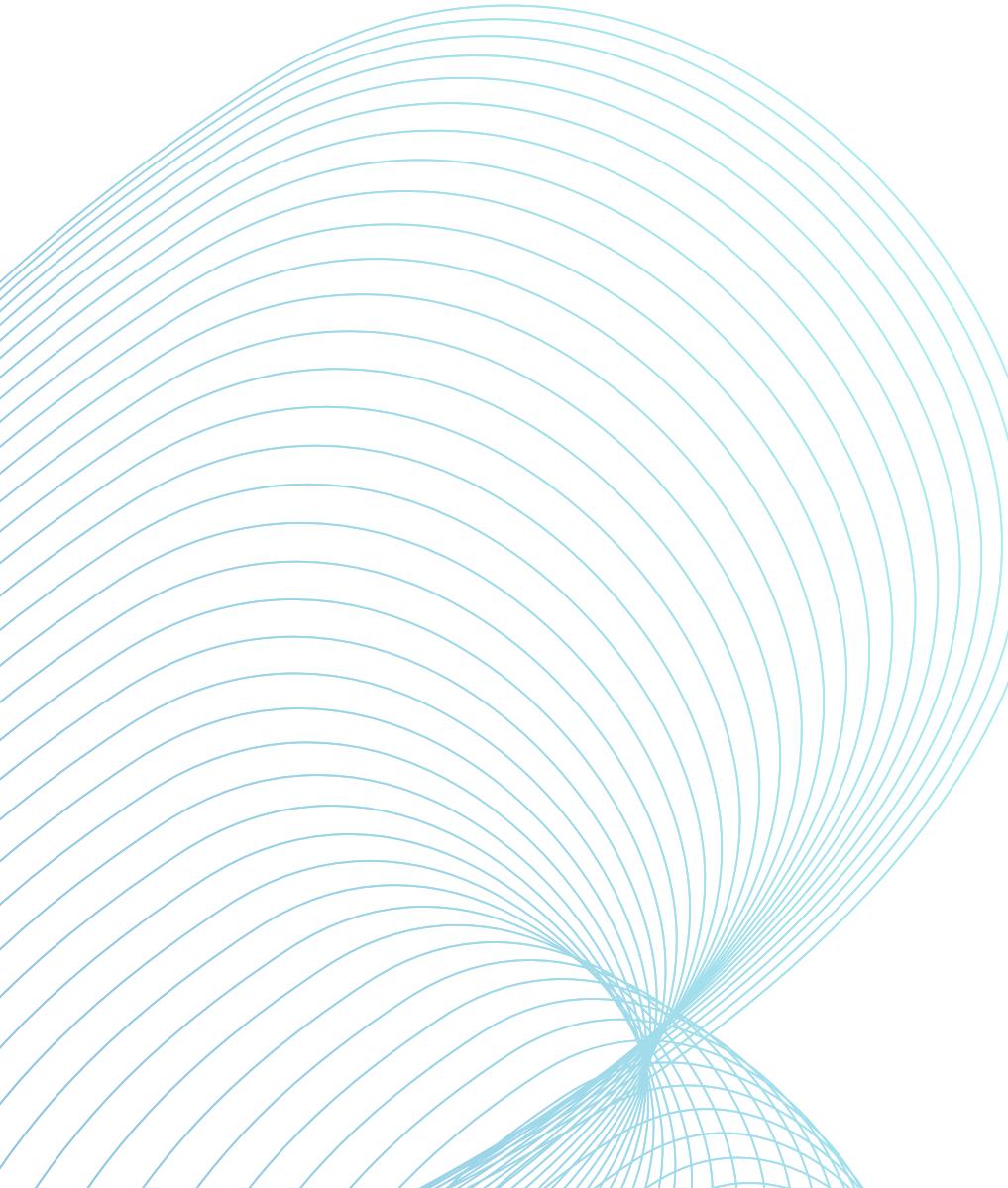


IMPLEMENTATION OF AREA, DELAY, AND ENERGY-EFFICIENT FULL DADDA MULTIPLIER



TEAM LEADER : TARANI J D

TEAM MEMBERS:

