MINI PROJECT

TITLE: Design and Implementation of 8x8 and 16x16 DADDA Multiplier using CSA in 45nm Technology

Course: Mini Project

Course Code: 23EECW301

Semester: V

Credits: 0-0-3

Guide: Prof. Suhas Shirol

Team Details				
Sl. No	**Roll No.**	**Div**	**SRN**	**Name**
1	524	E	01FE21BEC268	Prashanth S Aski
2	529	E	01FE21BEC273	Rachanna R U
3	530	E	01FE21BEC274	Sonal Sheth
4	553	E	01FE21BEC301	Srushti Maharajpet

CONTENT

- Problem Statement
- Introduction
- Literature Survey
  - Paper1
  - Paper2
  - Paper3
  - Paper4
  - Paper5
  - Paper6
- Results
- References

Problem Statement

Design and Implementation of 8x8 and 16x16 DADDA Multiplier using CSA in 45nm Technology

Introduction

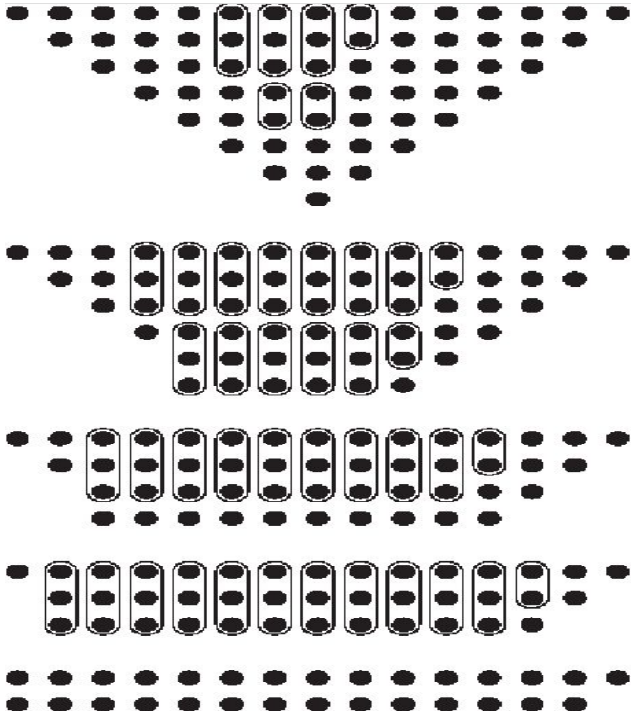
- Today, the use of portable electronic gadgets is growing every day, and these devices need batteries to function. In order to build such gadgets it is crucial to consider power dissipation. Addition multiplication and other operations are performed using different arithmetic and logical processes. When designing an optimal digital circuit, multipliers with lower latency, power consumption, and area are always employed to ensure that the maximum throughput is achieved with the shortest possible response time.

- The fundamental building elements of any multiplier design are full adders and half adders. To date, various half-adder and full-adder design architectures have been developed and put into use in order to reduce power consumption, area, and delay and produce an effective multiplier circuit. Along with this, several methods, like the Dadda algorithm, Wallace tree, Booth multiplier, and Vedic algorithms, have been developed to achieve optimal power, area, and latency.

Introduction

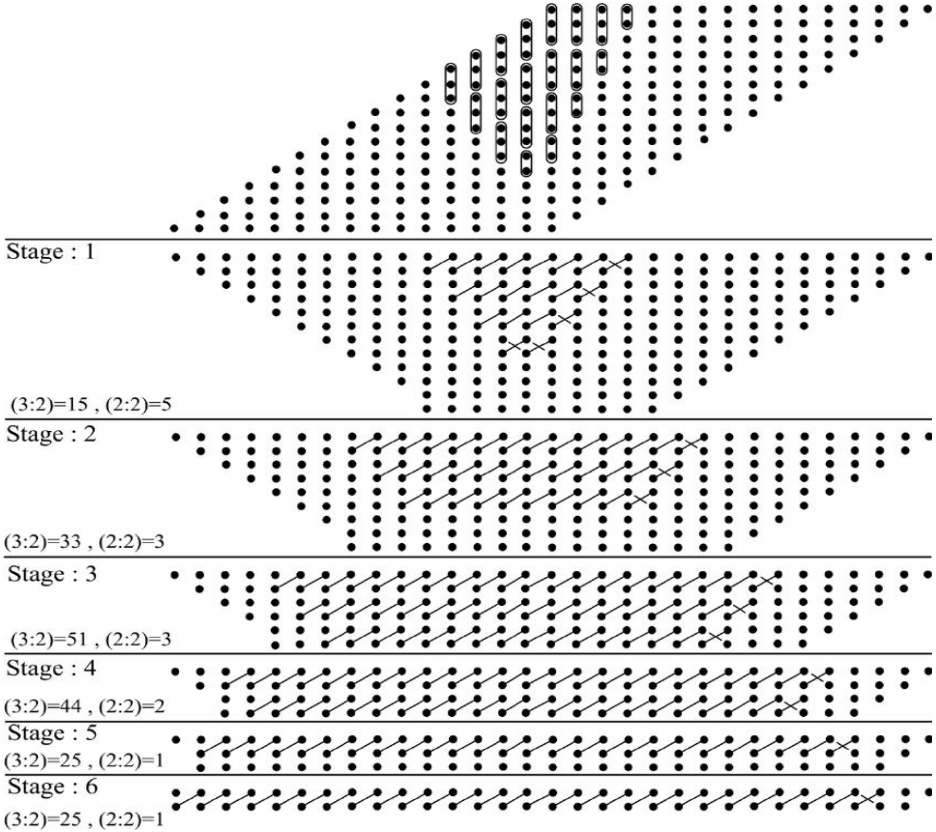
- The Dadda multiplier is a binary multiplier design invented by a scientist Luigi Dadda in 1965.It uses selection of full and half adders to sum the partial products in stages until 2 numbers are left .The design is similar to the Wallace multiplier but different reduction tree reduces the required number of gates and makes it slightly faster.

Introduction



Example of 8*8 DADDA Multiplication

Introduction



Example of 16*16 DADDA Multiplication

Literature survey : Paper 1

4*4 Dadda Multiplier

- Full adders and half adders have designed using pass transistor logic and CMOS process technology to reduce power dissipation and delay .
- The tool used for implementation is Cadance Virtuoso in 90nm technology . Starts its operation at 3.83GHz and 184uW with supply of 1V.
- In stage one 16 partial products are generated using AND gate.
- In stage 2 height of the tress is reduced using 1HA and 3FA.
- In stage 3 2HA and 3FA are used.
- In stage 4 we make use of ripple carry adder to sum up.
- Buffer is used to make output voltages better .

Literature Survey: Paper2

4*4 Dadda Multiplier using CSA and Binary to excess 3 converter

- It uses pass transistor logic.

•In the proposed paper, the Dadda tree reduction Algorithm is modified to use the Carry Select adder with
binary to excess 1 converter, instead of a normal Full Adder.

