not including an important adder.

In this paper, I have included the comparison of HCA with different adders by using another language, Verilog HDL. The evaluation amongst various adders has been completed notably.

1. RCA
2. CLA
3. CSKA
4. CSLA
5. KSA

The rest of the paper is organized as follows: The adders are explained in Sec. II. Sec. III provides the simulation results and conclusion is given in Sec. IV.

II. ADDER

A. Ripple Carry Adder

RCA is a series combination of full adder and combinational logic circuit. It is basically used for two n bit binary addition and also known as n bit parallel adder. It consists of a full adder for n bit addition. fig.1 shows the circuit block diagram. of RCA, that is showing the 4 bit RCA, in which we are using operands A and B that produce the result sum (S) and carry (C). First of all we give the input A1A2A3A4, B1B2B3B4 and initial carry Cin. Where the first full adder adds A1,B1 and initial carry Cin then it generates the output sum (S1) and carryout (Cout) of first full adder. The carryout (Cout) of first full adder becomes the input carry of second full adder and generates sum and carry and similarly this process is going on. Because of this it takes more time for addition. RCA has less device utilization and more delay.In this paper we analyze 8-bit ripple carry adder area, delay and power.

$$SUM = A\ xor\ B\ xor\ C\text{in} \quad (1)$$

$$Cout = AB + BC\text{in} + AC\text{in} \quad (2)$$

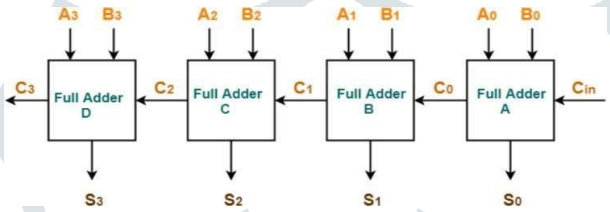

Fig. 1. Block diagram of RCA

B. Carry Look Ahead Adder

CLA is faster than RCA adder. It reduces the time delay and improves the speed but also increases the circuit complexity. This adder reduces the required time of carry bit in intermediate stage and also uses two concepts called generate carry and propagate carry. The term carry propagate will be used for the propagation of the next stage of adder and carry generation will be used for generation of carry regardless of input carry of the first stage.

Fig. 2. shows the circuit block diagram of CLA .In this, the generate and propagate operation are done by using the following equation. Here, A and B are considered as an operand term and that generate the sum and carry i.e., results.

CLA operation is done by using the following equation.

$$P\text{i} = A\text{i}\ xor\ B\text{i} \quad (3)$$

$$G\text{i} = A\text{i}\ and\ B\text{i} \quad (4)$$

$$S\text{i} = P\text{i}\ xor\ C\text{i} \quad (5)$$

$$C1 = G0\ or\ (P0\ and\ C0) \quad (6)$$

$$C2 = G1\ or\ (P1\ and\ G0)\ or\ (P1\ and\ P0\ and\ C0) \quad (7)$$

$$C3 = G2or(P2andG1)or(P2andP1andG0)$$
$$or\ (P2\ and\ P1\ and\ P0\ and\ C0) \quad (8)$$

$$C4 = G3or(P3andG2)or(P3andP2andG1)$$
$$or(P3andP2andP1andG0)$$
$$or\ (P3\ and\ P2\ and\ P1\ and\ C0) \quad (9)$$

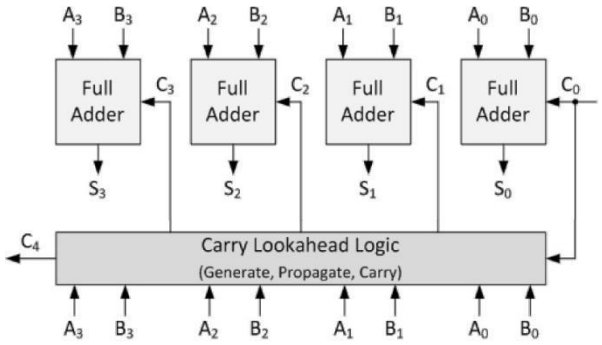

Fig. 2. Block diagram of CLA

C. Carry Skip Adder

CSKA is also called a carry bypass adder. To design of n bit, carry skip adder need n bit full adder, one n bit AND gate and one 2*1 multiplexer. This AND gate and Multiplexer are combinedly known as skip logic. We observe 3 case from the truth table:

- *A=B=0,*
- *A=B=1,*
- $A \neq B$

Fig. 3. shows the circuit block diagram of CSKA. In this figure, a single full adder has three inputs: A, B, Cin if two inputs are zero then Cout is always zero, if two inputs are one then Cout is always one, if two inputs are different then Cout is always equal to Cin. To avoid the delay, we are using the skip logic. In carry skip adder only one case (ab) that time only input carry propagates to output carry and remaining case Cout is fixed with zero or either one. In carry skip adder, when all the input of full adder is complemented to each other then the only output of full adder is one, then all output is applied to AND gate and it acts like a select line of multiplexer. It will select the initial carry or output carry then give the output carry (Cout). CSKA speed is high but device utilization is more comparatively other adder.

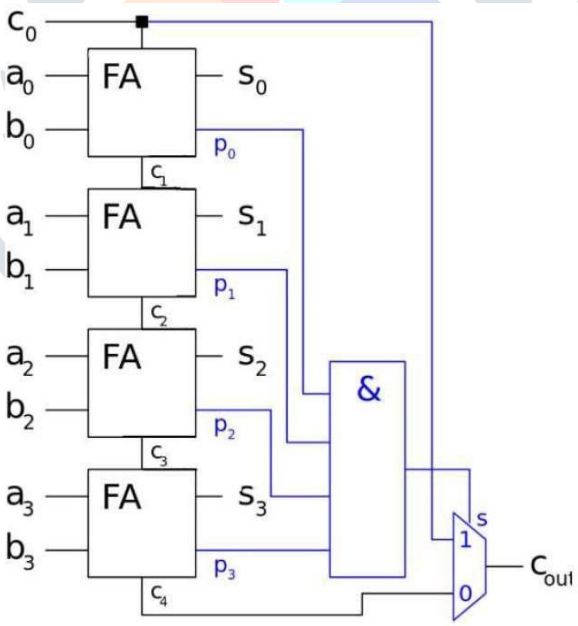

Fig. 3. Block diagram of CSKA

D. Carry Select Adder

CSLA is a fast adder and simple combinational circuit. It consists of two ripple carry adders and one multiplexer. CSA is similar to the RCA except that CSA uses single bit using two ripple carry adders in circuit design. In this adder, addition is similar to the RCA but carry is fixed with zero and one. Whatever, give the input as a carry according to this multiplexer select and give the sum and Cout. Fig.4 shows the circuit diagram of CSLA. In this figure, 4-bit addition carry select adder which gives the sum and Cout by using two ripple carry adder and multiplexer. Here the first batch of RCA gives input as a carry zero and another batch of the same

input of RCA gives input carry as a one. The addition of adders is allowed as inputs to a multiplexer series and the full adder carry finds the succeeding RCA that is to be considered and the similar process is applied for output carry. CSLA has high speed but device utilization is high.