- **VAMSIKA S**
 - **SUGAPRIYA V**
- 

MULTIPLIER

- A digital multiplier is an important building block of any logical processor in a digital system, and some of the main specifications of such systems like processing speed, power consumption, and energy efficiency highly depend on it
- There is always a need to improve the performance of a multiplier to meet the requirements of fast and energy-efficient processes
- In digital image processing systems, convolution neural networks and general-purpose processes, the performance of a multiplier must be considered, especially where mathematical data evaluation is of higher priority

THREE SEGMENT OF MULTIPLICATION ALGORITHM

- 1.The first part is where two n-bit numbers are given to the inputs of AND gates to generate the partial products (PPs)
- 2.PPs layers are compressed using full adders (FAs) and half adders (HAs), unless only two layers of binary numbers remain
- 3.Two layers are added, usually by a RCA, in order to generate the final result of multiplication

PREVIOUS APPROACHES

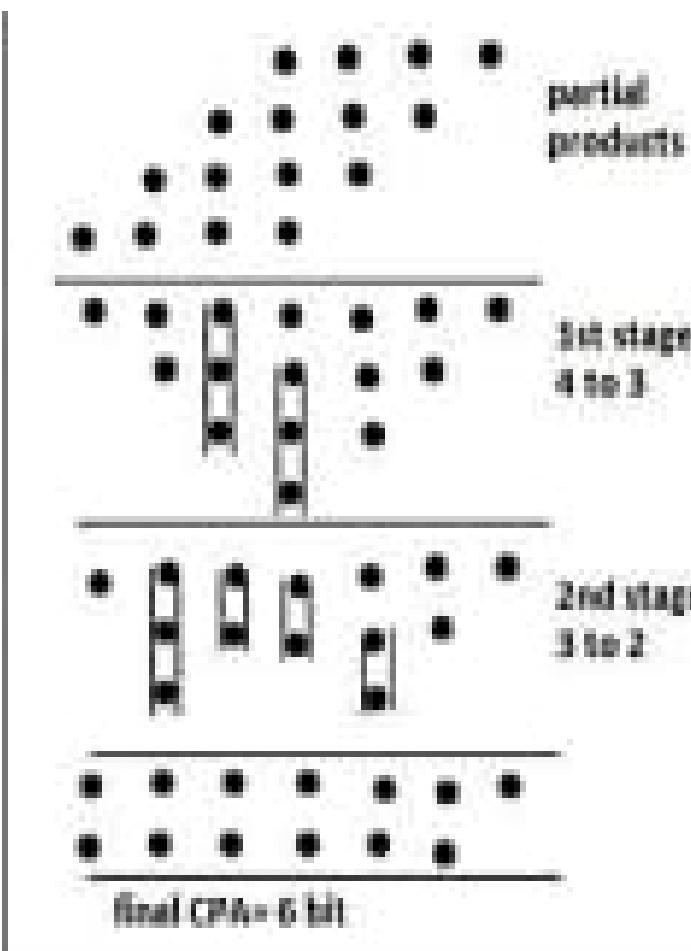
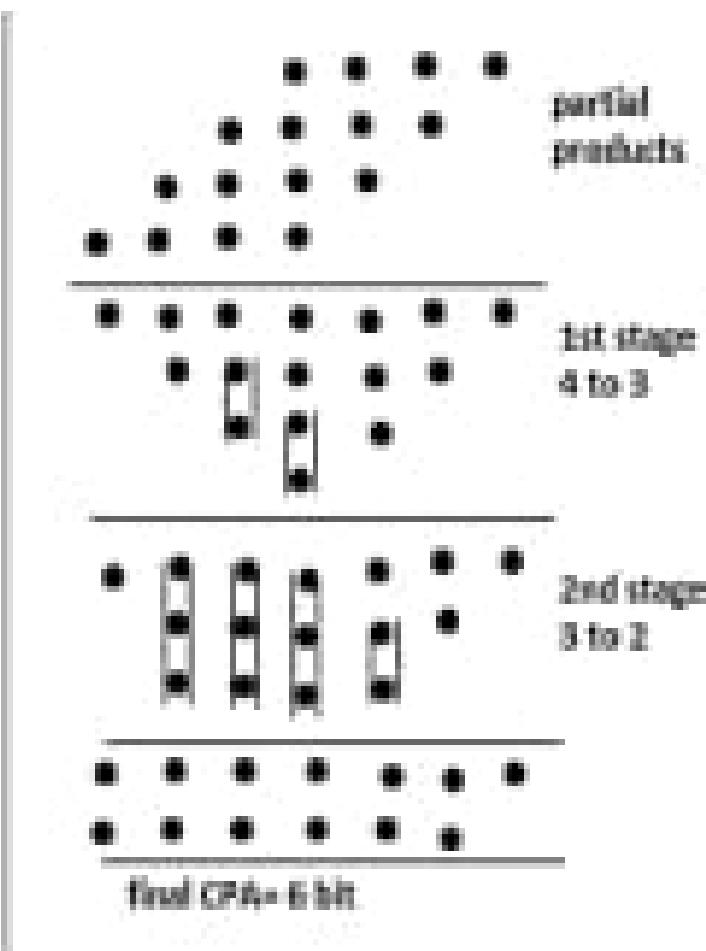
1. Parallel Prefix (PP) Multipliers
2. Approximate Adders
3. Attack-Based Multipliers
4. Novel Compressor-Based Techniques
5. Approximate Multipliers