•The main block to discuss here is the CSA-BEX-1 block which is the core modification of the
proposed Dadda algorithm multiplier

•In CSA-BEX1 case, when carry is 0 CSA-BEX1 works same as a simple CSA but when the carry is 1, a binary
to excess-1 converter will be used, which is not an actual adder however it will add 1 carry in our sum
which will be even faster and low power implementation of an adder.

•In the end, Mux will be used which selects the final results on the basis of actual carry in Cin.

Literature Survey: Paper3

Design and Analysis of CMOS Based DADDA Multiplier

- Logic Styles for CMOS Based DADDA Multiplier: Two types of logic styles
  - SR-CPL is a high-speed logic style that uses a combination of n-channel MOS transistors (NMOS) and p-channel MOS transistors (PMOS) to implement logic gates. SR-CPL circuits are known for their high speed and low power consumption.
  - DPL is a more robust logic style that uses both NMOS and PMOS transistors to implement logic gates. DPL circuits are known for their high robustness to noise and their tolerance to process variations.

Literature Survey: Paper4

Feature	SR-CPL-based DADDA Multiplier	DPL-based DADDA Multiplier
Power Consumption	1.2 nW	1.5 nW
Delay	2.3 ns	2.8 ns
Area	100 μm^2	80 μm^2

Literature Survey: Paper4

Design and Implementation of 16-bit Low Power, Area Efficient Dadda Multiplier

- Full adder functionality is
  - S = (A ^ B ^ C) (7)
  - Co = ((A&B) | (B&C) | (C&A)) (8)

- The proposed full adder function is given by,
  - Functionality for generating sum bit,
    - W1 = ~ ((~ (A | B)) | (A & B)) (9)
    - S = ~ ((~ (W1 | C)) | (W1 & C)) (10)
  - The functionality for generating carry bit,
    - Co = ((A & B) | (W1 & C))

- The proposed Dadda multiplier, designed using the TSMC 65nm library, achieves a power reduction of 15.32%, an area reduction of 1.91%, and a timing delay of 4163 ps compared to the conventional design. With these improvements, the proposed Dadda multiplier offers a viable solution for power-constrained and area-sensitive DSP applications.

Literature Survey: Paper5

To perform the addition of intermediate bits in a multiplier, the authors of this paper have proposed FA circuit using pass transistor logic (PTL) and a HA circuit. The proposed PTL based FA is designed by using 10 transistors.
The usage of pass transistor logic has resulted in a significant lower number of transistors as a result, the area has also reduced.

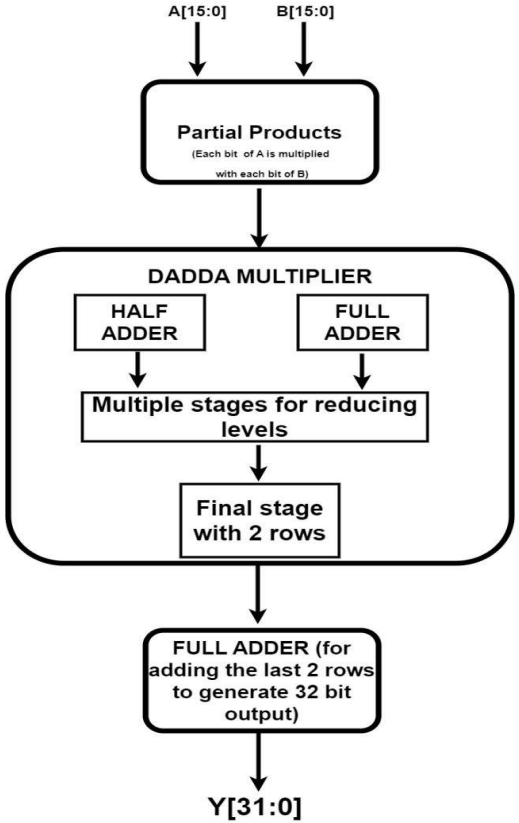
The analysis results suggest that the proposed multipliers offer

- A decrease in power up to ~47.6%,
- A decrease in delay up to ~63.96%,
- A decrease in transistor count up to ~58.7%

when compared with the CMOS based designs.

The multiplier is usually the slowest part of the circuit, hence its performance directly determines the system's total performance
For n bit multiplier, n(n-2) FA, n HA, and n^2 AND gates are required [5].

BLOCK DIAGRAM



RESULTS:

Table 1 Area, Delay and Power Dissipation of Adders

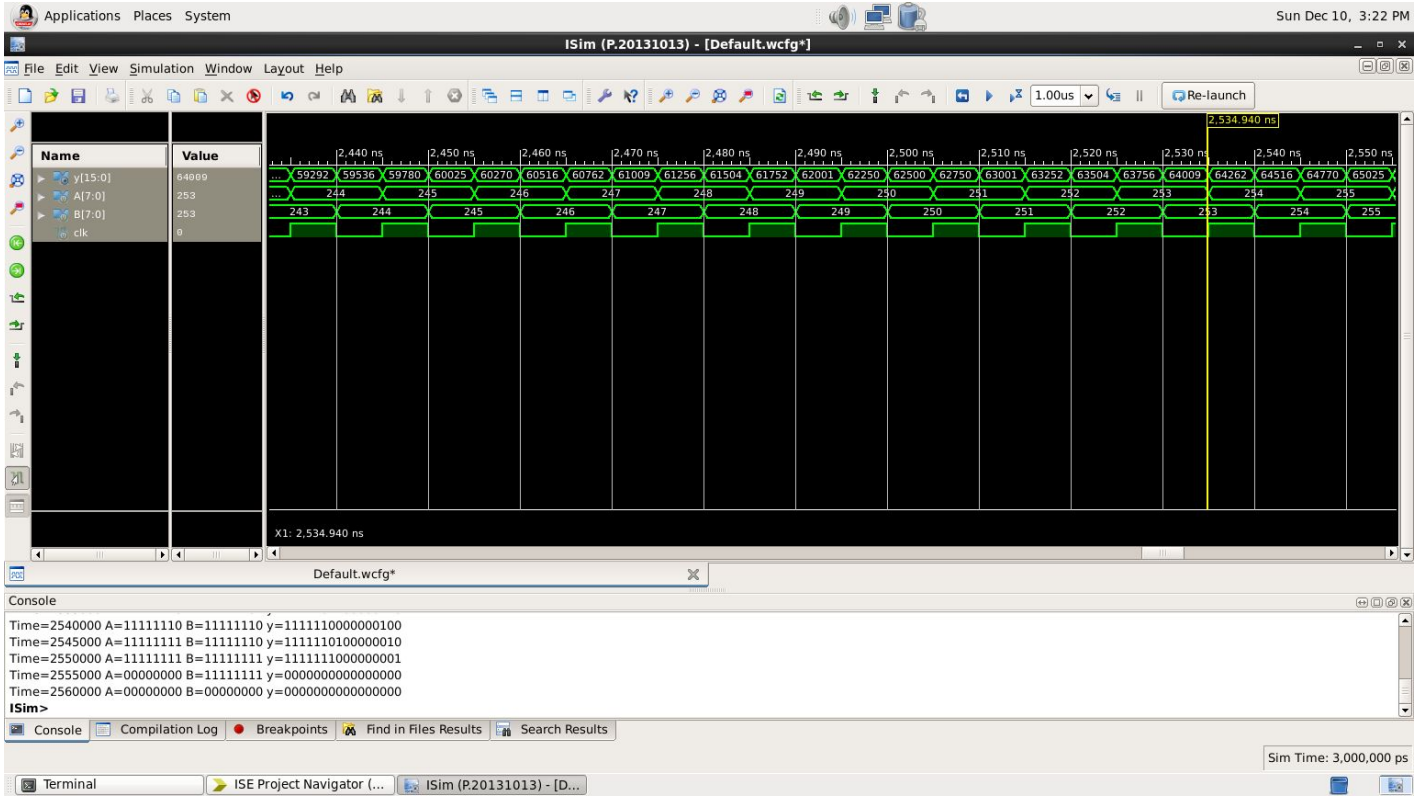
Adder topology	Gate count			Power dissipation (mW)	Area μm^2	Delay (ns)
	nMOS	pMOS	Total			
RCA	144	144	288	0.206	2214	4.208
CSaA	288	288	576	1.082	5904	2.924
CLA	136	136	272	0.312	2160	3.1
CIA	171	171	342	0.261	2793	2.880
CSkA	194	194	388	0.603	3486	3.022
CByA	186	186	372	0.459	3116	3.01
CSelA	300	300	600	1.109	6201	2.75