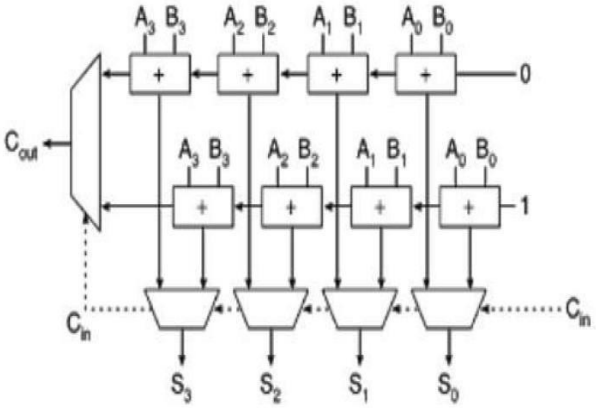

Fig. 4. Block diagram of CSLA

E. Kogge Stone Adder

KSA is the type of parallel prefix adder and fastest adder comparatively. It is based on the carry look ahead adder. KSA conducts operations with great speed and also it has low fans out of every stage. This adder totally decreases the propagation delay bit by bit in the circuit to produce the carry signal. Hence it is widely used by industries or circuits that demand great speed operations [4]. Fig. 5 shows the circuit diagram of KSA. In this, the circuit can be divided into three steps. In the first step, propagate (P) and generate(G) signals are generated using each input bit of the pair signal and this is known as the preprocessing stage . During the second step, carries are generated using each single bit separately and carried out parallelly, this is used for carry generation and carry propagation in intermediate stage logically and this is known as carry generation stage. Final steps are common for all adders, calculate the sum and carry and this is known as the post processing stage. P1 and G1 generate by the help of the below indicated equations to make the use of P and G. For the future estimate of the carries, the generate and propagate terms are obtained. The sum bits are produced depending on the outputs of the first stage and second stage. In this way we have to get results upto 8-bits of input data.

$$Pi = Ai\ xor\ Bi \quad (10)$$

$$Gi = Ai\ and\ Bi \quad (11)$$

$$CPi{:}j = Pi{:}k{+}1\ and\ Pk{:}j \quad (12)$$

$$CGi{:}j = Gi{:}k{+}1\ or\ Pi{:}k{+}1\ and\ Gk{:}j \quad (13)$$

$$Ci{-}1 = (Pi\ and\ Ci)\ or\ Gi \quad (14)$$

$$Si = Pi\ xor\ Ci{-}1 \quad (15)$$

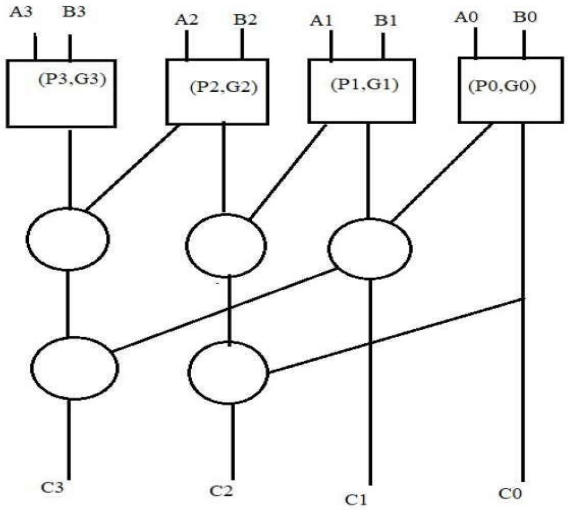

Fig. 5. Block diagram of KSA

F. Han Carlson adder

The Han Carlson adder is the networks family midway Brent-kung and KSA [5]. It can be viewed as KSA. This adder is similar to KSA in the sense that these executes propagate/generate operation on odd bits and carry-combine operation on even bits. As in last, producing the true carry bits odd bits recombine with even bits carry signals. It has five stages in which the central three stages are similar to KSA. It utilizes much less cells compared to KSA and is shorter. Thus, there is a depletion in complexity at the cost of extra stages for carry combine path. Speculative prefix processing stage of Han Carlson can be generated by deleting the last row of KSA. The HCA block diagram is shown fig.6. The HCA is calculated by using this equation which is shown below.

