Assignment 3
VLSI Design

Amit Kumar,16D070034

October 31, 2019

Q–1

Assume that the delay of some common gates (inclusive of parasitic delay) is as given below:

- **Inverter** 100 ps
- **NAND gate** 150 ps
- **NOR gate** 150 ps
- **A+B.C** 200 ps
- **Tiny XOR** 200 ps
- **Half Adder (carry)** 250 ps
- **Half Adder (sum)** 200 ps
- **Full Adder (carry)** 400 ps
- **Full Adder (sum)** 400 ps

16x16 bit Multiplier using Dadda Architecture

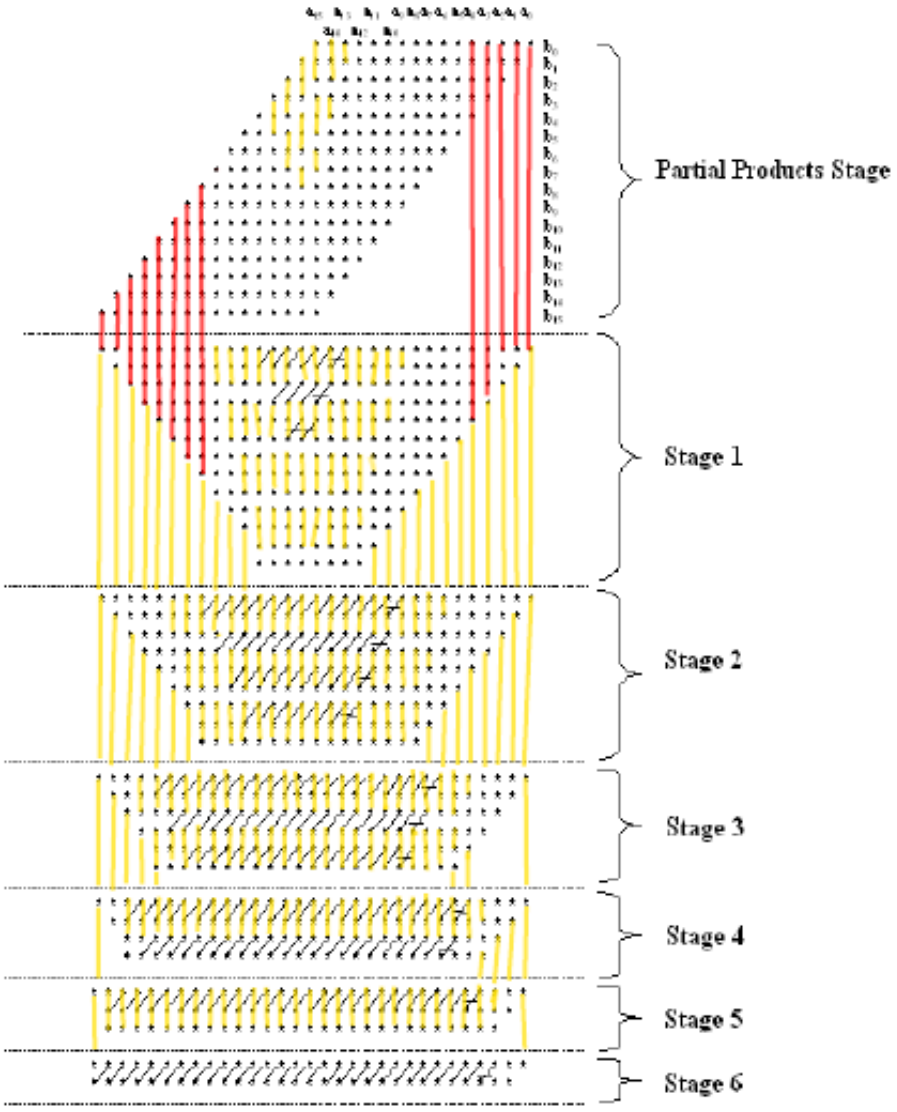

Figure 1: 16 x 16 bit Dadda Multiplier

Common Gates implementation in VHDL

```
1  ----------------------------------------------
2  library std;
3  use std.standard.all;
4  ----------------------------------------------------------
5
6  library ieee;
7  use ieee.std_logic_1164.all;
8  entity Inverter is
9    port (a: in std_logic;
10          b: out std_logic);
11 end entity Inverter;
12 architecture Behave of Inverter is
13 begin
14   b <= not a after 100ps;    -----Delay of Iinverter gate
15 end Behave;
16
17 ----------------------------------------------------------
18
19 library ieee;
20 use ieee.std_logic_1164.all;
21 entity NOR_2 is
22   port (a, b: in std_logic;
23          c: out std_logic);
24 end entity NOR_2;
25 architecture Behave of NOR_2 is
26 begin
27   c <= not(a or b) after 150ps; -----Delay of NOR gate
28 end Behave;
29
30 ----------------------------------------------------------
31
32 library ieee;
33 use ieee.std_logic_1164.all;
34 entity NAND_2 is
35   port (a, b: in std_logic;
36          c: out std_logic);
37 end entity NAND_2;
38 architecture Behave of NAND_2 is
39 begin
40   c <= not (a and b) after 150ps;    -----Delay of NAND gate
41 end Behave;
42
43 ----------------------------------------------------------
```

```
44
45  library ieee;
46  use ieee.std_logic_1164.all;
47  entity tinyXOR is
48    port (a, b: in std_logic;
49          c: out std_logic);
50  end entity tinyXOR;
51  architecture Behave of tinyXOR is
52  begin
53    c <= (a xor b) after 200ps;    -----Delay of Tiny_XOR gate
54  end Behave;
55
56  ------------------------------------------------------------
57
58  library ieee;