Table3 Energy Delay Parameters of Adders

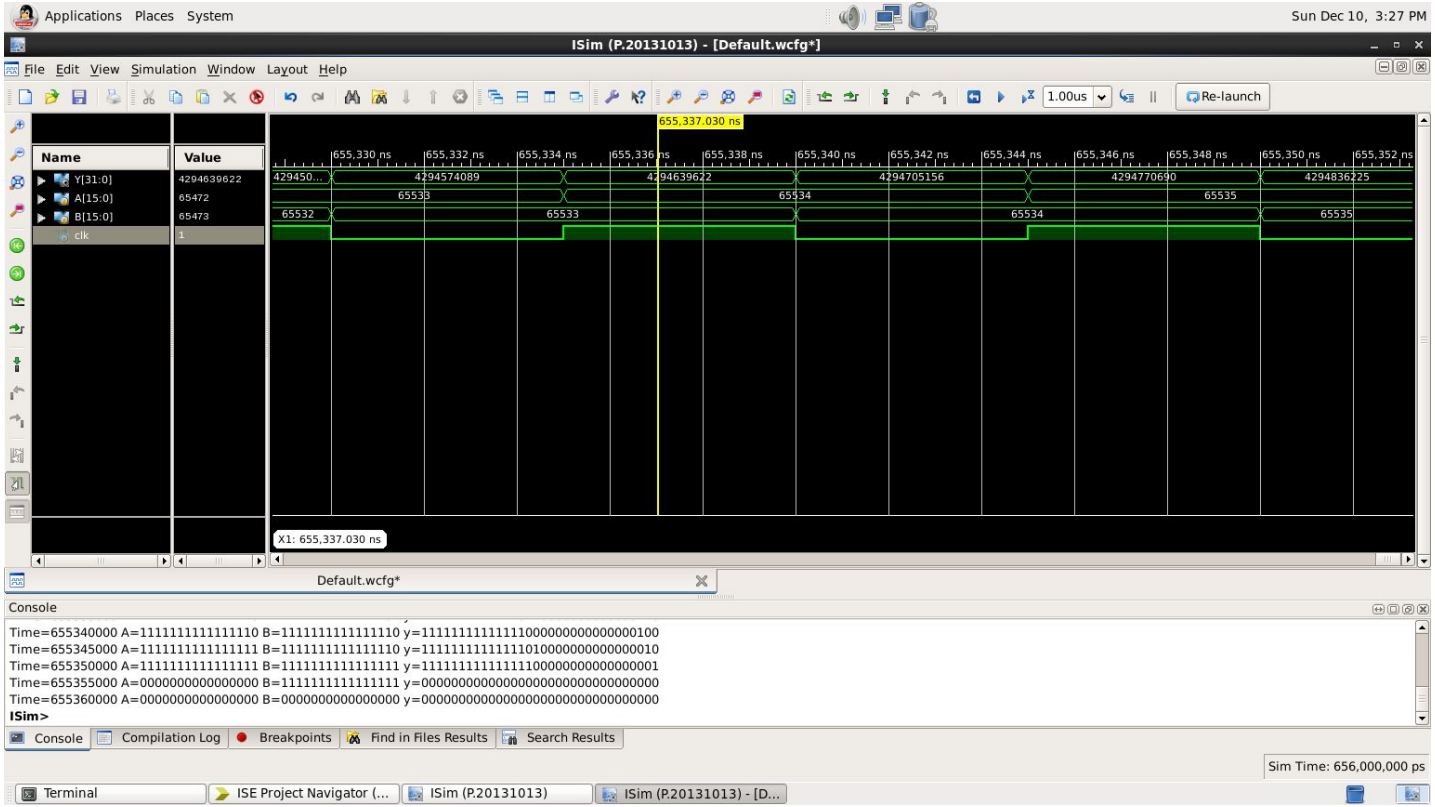
Adder topology	IDD MAX(mA)	IDD AVG(mA)	Rise and Fall delay (ns)
RCA	3.32	0.369	0.043
CSaA	4.820	0.657	0.024
CLA	2.389	0.200	0.04
CIA	4.453	0.355	0.020
CSkA	4.712	0.482	0.031
CByA	6.262	0.941	0.024
CSelA	7.765	1.151	0.017

Comparing various parameters such as Power Dissipation, Area Used and Propagation Delay of different adders we have decided to go ahead with Carry Select Adders for implementation of our DADDA Multiplier.