$$Pi = Ai\ xor\ Bi \quad (16)$$

$$Gi = Ai\ and\ Bi \quad (17)$$

$$CPi{:}j = Pi{:}k{+}1\ and\ Pk{:}j \quad (18)$$

$$CGi{:}j = Gi{:}k{+}1\ or\ Pi{:}k{+}1\ and\ Gk{:}j \quad (19)$$

$$Ci{-}1 = (Pi\ and\ Ci)\ or\ Gi \quad (20)$$

$$S_i = P_i\ xor\ C_{i-1} \quad (21)$$

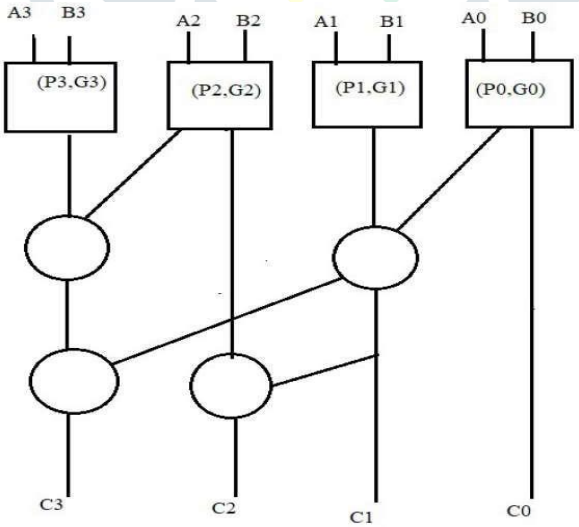

Fig. 6. Block diagram of HCA

III. SIMULATIONS AND RESULTS

Delay, device utilization and power has been calculated by using XILINX ISE 14.7 using Verilog.

A. 8-bit Ripple Carry Adder

1.Delay

```
Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 97 / 9
-------------------------------------------------------------------------
Delay:               14.846ns (Levels of Logic = 10)
  Source:            b<0> (PAD)
  Destination:       carry (PAD)

  Data Path: b<0> to carry
                                   Gate     Net
    Cell:in->out      fanout   Delay    Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2    1.218    0.622  b_0_IBUF (b_0_IBUF)
     LUT3:I0->O            2    0.704    0.526  c_1_or00001 (c<1>)
     LUT3:I1->O            2    0.704    0.526  c_2_or00001 (c<2>)
     LUT3:I1->O            2    0.704    0.526  c_3_or00001 (c<3>)
     LUT3:I1->O            2    0.704    0.526  c_4_or00001 (c<4>)
     LUT3:I1->O            2    0.704    0.526  c_5_or00001 (c<5>)
     LUT3:I1->O            2    0.704    0.526  c_6_or00001 (c<6>)
     LUT3:I1->O            2    0.704    0.526  c_7_or00001 (c<7>)
     LUT3:I1->O            1    0.704    0.420  c_8_or00001 (carry_OBUF)
     OBUF:I->O                  3.272           carry_OBUF (carry)
    ----------------------------------------
    Total                      14.846ns (10.122ns logic, 4.724ns route)
                                        (68.2% logic, 31.8% route)

=========================================================================
```

2. Device Utilization

rca8 Project Status			
Project File:	rca.xise	Parser Errors:	No Errors
Module Name:	rca8	Implementation State:	Synthesized
Target Device:	xc3s500e-4fg320	• Errors:	No Errors
Product Version:	ISE 14.7	• Warnings:	No Warnings
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	9	4656	0%
Number of 4 input LUTs	16	9312	0%
Number of bonded IOBs	26	232	11%

3.Power

Device	
Family	Spartan3e
Part	xc3s500e
Package	fg320
Temp Grade	Commercial
Process	Typical
Speed Grade	-4

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	16	9312	0
Signals	0.000	33	---	---
IOs	0.000	26	232	11
Leakage	0.081			
Total	0.081			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	26.1	82.9	27.1

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.026	0.000	0.026
Vccaux	2.500	0.018	0.000	0.018
Vcco25	2.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.081	0.000	0.081