- Each of these methods targets the crucial second step of multiplication, where the reduction of different layers of partial products (PPs) plays a vital role
- Approximate multipliers, in particular have emerged as more area and power-efficient alternatives to exact multipliers, albeit being suited for error-tolerant applications due to their potential error introduction.

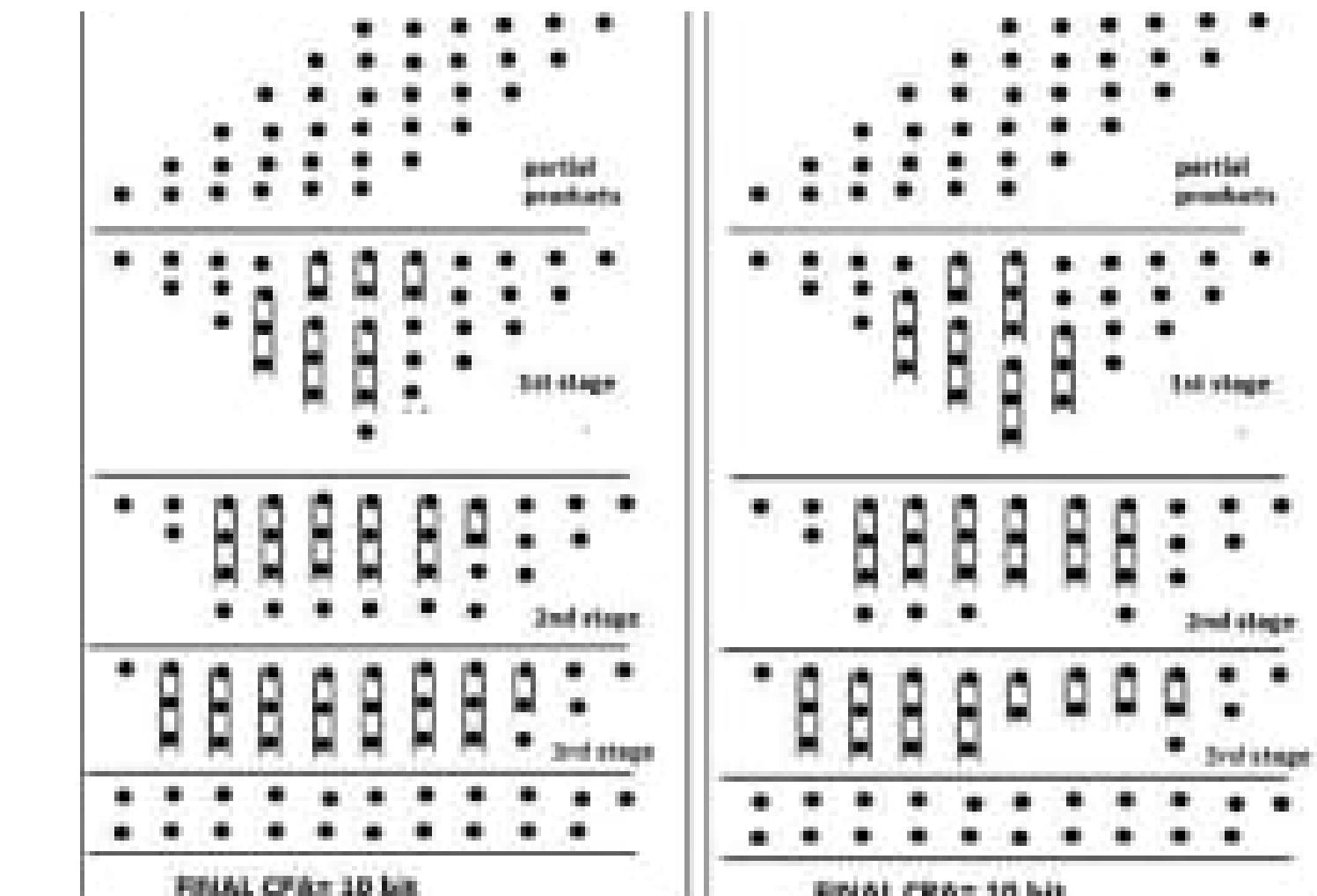
FULL DADDA ALGORITHM

- Our digital multiplier design is based on the modified Dadda algorithm, also known as the "full Dadda algorithm."
- To compress the layers of PPs in our design, we have introduced a novel 3:2 adder.
- This adder offers improved speed, area efficiency, and power-energy efficiency compared to traditional adders like carry look-ahead adders (CLAs) and simple full adders (FAs).
- Finally, we employ an improved Ripple Carry Adder (RCA) to calculate the final result.