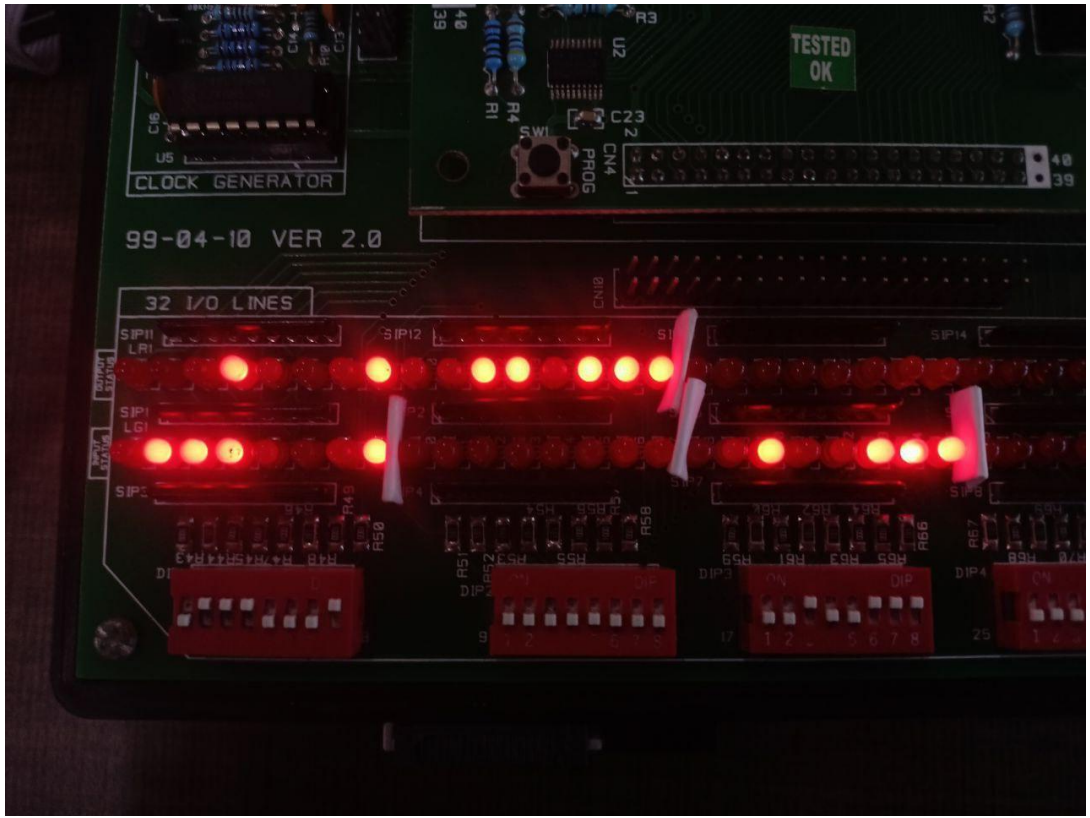
Simulation Results of 8x8 Multiplier



Simulation Results of 16x16 Multiplier



Title: 8x8 and 16x16 DADDA Multiplier using CSA



Hardware Results of 8x8 Multiplier

BINARY VALUES

A=01110001

B=00100111

Y=000100100110111

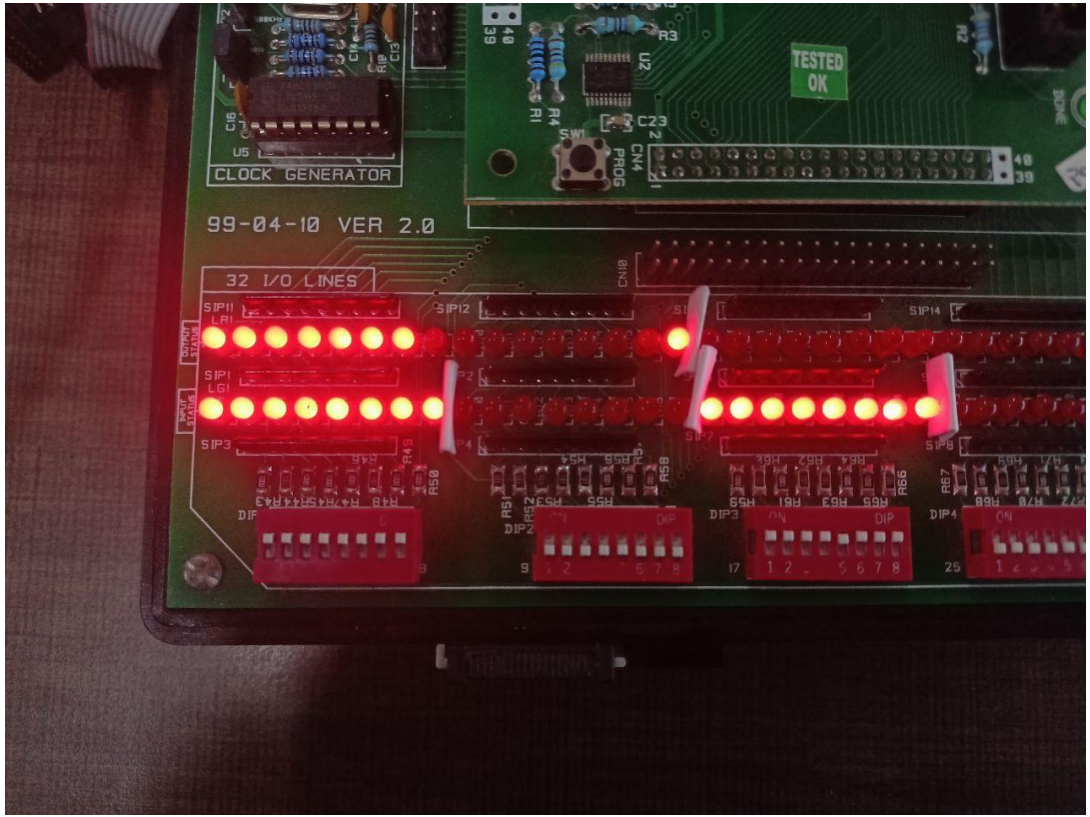
DECIMAL VALUES

A=113

B=39

Y=4407

Title: 8x8 and 16x16 DADDA Multiplier using CSA



Hardware Results of 8x8 Multiplier

BINARY VALUES

A=11111111

B=11111111

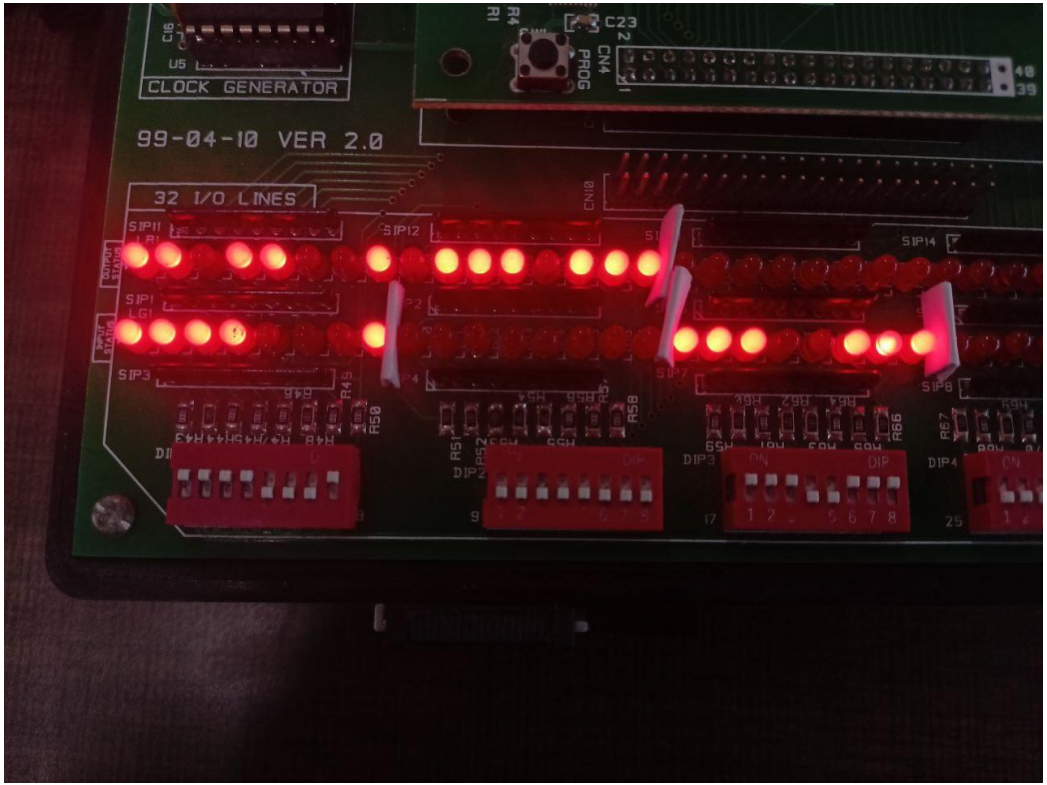
Y=1111111000000001

DECIMAL VALUES

A=255

B=255

Y=65025



Hardware Results of 8x8 Multiplier

BINARY VALUES

A=11110001

B=11100111

Y=110110010111110111

DECIMAL VALUES

A=241

B=231

Y=55671

DADDA 8X8(Area report)

```
==========================================================
 Generated by:          Genus(TM) Synthesis Solution 20.11-s111_1
 Generated on:          Jan 27 2024  10:19:10 am
 Module:                DADDA_8
 Technology library:    slow
 Operating conditions:  slow (balanced_tree)
 Wireload mode:         enclosed
 Area mode:             timing library
==========================================================

Instance  Module  Cell Count  Cell Area  Net Area  Total Area  Wireload
-----------------------------------------------------------------------
DADDA_8             136      0.000      0.000       0.000 <none> (D)
  c11   CSA           1      0.000      0.000       0.000 <none> (D)
  c12   CSA_211       1      0.000      0.000       0.000 <none> (D)
  c13   CSA_210       1      0.000      0.000       0.000 <none> (D)
  c21   CSA_209       1      0.000      0.000       0.000 <none> (D)
  c22   CSA_208       1      0.000      0.000       0.000 <none> (D)
  c23   CSA_207       1      0.000      0.000       0.000 <none> (D)
  c24   CSA_206       1      0.000      0.000       0.000 <none> (D)
  c25   CSA_205       1      0.000      0.000       0.000 <none> (D)
  c26   CSA_204       1      0.000      0.000       0.000 <none> (D)
  c32   CSA_196       1      0.000      0.000       0.000 <none> (D)
  c36   CSA_192       1      0.000      0.000       0.000 <none> (D)
  c37   CSA_191       1      0.000      0.000       0.000 <none> (D)
  c38   CSA_190       1      0.000      0.000       0.000 <none> (D)
  c39   CSA_189       1      0.000      0.000       0.000 <none> (D)
  c41   CSA_188       1      0.000      0.000       0.000 <none> (D)
  c42   CSA_187       1      0.000      0.000       0.000 <none> (D)
  c43   CSA_186       1      0.000      0.000       0.000 <none> (D)
  c44   CSA_185       1      0.000      0.000       0.000 <none> (D)