B. Carry Look Ahead Adder

1. Delay

```
Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 22 / 1
-------------------------------------------------------------------------
Delay:               10.814ns (Levels of Logic = 7)
  Source:            i_add1<1> (PAD)
  Destination:       o_result<8> (PAD)

  Data Path: i_add1<1> to o_result<8>
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             1    1.218   0.595  i_add1_1_IBUF (i_add1_1_IBUF)
     LUT4:I0->O            1    0.704   0.455  w_C_8_or000029 (w_C_8_or000029)
     LUT3:I2->O            1    0.704   0.424  w_C_8_or000057_SW0 (N9)
     LUT4:I3->O            1    0.704   0.455  w_C_8_or000057 (w_C_8_or000057)
     LUT4:I2->O            1    0.704   0.420  w_C_8_or0000134_SW1_SW0 (N17)
     MUXF5:S->O            1    0.739   0.420  w_C_8_or0000134_f5 (o_result_8_OBUF)
     OBUF:I->O                  3.272          o_result_8_OBUF (o_result<8>)
    ----------------------------------------
    Total                      10.814ns (8.045ns logic, 2.769ns route)
                                        (74.4% logic, 25.6% route)
```

2. Device Utilization

carry_lookahead_adder_8_bit Project Status			
Project File:	cla.xise	Parser Errors:	No Errors
Module Name:	carry_lookahead_adder_8_bit	Implementation State:	Synthesized
Target Device:	xc3s500e-4fg320	• Errors:	No Errors
Product Version:	ISE 14.7	• Warnings:	2 Warnings (2 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	4	4656	0%
Number of 4 input LUTs	8	9312	0%
Number of bonded IOBs	25	232	10%

3. Power

Device	
Family	Spartan3e
Part	xc3s500e
Package	fg320
Temp Grade	Commercial
Process	Typical
Speed Grade	-4

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	16	9312	0
Signals	0.000	33	---	---
IOs	0.000	26	232	11
Leakage	0.081			
Total	0.081			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	26.1	82.9	27.1

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.026	0.000	0.026
Vccaux	2.500	0.018	0.000	0.018
Vcco25	2.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.081	0.000	0.081

C. Carry Select Adder

1. Delay

```
Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 104 / 9
-------------------------------------------------------------------------
Delay:               11.208ns (Levels of Logic = 7)
  Source:            A<1> (PAD)
  Destination:       C (PAD)

  Data Path: A<1> to C
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   1.218   0.622  A_1_IBUF (A_1_IBUF)
     LUT4:I0->O            2   0.704   0.482  rc_low_nibble_0/fa2/Mxor_S_xo<0>21 (N9)
     LUT3:I2->O            2   0.704   0.526  rc_low_nibble_0/fa2/Cout1 (rc_low_nibble_0/C2)
     LUT3:I1->O            4   0.704   0.666  rc_low_nibble_0/fa3/Cout1 (Clow)
     LUT4:I1->O            2   0.704   0.482  muxs/X<3>128 (muxs/X<3>128)
     LUT4:I2->O            1   0.704   0.420  muxc/X_0_mux00001 (C_OBUF)
     OBUF:I->O                 3.272          C_OBUF (C)
    ----------------------------------------
    Total                     11.208ns (8.010ns logic, 3.198ns route)
                                       (71.5% logic, 28.5% route)
```

2 Device Utilization

carry_select_adder Project Status (07/07/2022 - 08:59:41)			
Project File:	csla.xise	Parser Errors:	No Errors
Module Name:	carry_select_adder	Implementation State:	Synthesized
Target Device:	xc3s500e-4fg320	•Errors:	No Errors
Product Version:	ISE 14.7	•Warnings:	3 Warnings (3 new)
Design Goal:	Balanced	•Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	•Timing Constraints:	
Environment:	System Settings	•Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	10	4656	0%
Number of 4 input LUTs	18	9312	0%
Number of bonded IOBs	25	232	10%

3.Power

Device	
Family	Spartan3e
Part	xc3s500e
Package	fg320
Temp Grade	Commercial
Process	Typical
Speed Grade	-4