COMPARSION OF THE DADDA AND FULL DADDA PATTERN



4-bit Dadda and full dadda

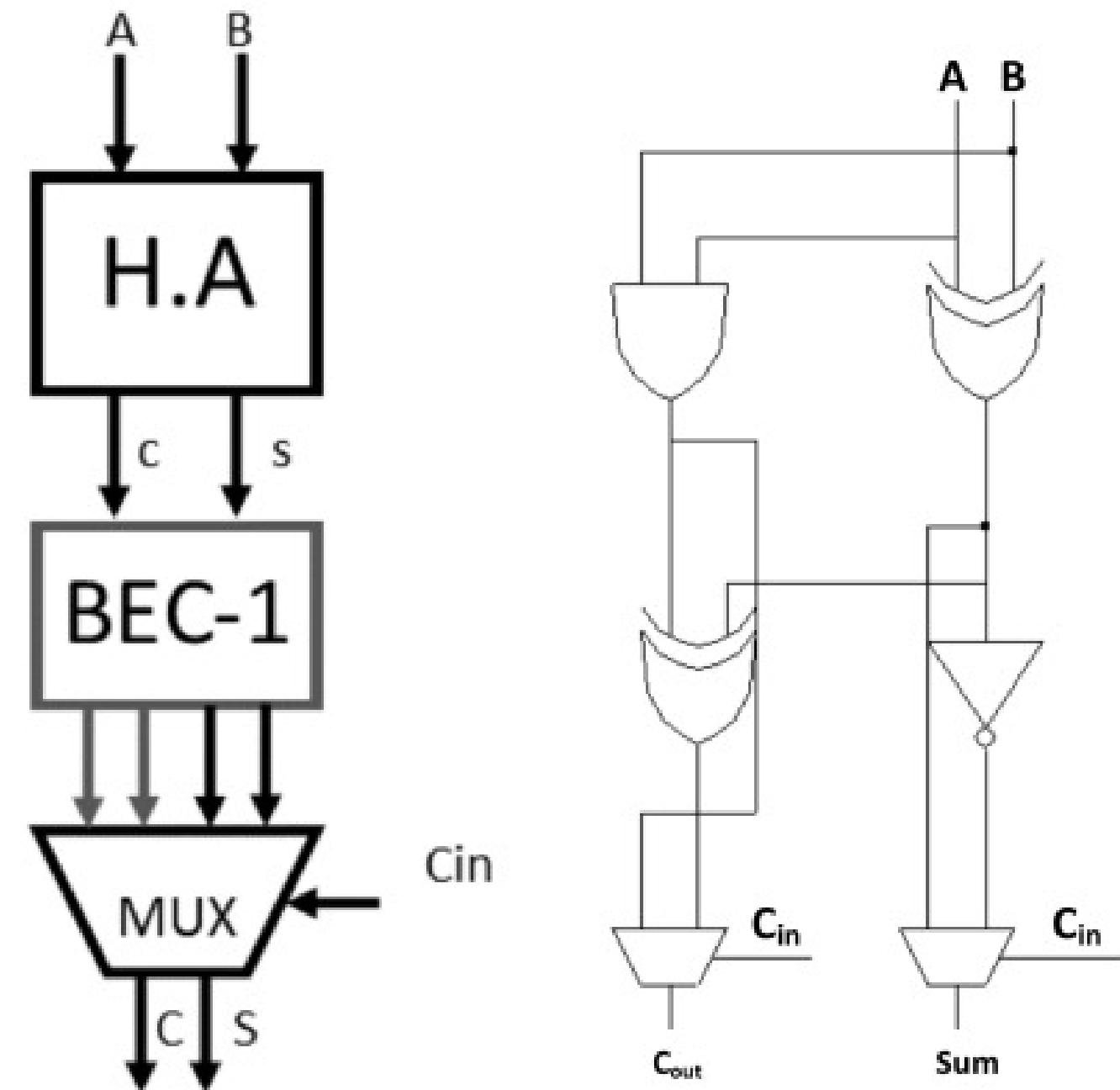