  c45   CSA_184       1      0.000      0.000       0.000 <none> (D)
  c46   CSA_183       1      0.000      0.000       0.000 <none> (D)
  c47   CSA_182       1      0.000      0.000       0.000 <none> (D)
  c48   CSA_181       1      0.000      0.000       0.000 <none> (D)
  c49   CSA_180       1      0.000      0.000       0.000 <none> (D)
  c51   CSA_177       1      0.000      0.000       0.000 <none> (D)
  c52   CSA_176       1      0.000      0.000       0.000 <none> (D)
  c54   CSA_175       1      0.000      0.000       0.000 <none> (D)
  c55   CSA_174       1      0.000      0.000       0.000 <none> (D)
  c56   CSA_173       1      0.000      0.000       0.000 <none> (D)
  c57   CSA_172       1      0.000      0.000       0.000 <none> (D)
  c58   CSA_171       1      0.000      0.000       0.000 <none> (D)
  c59   CSA_170       1      0.000      0.000       0.000 <none> (D)
  c210  CSA_200       1      0.000      0.000       0.000 <none> (D)
  c211  CSA_199       1      0.000      0.000       0.000 <none> (D)
  c212  CSA_198       1      0.000      0.000       0.000 <none> (D)
  c410  CSA_179       1      0.000      0.000       0.000 <none> (D)
  c411  CSA_178       1      0.000      0.000       0.000 <none> (D)
  c510  CSA_169       1      0.000      0.000       0.000 <none> (D)
  c511  CSA_168       1      0.000      0.000       0.000 <none> (D)
  c512  CSA_167       1      0.000      0.000       0.000 <none> (D)
  c513  CSA_166       1      0.000      0.000       0.000 <none> (D)
  c514  CSA_165       1      0.000      0.000       0.000 <none> (D)
  h1    HA            1      0.000      0.000       0.000 <none> (D)
  h2    HA_218        1      0.000      0.000       0.000 <none> (D)
  h3    HA_217        1      0.000      0.000       0.000 <none> (D)
  h4    HA_216        1      0.000      0.000       0.000 <none> (D)
  h5    HA_215        1      0.000      0.000       0.000 <none> (D)
  h6    HA_214        1      0.000      0.000       0.000 <none> (D)
  h7    HA_213        1      0.000      0.000       0.000 <none> (D)
  h8    HA_212        1      0.000      0.000       0.000 <none> (D)
(D) = wireload is default in technology library
```

DADDA 8X8(Power report)

Instance: /DADDA_8
Power Unit: W
PDB Frames: /stim#0/frame#0

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	3.56072e-08	2.24952e-05	9.91729e-06	3.24481e-05	100.00%
bbox	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
clock	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	3.56072e-08	2.24952e-05	9.91729e-06	3.24481e-05	100.00%
Percentage	0.11%	69.33%	30.56%	100.00%	100.00%