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	16	9312	0
Signals	0.000	33	--	--
IOs	0.000	26	232	11
Leakage	0.081			
Total	0.081			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	26.1	82.9	27.1

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.026	0.000	0.026
Vccaux	2.500	0.018	0.000	0.018
Vcco25	2.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.081	0.000	0.081

D. Kogge Stone Adder

1. Delay

```
Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 89 / 9
-------------------------------------------------------------------------
Delay:               8.815ns (Levels of Logic = 5)
  Source:            a<4> (PAD)
  Destination:       sum<7> (PAD)

  Data Path: a<4> to sum<7>
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             7   1.218   0.883  a_4_IBUF (a_4_IBUF)
     LUT4:I0->O            1   0.704   0.455  ccg_6_or000011 (N3)
     LUT3:I2->O            1   0.704   0.420  ccg_6_or0000_SW0 (N14)
     MUXF5:S->O            1   0.739   0.420  Mxor_sum<7>_Result1_f5 (sum_7_OBUF)
     OBUF:I->O                 3.272          sum_7_OBUF (sum<7>)
    ----------------------------------------
    Total                      8.815ns (6.637ns logic, 2.178ns route)
                                       (75.3% logic, 24.7% route)

=========================================================================
```

2. Device Utilization

ksa8 Project Status (07/07/2022 - 09:30:26)			
Project File:	ksadd.xise	Parser Errors:	No Errors
Module Name:	ksa8	Implementation State:	Synthesized
Target Device:	xc3s500e-4fg320	• Errors:	No Errors
Product Version:	ISE 14.7	• Warnings:	1 Warning (0 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	12	4656	0%
Number of 4 input LUTs	23	9312	0%
Number of bonded IOBs	26	232	11%

3.Power

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Device			On-Chip	Power (W)	Used	Available	Utilization (%)		Supply Summary		Total	Dynamic	Quiescent
Family	Spartan3e		Logic	0.000	16	9312	0		Source	Voltage	Current (A)	Current (A)	Current (A)
Part	xc3s500e		Signals	0.000	33	---	---		Vccint	1.200	0.026	0.000	0.026
Package	fg320		IOs	0.000	26	232	11		Vccaux	2.500	0.018	0.000	0.018
Temp Grade	Commercial		Leakage	0.081					Vcco25	2.500	0.002	0.000	0.002
Process	Typical		Total	0.081									
Speed Grade	-4										Total	Dynamic	Quiescent
									Supply Power (W)		0.081	0.000	0.081
Environment			Thermal Properties		Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)						
Ambient Temp (C)	25.0				26.1	82.9	27.1						
Use custom TJA?	No												
Custom TJA (C/W)	NA												
Airflow (LFM)	0												
Characterization													
PRODUCTION	v1.2,06-23-09												

E. Prefix Adder

1. Delay

```
================================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 100 / 8
--------------------------------------------------------------------------------
Delay:               12.294ns (Levels of Logic = 8)
  Source:            b<0> (PAD)
  Destination:       S<7> (PAD)

  Data Path: b<0> to S<7>
                                   Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   1.218   0.622  b_0_IBUF (b_0_IBUF)
     LUT3:I0->O            2   0.704   0.526  c1/o2/o1 (cg<1>)
     LUT3:I1->O            2   0.704   0.482  c2/o2/o1 (cg<2>)
     LUT3:I2->O            3   0.704   0.531  c3_1/o2/o1 (cg<3>)
     MUXF5:S->O            2   0.739   0.482  c5_1/o2/o_f5 (cg<5>)
     LUT3:I2->O            2   0.704   0.482  c6_1/o2/o1 (cg<6>)
     LUT3:I2->O            1   0.704   0.420  s6/x3/xor2_1/Mxor_o_Result1 (S_6_OBUF)
     OBUF:I->O                 3.272          S_6_OBUF (S<6>)
    ----------------------------------------
    Total                     12.294ns (8.749ns logic, 3.545ns route)
                                       (71.2% logic, 28.8% route)

================================================================================
```