59  use ieee.std_logic_1164.all;
60  entity Custom is
61    port (a, b, c : in std_logic;
62          d: out std_logic);
63  end entity Custom;
64  architecture Behave of Custom is
65  begin
66    d <= not(a or (b and c)) after 200ps;
67  end Behave;
68
69  ------------------------------------------------------------
```

Half Adder and full adder in VHDL

Half Adder

$$Sum_{HA} = A_i \oplus B_i \tag{1}$$

$$Carry_{HA} = A_i \cdot B_i \tag{2}$$

Full Adder

$$Sum_{FA} = A_i \oplus B_i \oplus C_{in} \tag{3}$$

$$Carry_{FA} = A_i \cdot B_i + C_{in}(A + B) \tag{4}$$

```
--------------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
entity HA_sum is
  port (a, b: in std_logic;
        c: out std_logic);
end entity HA_sum;
architecture Behave of HA_sum is
begin
  c <= (a xor b) after 200ps;   -----Delay of HA_sum gate
end Behave;
--------------------------------------------------------------------------------


library ieee;
use ieee.std_logic_1164.all;
entity HA_carry is
  port (a, b: in std_logic;
        c: out std_logic);
end entity HA_carry;
architecture Behave of HA_carry is
begin
  c <=  (a and b)  after 250ps;   -----Delay of HA_carry gate
end Behave;

--------------------------------------------------------------------------------


library ieee;
use ieee.std_logic_1164.all;
entity FA_carry is
  port (a, b, c: in std_logic;
        d: out std_logic);
end entity FA_carry;
architecture Behave of FA_carry is
begin
  d <=  ( (a and b) or (c and (a or b)) )  after 400ps;   -----Delay of FA_carry gate
end Behave;

--------------------------------------------------------------------------------


library ieee;
```

```
use ieee.std_logic_1164.all;
entity FA_sum is
  port (a, b, c: in std_logic;
        d: out std_logic);
end entity FA_sum;
architecture Behave of FA_sum is
begin
  d <= ((not a ) and (not b) and c      )
    or ((not a ) and b       and (not c))
    or (a        and (not b) and (not c))
         or (a        and b       and c      ) after 400ps;   -----Delay of FA_sum gate
end Behave;
--------------------------------------------------------------------------------
```