6-bit Dadda and full Dadda

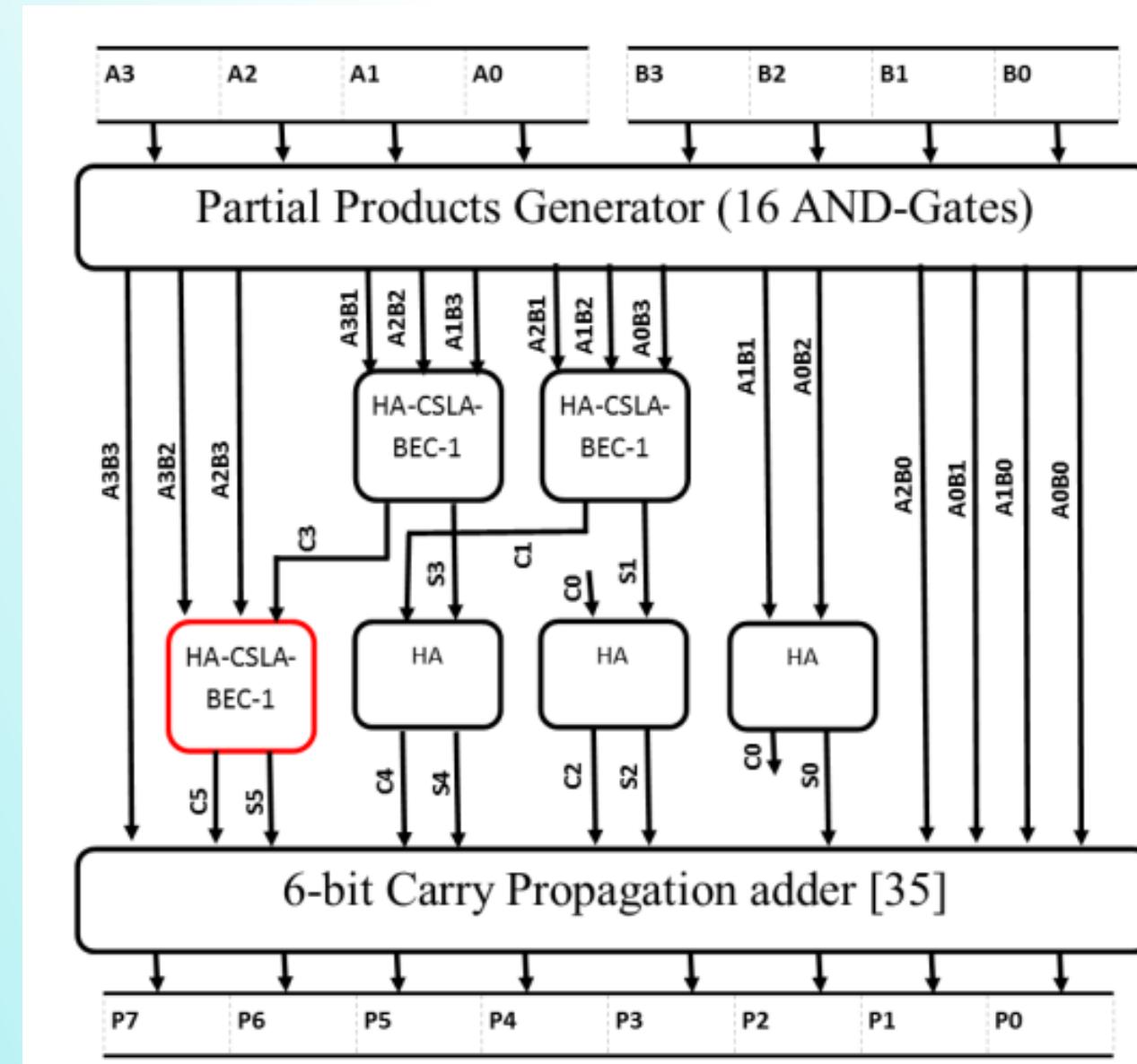
BENEFITS OF PROPOSED HA BASED CSLA-BEC-1