DADDA 8X8(Delay report)

```
==============================================================
 Generated by:          Genus(TM) Synthesis Solution 20.11-s111_1
 Generated on:          Jan 27 2024  11:10:11 am
 Module:                DADDA_8
 Operating conditions:  slow (balanced_tree)
 Wireload mode:         enclosed
 Area mode:             timing library
==============================================================
Path 1: UNCONSTRAINED
    Startpoint: (R) A[0]
     Endpoint: (F) y[14]
               Capture    Launch
     Drv Adjust:+      0       0
      Data Path:-    3059
#-------------------------------------------------------------------------------------
# Timing Point   Flags    Arc   Edge   Cell     Fanout Load Trans Delay Arrival Instance
#                                        (fF)  (ps)  (ps)   (ps)  Location
#-------------------------------------------------------------------------------------
 A[0]         -      -      R     (arrival)     1  5.4    0    0      0    (-,-)
 g966/Y       -      A->Y   F     CLKINVX8      8  6.4   23   14     14    (-,-)
 g932__8246/Y -      A->Y   R     NOR2X1        1  1.3   66   50     64    (-,-)
 h4/g28__5477/S -    A->S   R     ADDHX1        1  2.5   64  168    232    (-,-)
 c31/g94/S    -      CI->S  F     ADDFX4        1  2.5   69  316    549    (-,-)
 c42/g94/S    -      CI->S  R     ADDFX4        1  3.1   72  324    872    (-,-)
 c54/g2/CO    -      B->CO  R     ADDFX4        1  3.3   65  201   1074    (-,-)
 c55/g94/CO   -      A->CO  R     ADDFX4        1  2.5   62  199   1272    (-,-)
 c56/g2/CO    -      CI->CO R     ADDFX4        1  2.5   62  185   1457    (-,-)
 c57/g94/CO   -      CI->CO R     ADDFX4        1  2.5   62  185   1642    (-,-)
 c58/g94/CO   -      CI->CO R     ADDFX4        1  2.5   62  185   1826    (-,-)
 c59/g94/CO   -      CI->CO R     ADDFX4        1  2.5   62  185   2011    (-,-)
 c510/g2/CO   -      CI->CO R     ADDFX4        1  2.5   62  185   2195    (-,-)
 c511/g94/CO  -      CI->CO R     ADDFX4        1  2.5   62  185   2380    (-,-)
 c512/g2/CO   -      CI->CO R     ADDFX4        1  2.5   62  185   2564    (-,-)
 c513/g94/CO  -      CI->CO R     ADDFX4        1  2.5   62  185   2749    (-,-)
 c514/g94/S   -      CI->S  F     ADDFX4        1  0.0   59  310   3059    (-,-)
 y[14]        -      -      F     (port)        -  -     -    0   3059    (-,-)
#-------------------------------------------------------------------------------------
```

DADDA 16X16(Area report)