2. Device Utilization

prefixAdd Project Status (07/07/2022 - 09:03:50)			
Project File:	prefix.xise	Parser Errors:	No Errors
Module Name:	prefixadd	Implementation State:	Synthesized
Target Device:	xc3s500e-4fg320	•Errors:	No Errors
Product Version:	ISE 14.7	•Warnings:	25 Warnings (25 new)
Design Goal:	Balanced	•Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	•Timing Constraints:	
Environment:	System Settings	•Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	9	4656	0%
Number of 4 input LUTs	16	9312	0%
Number of bonded IOBs	25	232	10%

3. Power

Device		On-Chip	Power (W)	Used	Available	Utilization (%)
Family	Spartan3e	Logic	0.000	16	9312	0
Part	xc3s500e	Signals	0.000	33	---	---
Package	fg320	IOs	0.000	26	232	11
Temp Grade	Commercial	Leakage	0.081			
Process	Typical	Total	0.081			
Speed Grade	-4					

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	26.1	82.9	27.1

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.026	0.000	0.026
Vccaux	2.500	0.018	0.000	0.018
Vcco25	2.500	0.002	0.000	0.002

		Total	Dynamic	Quiescent
Supply	Power (W)	0.081	0.000	0.081

Table. 1 shows the comparison of different adders with respect to delay, power and device utilization. The evaluation was made below the family of spartan 3e, device part xc3s100e, package deal of fg320 and speed grade -four. From the table, we can see that CLA has less device utilization i.e., number of slices and also more delay. KSA has occupied moderate device utilization i.e., number of slices and also less delay.

TABLE I. COMPARISON OF ADDER

Adder	Delay (ns)	Power (mw)	Device Utilization	
			Slices	LUT
RCA(8-bit)	14.846	81	9	16
CLA(8-bit)	10.814	81	4	8
CSLA(8-bit)	11.208	81	10	18
KSA(8-bit)	8.815	81	12	23
PA(8-bit)	12.294	81	9	16

Fig 7: Delay analysis report

CONCLUSION

In this paper, we compare all different types of 8-bit adder with power, device utilization and delay. In respect of delay, Kogge Stone Adder, which is a high-speed adder, gives high quality end consequence out of the adders chosen. KSA and PA work following the parallel prefix computation. In our work it was observed that all adders consume the same power. The area of all adders depends on the number of slices and RCA, CLA and PA have the least slice be counted. LUT's another manner has to have a significant range and right here in the work carried out CSKA and KSA has the highest LUT remember out of the six adders. Hence KSA is nice amongst all the adders proven above, because it requires a much moderate location like that of RCA and also requires less time than that of RCA. The effective response can be acquired using the KSA.

REFERENCES

[1] N. Varshney and G. Arya,"Design and Execution of Enhanced Carry Increment Adder using Han-Carlson and Kogge-Stone adder

Technique," IEEE. Conf. on Electronics Communication and Aerospace Technology, 2019.

[2] V.G. Oklobdzija, B.R. Zeydel and H.Q. Dao,"Comparison of high performance VLSI adders in the energy-delay space," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 13, no. 6, 2005.

[3] S. Pandu, A. Benerjee , B. Maji , and Dr. A.K. Mukhopadhya, " Power and delay comparison between different types of full adder circuits",in International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 1, no. 3, 2012.

[4] S. Ghosh , P. Ndai and K. Roy, " A novel low overhead fault tolerant Kogge-Stone adder using adaptive clocking," Design, Automation and Test in Europe IEEE, 2008 .

[5] G. G., S. S. Raju and S. Suresh,"Parallel Prefix Speculative Han Carlson Adder," in IOSR Journal of Electronics and Communication, vol. 11, no.3, 2016.

[6] B. Koyada, N. Meghana., Md.O. Jaleel. andP. R. Jeripotula,"A Comparative Study on Adders," in IEEE Conf. on Wireless Communication, Signal Processing and Networking , 2017.