- Replaced Full Adders (FAs) with Half Adders (HAs) while maintaining CSA speed advantages.
- Eliminated horizontal interconnections between HA adders and Binary-to-Excess-1 Converter (BEC1), shifting carry propagation overhead to Multiplexer (MUX) stages.
- Implemented MUX using pass transistor logic (PTL) and saved a minimum of 18 transistors per 1-bit by replacing FAs with HAs.
- Reduced the overall number of transistors, resulting in a smaller layout area

1 BIT HA - CSLA - BEC - 1



FULL DADDA MULTIPLIER



VERILOG CODE:

```
module dadda_multiplier(s,a,b);
    output [7:0]s;
    input [3:0]a,b;
    wire [11:0]w;
    wire [5:0]x,y;
    wire [15:0]p;
    ppg p1(p,a,b);

    assign s[0]=p[0];
    HA_CSLA_BEC1 h1(p[8],p[7],p[6],w[0],w[1]);
    HA_CSLA_BEC1 h2(p[11],p[12],p[10],w[2],w[3]);
    HA_CSLA_BEC1 h3(w[3],p[14],p[13],w[10],w[11]);
    half_adder ha1(p[4],p[5],w[4],w[5]);
    half_adder ha2(p[9],w[0],w[6],w[7]);
    half_adder ha3(w[1],w[2],w[8],w[9]);

    assign x={w[11],w[9],w[7],w[5],p[3],p[1]},y=
    {p[15],w[10],w[8],w[6],w[4],p[2]};

    rca r1(x,y,1'b0,s[7:1]);
endmodule
```

```
module HA_CSLA_BEC1(A,B,Cin,Y,Cout);
    input A,B,Cin;
    output Y,Cout;
    wire a,b,c,d,e,f;
    half_adder h1(A,B,a,b);
    assign c= ~a;
    assign e = a^b;
    mux_2to1 m1(c,a,Cin,Y);
    mux_2to1 m2(e,b,Cin,Cout);
endmodule
```

```
module half_adder(A,B,Y,Cout);
    input A,B;
    output Y,Cout;
    assign {Cout,Y}=A+B;
endmodule
```

```
module ppg(p,a,b);
    input[3:0]a,b;
    output [15:0]p;
    and (p[0],a[0],b[0]),(p[1],a[1],b[0]),(p[2],a[0],b[1]),
    (p[3],a[2],b[0]),(p[4],a[0],b[2]),(p[5],a[1],b[1]),
    (p[6],a[2],b[1]),(p[7],a[1],b[2]),(p[8],a[0],b[3]),
    (p[9],a[3],b[0]),(p[10],a[3],b[1]),(p[11],a[1],b[3]),
    (p[12],a[2],b[2]),(p[13],a[3],b[2]),(p[14],a[2],b[3]),
    (p[15],a[3],b[3]);
endmodule
```

```
module rca(
    input [5:0]a,b,
    input cin,
    output [6:0]sum);
    wire c1,c2,c3,c4,c5; //Carry out of each full adder
    assign cin=1'b0;
    HA_CSLA_BEC1 fa0(a[0],b[0],cin,sum[0],c1);
    HA_CSLA_BEC1 fa1(a[1],b[1],c1,sum[1],c2);
    HA_CSLA_BEC1 fa2(a[2],b[2],c2,sum[2],c3);
    HA_CSLA_BEC1 fa3(a[3],b[3],c3,sum[3],c4);
    HA_CSLA_BEC1 fa4(a[4],b[4],c4,sum[4],c5);
    HA_CSLA_BEC1 fa5(a[5],b[5],c5,sum[5],sum[6]);
endmodule

module mux_2to1 (
    input wire A, // Input A
    input wire B, // Input B
    input wire S, // Select input
    output wire Y // Output
);
    assign Y = (S == 1'b1) ? A : B; // MUX logic
endmodule
```