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:           Jan 24 2024  06:32:23 pm
  Module:                 DADDA_16
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Instance    Module    Cell Count  Cell Area  Net Area   Total Area   Wireload
-------------------------------------------------------------------------------
DADDA_16                   583      0.000     0.000        0.000 <none> (D)
  c_11   CSA_772             1      0.000     0.000        0.000 <none> (D)
  c_12   CSA_771             1      0.000     0.000        0.000 <none> (D)
  c_13   CSA_770             1      0.000     0.000        0.000 <none> (D)
  c_14   CSA_769             1      0.000     0.000        0.000 <none> (D)
  c_15   CSA_768             1      0.000     0.000        0.000 <none> (D)
  c_16   CSA_767             1      0.000     0.000        0.000 <none> (D)
  c_17   CSA_766             1      0.000     0.000        0.000 <none> (D)
  c_18   CSA_765             1      0.000     0.000        0.000 <none> (D)
  c_19   CSA_764             1      0.000     0.000        0.000 <none> (D)
  c_22   CSA_756             1      0.000     0.000        0.000 <none> (D)
  c_23   CSA_755             1      0.000     0.000        0.000 <none> (D)
  c_24   CSA_754             1      0.000     0.000        0.000 <none> (D)
  c_25   CSA_753             1      0.000     0.000        0.000 <none> (D)
  c_26   CSA_752             1      0.000     0.000        0.000 <none> (D)
  c_27   CSA_751             1      0.000     0.000        0.000 <none> (D)
  c_28   CSA_750             1      0.000     0.000        0.000 <none> (D)
  c_29   CSA_749             1      0.000     0.000        0.000 <none> (D)
  c_110  CSA_763             1      0.000     0.000        0.000 <none> (D)
  c_111  CSA_762             1      0.000     0.000        0.000 <none> (D)
  c_112  CSA_761             1      0.000     0.000        0.000 <none> (D)
  c_113  CSA_760             1      0.000     0.000        0.000 <none> (D)
  c_114  CSA_759             1      0.000     0.000        0.000 <none> (D)
  c_115  CSA_758             1      0.000     0.000        0.000 <none> (D)
  c_116  CSA_757             1      0.000     0.000        0.000 <none> (D)
  c_210  CSA_748             1      0.000     0.000        0.000 <none> (D)
  c_211  CSA_747             1      0.000     0.000        0.000 <none> (D)
  c_212  CSA_746             1      0.000     0.000        0.000 <none> (D)
  c_213  CSA_745             1      0.000     0.000        0.000 <none> (D)
  c_214  CSA_744             1      0.000     0.000        0.000 <none> (D)
  c_215  CSA_743             1      0.000     0.000        0.000 <none> (D)
  c_216  CSA_742             1      0.000     0.000        0.000 <none> (D)
  d1     DADDA_8           136      0.000     0.000        0.000 <none> (D)
    c11  CSA                 1      0.000     0.000        0.000 <none> (D)
    c12  CSA_963             1      0.000     0.000        0.000 <none> (D)
    c13  CSA_962             1      0.000     0.000        0.000 <none> (D)
    c21  CSA_961             1      0.000     0.000        0.000 <none> (D)
    c22  CSA_960             1      0.000     0.000        0.000 <none> (D)
    c23  CSA_959             1      0.000     0.000        0.000 <none> (D)
    c24  CSA_958             1      0.000     0.000        0.000 <none> (D)
    c25  CSA_957             1      0.000     0.000        0.000 <none> (D)
    c26  CSA_956             1      0.000     0.000        0.000 <none> (D)
    c27  CSA_955             1      0.000     0.000        0.000 <none> (D)
    c28  CSA_954             1      0.000     0.000        0.000 <none> (D)
    c29  CSA_953             1      0.000     0.000        0.000 <none> (D)
    c31  CSA_949             1      0.000     0.000        0.000 <none> (D)
    c32  CSA_948             1      0.000     0.000        0.000 <none> (D)
    c33  CSA_947             1      0.000     0.000        0.000 <none> (D)
    c34  CSA_946             1      0.000     0.000        0.000 <none> (D)
    c35  CSA_945             1      0.000     0.000        0.000 <none> (D)
    c36  CSA_944             1      0.000     0.000        0.000 <none> (D)
    c37  CSA_943             1      0.000     0.000        0.000 <none> (D)
    c38  CSA_942             1      0.000     0.000        0.000 <none> (D)
    c39  CSA_941             1      0.000     0.000        0.000 <none> (D)
    c41  CSA_940             1      0.000     0.000        0.000 <none> (D)
    c42  CSA_939             1      0.000     0.000        0.000 <none> (D)
    c43  CSA_938             1      0.000     0.000        0.000 <none> (D)
    c44  CSA_937             1      0.000     0.000        0.000 <none> (D)
    c45  CSA_936             1      0.000     0.000        0.000 <none> (D)
    c46  CSA_935             1      0.000     0.000        0.000 <none> (D)
    c47  CSA_934             1      0.000     0.000        0.000 <none> (D)
    c48  CSA_933             1      0.000     0.000        0.000 <none> (D)
    c49  CSA_932             1      0.000     0.000        0.000 <none> (D)
    c51  CSA_929             1      0.000     0.000        0.000 <none> (D)
    c52  CSA_928             1      0.000     0.000        0.000 <none> (D)
    c38  CSA_798             1      0.000     0.000        0.000 <none> (D)
    c39  CSA_797             1      0.000     0.000        0.000 <none> (D)
    c41  CSA_796             1      0.000     0.000        0.000 <none> (D)
    c42  CSA_795             1      0.000     0.000        0.000 <none> (D)
    c43  CSA_794             1      0.000     0.000        0.000 <none> (D)
    c44  CSA_793             1      0.000     0.000        0.000 <none> (D)
    c45  CSA_792             1      0.000     0.000        0.000 <none> (D)
    c46  CSA_791             1      0.000     0.000        0.000 <none> (D)
    c47  CSA_790             1      0.000     0.000        0.000 <none> (D)
    c48  CSA_789             1      0.000     0.000        0.000 <none> (D)
    c49  CSA_788             1      0.000     0.000        0.000 <none> (D)
    c51  CSA_785             1      0.000     0.000        0.000 <none> (D)
    c52  CSA_784             1      0.000     0.000        0.000 <none> (D)
    c54  CSA_783             1      0.000     0.000        0.000 <none> (D)
    c55  CSA_782             1      0.000     0.000        0.000 <none> (D)
    c56  CSA_781             1      0.000     0.000        0.000 <none> (D)
    c57  CSA_780             1      0.000     0.000        0.000 <none> (D)
    c58  CSA_779             1      0.000     0.000        0.000 <none> (D)
    c59  CSA_778             1      0.000     0.000        0.000 <none> (D)
    c210 CSA_808             1      0.000     0.000        0.000 <none> (D)
    c211 CSA_807             1      0.000     0.000        0.000 <none> (D)
    c212 CSA_806             1      0.000     0.000        0.000 <none> (D)
    c410 CSA_787             1      0.000     0.000        0.000 <none> (D)
    c411 CSA_786             1      0.000     0.000        0.000 <none> (D)
    c510 CSA_777             1      0.000     0.000        0.000 <none> (D)
    c511 CSA_776             1      0.000     0.000        0.000 <none> (D)
    c512 CSA_775             1      0.000     0.000        0.000 <none> (D)
    c513 CSA_774             1      0.000     0.000        0.000 <none> (D)
    c514 CSA_773             1      0.000     0.000        0.000 <none> (D)
    h1   HA_979              1      0.000     0.000        0.000 <none> (D)
    h2   HA_978              1      0.000     0.000        0.000 <none> (D)
    h3   HA_977              1      0.000     0.000        0.000 <none> (D)
    h4   HA_976              1      0.000     0.000        0.000 <none> (D)
    h5   HA_975              1      0.000     0.000        0.000 <none> (D)
    h6   HA_974              1      0.000     0.000        0.000 <none> (D)
    h7   HA_973              1      0.000     0.000        0.000 <none> (D)
    h8   HA_972              1      0.000     0.000        0.000 <none> (D)
  h1     HA_971              1      0.000     0.000        0.000 <none> (D)
  h2     HA_970              1      0.000     0.000        0.000 <none> (D)
  h3     HA_969              1      0.000     0.000        0.000 <none> (D)
  h4     HA_968              1      0.000     0.000        0.000 <none> (D)
  h5     HA_967              1      0.000     0.000        0.000 <none> (D)
  h6     HA_966              1      0.000     0.000        0.000 <none> (D)
  h7     HA_965              1      0.000     0.000        0.000 <none> (D)
  h8     HA_964              1      0.000     0.000        0.000 <none> (D)
(D) = wireload is default in technology library
```

DADDA 16X16(Power report)

Instance: /DADDA_16
Power Unit: W
PDB Frames: /stim#0/frame#0

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	1.59905e-07	1.38303e-04	5.44857e-05	1.92949e-04	100.00%
bbox	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
clock	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	1.59905e-07	1.38303e-04	5.44857e-05	1.92949e-04	100.00%
Percentage	0.08%	71.68%	28.24%	100.00%	100.00%

DADDA 16X16(Power report)

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:           Jan 24 2024  06:31:50 pm
  Module:                 DADDA_16
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================
Path 1: UNCONSTRAINED
    Startpoint: (R) A[0]
      Endpoint: (F) Y[31]
                Capture    Launch
     Drv Adjust:+      0       0
      Data Path:-    6344
#-------------------------------------------------------------------------------------------
#  Timing Point    Flags    Arc   Edge   Cell     Fanout Load Trans Delay Arrival Instance
#                                          (fF)  (ps)  (ps)   (ps)  Location
#-------------------------------------------------------------------------------------------
  A[0]            -      -     R     (arrival)     2 10.8    0     0      0    (-,-)
  d1/g966/Y       -      A->Y  F     CLKINVX8      8  6.4   23    14     14    (-,-)
  d1/g932__5107/Y -      A->Y  R     NOR2X1        1  1.3   66    50     64    (-,-)
  d1/h4/g28__5115/S -    A->S  R     ADDHX1        1  2.5   64   168    232    (-,-)
  d1/c31/g89/S    -      CI->S F     ADDFX4        1  2.5   69   316    549    (-,-)
  d1/c42/g89/S    -      CI->S R     ADDFX4        1  2.5   70   323    871    (-,-)
  d1/c54/g89/CO   -      CI->CO R    ADDFX4        1  2.5   62   187   1058    (-,-)
  d1/c55/g89/CO   -      CI->CO R    ADDFX4        1  2.5   62   185   1243    (-,-)
  d1/c56/g89/CO   -      CI->CO R    ADDFX4        1  2.5   62   185   1428    (-,-)
  d1/c57/g89/CO   -      CI->CO R    ADDFX4        1  2.5   62   185   1612    (-,-)
  d1/c58/g89/CO   -      CI->CO R    ADDFX4        1  2.5   62   185   1797    (-,-)
  d1/c59/g89/CO   -      CI->CO R    ADDFX4        1  2.5   62   185   1981    (-,-)
  d1/c510/g89/CO  -      CI->CO R    ADDFX4        1  2.5   62   185   2166    (-,-)
  d1/c511/g89/S   -      CI->S F     ADDFX4        1  2.5   69   316   2482    (-,-)
  c_14/g89/S      -      CI->S R     ADDFX4        1  3.1   72   324   2805    (-,-)
  c_23/g89/CO     -      B->CO R     ADDFX4        1  2.5   62   199   3005    (-,-)
  c_24/g89/CO     -      CI->CO R    ADDFX4        1  2.5   62   185   3189    (-,-)
  c_25/g89/CO     -      CI->CO R    ADDFX4        1  2.5   62   185   3374    (-,-)
  c_26/g89/CO     -      CI->CO R    ADDFX4        1  2.5   62   185   3559    (-,-)
  c_27/g89/CO     -      CI->CO R    ADDFX4        1  2.5   62   185   3743    (-,-)
  c_28/g89/CO     -      CI->CO R    ADDFX4        1  2.5   62   185   3928    (-,-)
  c_29/g89/CO     -      CI->CO R    ADDFX4        1  2.5   62   185   4112    (-,-)
  c_210/g89/CO    -      CI->CO R    ADDFX4        1  2.5   62   185   4297    (-,-)
  c_211/g89/CO    -      CI->CO R    ADDFX4        1  2.5   62   185   4482    (-,-)
  c_212/g89/CO    -      CI->CO R    ADDFX4        1  2.5   62   185   4666    (-,-)
  c_213/g89/CO    -      CI->CO R    ADDFX4        1  2.5   62   185   4851    (-,-)
  c_214/g89/CO    -      CI->CO R    ADDFX4        1  2.5   62   185   5035    (-,-)
  c_215/g89/CO    -      CI->CO R    ADDFX4        1  2.5   62   185   5220    (-,-)
  c_216/g89/CO    -      CI->CO R    ADDFX4        1  1.6   59   183   5403    (-,-)
  h2/g28__6783/CO -      B->CO  R    ADDHX1        1  1.6   52   130   5532    (-,-)
  h3/g28__3680/CO -      B->CO  R    ADDHX1        1  1.6   52   127   5660    (-,-)
  h4/g28__1617/CO -      B->CO  R    ADDHX1        1  1.6   52   127   5787    (-,-)
  h5/g28__2802/CO -      B->CO  R    ADDHX1        1  1.6   52   127   5914    (-,-)
  h6/g28__1705/CO -      B->CO  R    ADDHX1        1  1.6   52   127   6041    (-,-)
  h7/g28__5122/CO -      B->CO  R    ADDHX1        1  1.4   49   126   6167    (-,-)
  h8/g2__8246/Y   -      B->Y   F    XOR2X4        1  0.0   55   177   6344    (-,-)
  Y[31]           -      -      F    (port)        -  -     -    0     6344    (-,-)
#-------------------------------------------------------------------------------------------
```