STIMULATION RESULTS:

DADDA MULTIPLIER USING PROPOSED FULL ADDER:

Operating Conditions: sub0p03v125c Library: nsecd2lvt_sub0p03v125c
Wire Load Model: Poder: enclosed

Design	Wire Load Model	Library
dadda_multiplier	Poder	nsecd2lvt_sub0p03v125c
ppg	Poder	nsecd2lvt_sub0p03v125c
half_adder_0	Poder	nsecd2lvt_sub0p03v125c
HA_CSLA_BRCA_0	Poder	nsecd2lvt_sub0p03v125c
mca	Poder	nsecd2lvt_sub0p03v125c
full_adder_0	Poder	nsecd2lvt_sub0p03v125c
HA_CSLA_BRCA_1	Poder	nsecd2lvt_sub0p03v125c
HA_CSLA_BRCA_2	Poder	nsecd2lvt_sub0p03v125c
half_adder_1	Poder	nsecd2lvt_sub0p03v125c
half_adder_2	Poder	nsecd2lvt_sub0p03v125c
full_adder_1	Poder	nsecd2lvt_sub0p03v125c
full_adder_2	Poder	nsecd2lvt_sub0p03v125c
full_adder_3	Poder	nsecd2lvt_sub0p03v125c
full_adder_4	Poder	nsecd2lvt_sub0p03v125c
full_adder_5	Poder	nsecd2lvt_sub0p03v125c

Global Operating Voltage = 0.95

Power-specific unit information :

Voltage Units = mV

Capacitance Units = 1.000000000000000

Time Units = 1ns

Dynamic Power Units = 1udJ (derived from V,C,T units)]

Leakage Power Units = 1pW

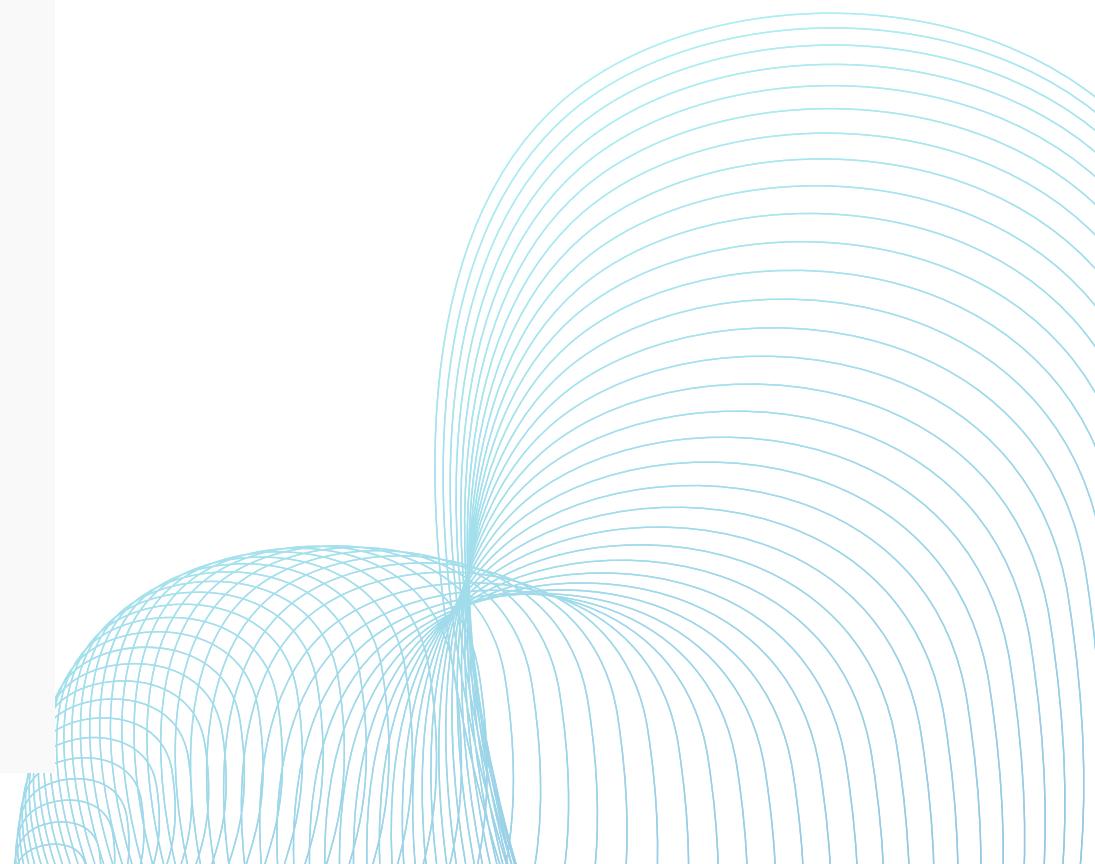
Cell Internal Power = 13.6550 uJ (72%)
Net Switching Power = 3.3679 uJ (28%)