DADDA 16X16(Power report)

```
====================================================================
 Generated by:          Genus(TM) Synthesis Solution 20.11-s111_1
 Generated on:          Jan 24 2024  06:31:50 pm
 Module:                DADDA_16
 Operating conditions:  slow (balanced_tree)
 Wireload mode:         enclosed
 Area mode:             timing library
====================================================================
Path 1: UNCONSTRAINED
   Startpoint: (R) A[0]
    Endpoint: (F) Y[31]
              Capture    Launch
    Drv Adjust:+      0        0
     Data Path:-    6344
#-------------------------------------------------------------------------------------------
#  Timing Point    Flags    Arc   Edge   Cell     Fanout Load Trans Delay Arrival Instance
#                                          (fF)  (ps)  (ps)   (ps)  Location
#-------------------------------------------------------------------------------------------
 A[0]           -     -     R     (arrival)      2 10.8     0     0      0    (-,-)
 d1/g966/Y      -     A->Y  F     CLKINVX8       8  6.4    23    14     14    (-,-)
 d1/g932__5107/Y -    A->Y  R     NOR2X1         1  1.3    66    50     64    (-,-)
 d1/h4/g28__5115/S -  A->S  R     ADDHX1         1  2.5    64   168    232    (-,-)
 d1/c31/g89/S   -     CI->S F     ADDFX4         1  2.5    69   316    549    (-,-)
 d1/c42/g89/S   -     CI->S R     ADDFX4         1  2.5    70   323    871    (-,-)
 d1/c54/g89/CO  -     CI->CO R    ADDFX4         1  2.5    62   187   1058    (-,-)
 d1/c55/g89/CO  -     CI->CO R    ADDFX4         1  2.5    62   185   1243    (-,-)
 d1/c56/g89/CO  -     CI->CO R    ADDFX4         1  2.5    62   185   1428    (-,-)
 d1/c57/g89/CO  -     CI->CO R    ADDFX4         1  2.5    62   185   1612    (-,-)
 d1/c58/g89/CO  -     CI->CO R    ADDFX4         1  2.5    62   185   1797    (-,-)
 d1/c59/g89/CO  -     CI->CO R    ADDFX4         1  2.5    62   185   1981    (-,-)
 d1/c510/g89/CO -     CI->CO R    ADDFX4         1  2.5    62   185   2166    (-,-)
 d1/c511/g89/S  -     CI->S  F    ADDFX4         1  2.5    69   316   2482    (-,-)
 c_14/g89/S     -     CI->S  R    ADDFX4         1  3.1    72   324   2805    (-,-)
 c_23/g89/CO    -     B->CO  R    ADDFX4         1  2.5    62   199   3005    (-,-)
 c_24/g89/CO    -     CI->CO R    ADDFX4         1  2.5    62   185   3189    (-,-)
 c_25/g89/CO    -     CI->CO R    ADDFX4         1  2.5    62   185   3374    (-,-)
 c_26/g89/CO    -     CI->CO R    ADDFX4         1  2.5    62   185   3559    (-,-)
 c_27/g89/CO    -     CI->CO R    ADDFX4         1  2.5    62   185   3743    (-,-)
 c_28/g89/CO    -     CI->CO R    ADDFX4         1  2.5    62   185   3928    (-,-)
 c_29/g89/CO    -     CI->CO R    ADDFX4         1  2.5    62   185   4112    (-,-)
 c_210/g89/CO   -     CI->CO R    ADDFX4         1  2.5    62   185   4297    (-,-)
 c_211/g89/CO   -     CI->CO R    ADDFX4         1  2.5    62   185   4482    (-,-)
 c_212/g89/CO   -     CI->CO R    ADDFX4         1  2.5    62   185   4666    (-,-)
 c_213/g89/CO   -     CI->CO R    ADDFX4         1  2.5    62   185   4851    (-,-)
 c_214/g89/CO   -     CI->CO R    ADDFX4         1  2.5    62   185   5035    (-,-)
 c_215/g89/CO   -     CI->CO R    ADDFX4         1  2.5    62   185   5220    (-,-)
 c_216/g89/CO   -     CI->CO R    ADDFX4         1  1.6    59   183   5403    (-,-)
 h2/g28__6783/CO -    B->CO  R    ADDHX1         1  1.6    52   130   5532    (-,-)
 h3/g28__3680/CO -    B->CO  R    ADDHX1         1  1.6    52   127   5660    (-,-)
 h4/g28__1617/CO -    B->CO  R    ADDHX1         1  1.6    52   127   5787    (-,-)
 h5/g28__2802/CO -    B->CO  R    ADDHX1         1  1.6    52   127   5914    (-,-)
 h6/g28__1705/CO -    B->CO  R    ADDHX1         1  1.6    52   127   6041    (-,-)
 h7/g28__5122/CO -    B->CO  R    ADDHX1         1  1.4    49   126   6167    (-,-)
 h8/g2__8246/Y   -    B->Y   F    XOR2X4         1  0.0    55   177   6344    (-,-)
 Y[31]           -    -      F    (port)         -  -      -     0   6344    (-,-)
#-------------------------------------------------------------------------------------------
```

RESULTS

	AREA	POWER (μW)				DELAY
	(Cell count)	Leakage	Internal	Switching	Total Power	(ps)
8-Bit Multiplier	136	0.0356	22.4952	9.9172	32.4481	3059
16-Bit Multiplier	583	0.1599	138.3030	54.4857	192.949	6344

CONCLUSION

- Dadda Multiplier is designed and implemented using Carry save adder on 45nm Technology library.
- Designed module is simulated on Cadence as well as on Xilinx softwares
- The modules are synthesized using Genus tool in cadence and the Power, delay and area of 8-bit and 16-bit multipliers are analysed by comparing the reports.

Thank You