Total Dynamic Power = 19.2329 uJ (100%)

Cell Leakage Power = 1.6932 uJ

Information: report_power power group summary does not include estimated clock tree power. (Pdb=700)

Power Group	Internal Power	Switching Power	Leakage Power	Total Power (uJ)	Active



DADDA MULTIPLIER USING PROPOSED ADDER:

Global Generating Values = 9.12

Power-specific unit information

Voltages Units ■ IV

Conductance Units = 1.000000

The Week 10

Dynamic Power Units = 1.00 (derived from V,C,T units)

Leakage Power Datas in Table

Cell Internal Power = 0.0000 mJ (0%)
Net Switching Power = 1.6573 mJ (100%)

Total Synoptic Power = 1,687.3 mW (1687)

FULLDADDA MULTIPLIER USING PROPOSED FULL ADDER:

Operating Conditions: Temperature: 25°C
Power supply voltage: 0.9V

Design	Wire load (nΩ)	Library
main_initializer	0.000	modified_solidarity
reg	0.000	modified_solidarity
ma_diva_0001_0	0.000	modified_solidarity
half_adder_0	0.000	modified_solidarity
reg	0.000	modified_solidarity
Full_adder_0	0.000	modified_solidarity
ma_diva_0001_1	0.000	modified_solidarity
ma_diva_0001_2	0.000	modified_solidarity
half_adder_1	0.000	modified_solidarity
half_adder_2	0.000	modified_solidarity
Full_adder_1	0.000	modified_solidarity
Full_adder_2	0.000	modified_solidarity
Full_adder_3	0.000	modified_solidarity
Full_adder_4	0.000	modified_solidarity
Full_adder_5	0.000	modified_solidarity

Global Operating voltage = 0.9V

Power-aware cell information:

voltage width = 37

Capacitance width = 1.000000

Time width = 100

Synthetic Power width = 100 (derived from V, C, T width)

Usage power width = 100

Cell internal Power = 14.400 nJ (100)

net switching power = 0.000 nJ (100)

.....

Total synthetic Power = 14.400 nJ (100)

Cell usage Power = 0.000 nJ

Information report: Net 0 has 0 fanout (This summary does not include unclocked sinks from user. (Par=1))

FULL DADDA MULTIPLIER USING PROPOSED ADDER:

Classical Overhead voltage = 8.9V
Power-supply with compensation =
voltage output = 10V
Capacitance output = 1.000000
Time output = 100.

FUTURE WORK

- Our team is committed to implementing the proposed multiplier using reversible logic gates
- This strategic shift holds the promise of significant advancements in various aspects of digital multipliers, particularly in terms of area utilization, delay reduction, and energy efficiency.
- This approach aligns with the growing demand for energy-efficient and high-performance integrated circuits, ensuring that our multiplier remains competitive in various applications, from processors to specialized hardware accelerators

THANK YOU!!