Dadda Stages network

```
--------------------------------------------
library std;
use std.standard.all;
-----------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
entity Dadda is
  port (a,b: in std_logic_vector(15 downto 0);
        prod1,prod2: out std_logic_vector(31 downto 0)
                    );
end entity Dadda;

architecture Behave of Dadda is

signal s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,s15 : std_logic_vector(15 downto 0) ;
--signal s5_1_29, temp1, temp2 : std_logic ;
signal s1_1  : std_logic_vector(30 downto 0)  ;
signal s1_2  : std_logic_vector(29 downto 1)  ;
signal s1_3  : std_logic_vector(28 downto 2)  ;
signal s1_4  : std_logic_vector(27 downto 3)  ;
signal s1_5  : std_logic_vector(26 downto 4)  ;
signal s1_6  : std_logic_vector(25 downto 5)  ;
signal s1_7  : std_logic_vector(24 downto 6)  ;
signal s1_8  : std_logic_vector(23 downto 7)  ;
signal s1_9  : std_logic_vector(22 downto 8)  ;
signal s1_10 : std_logic_vector(21 downto 9)  ;
signal s1_11 : std_logic_vector(20 downto 10) ;
```

```
29  signal s1_12 : std_logic_vector(19 downto 11) ;
30  signal s1_13 : std_logic_vector(19 downto 12) ;
31
32
33  signal s2_1  : std_logic_vector(30 downto 0)  ;
34  signal s2_2  : std_logic_vector(29 downto 1)  ;
35  signal s2_3  : std_logic_vector(28 downto 2)  ;
36  signal s2_4  : std_logic_vector(27 downto 3)  ;
37  signal s2_5  : std_logic_vector(26 downto 4)  ;
38  signal s2_6  : std_logic_vector(25 downto 5)  ;
39  signal s2_7  : std_logic_vector(24 downto 6)  ;
40  signal s2_8  : std_logic_vector(23 downto 7)  ;
41  signal s2_9  : std_logic_vector(23 downto 8)  ;
42
43
44  signal s3_1  : std_logic_vector(30 downto 0)  ;
45  signal s3_2  : std_logic_vector(29 downto 1)  ;
46  signal s3_3  : std_logic_vector(28 downto 2)  ;
47  signal s3_4  : std_logic_vector(27 downto 3)  ;
48  signal s3_5  : std_logic_vector(26 downto 4)  ;
49  signal s3_6  : std_logic_vector(26 downto 5)  ;
50
51
52  signal s4_1  : std_logic_vector(30 downto 0)  ;
53  signal s4_2  : std_logic_vector(29 downto 1)  ;
54  signal s4_3  : std_logic_vector(28 downto 2)  ;
55  signal s4_4  : std_logic_vector(28 downto 3)  ;
56
57
58  signal s5_1  : std_logic_vector(30 downto 0)  ;
59  signal s5_2  : std_logic_vector(29 downto 1)  ;
60  signal s5_3  : std_logic_vector(29 downto 2)  ;
61
62  signal s6_1  : std_logic_vector(30 downto 0)  ;
63  signal s6_2  : std_logic_vector(30 downto 1)  ;
64
65
66  ------------------Components-----------------------------
67  component and16 is
68    port (a: in std_logic_vector(15 downto 0);
69          b: in std_logic;
70                    c: out std_logic_vector(15 downto 0)
71                    );
72  end component and16;
73
```

```
74  component FA_sum is
75    port (a, b, c: in std_logic;
76          d: out std_logic);
77  end component FA_sum;
78
79  component FA_carry is
80    port (a, b, c: in std_logic;
81         d: out std_logic);
82  end component FA_carry;
83
84
85  component HA_carry is
86    port (a, b: in std_logic;
87          c: out std_logic);
88  end component HA_carry;
89
90  component HA_sum is
91    port (a, b: in std_logic;
92          c: out std_logic);
93  end component HA_sum;
94
95  ----------------------------------------------------------------------
96  begin
97
98  --------Stage 0
99  ----Creating all partial products for stage 0
100
101 s_01: and16 port map ( a => a, b => b(0)  , c => s0)  ;
102 s_02: and16 port map ( a => a, b => b(1)  , c => s1)  ;
103 s_03: and16 port map ( a => a, b => b(2)  , c => s2)  ;
104 s_04: and16 port map ( a => a, b => b(3)  , c => s3)  ;
105 s_05: and16 port map ( a => a, b => b(4)  , c => s4)  ;
106 s_06: and16 port map ( a => a, b => b(5)  , c => s5)  ;
107 s_07: and16 port map ( a => a, b => b(6)  , c => s6)  ;
108 s_08: and16 port map ( a => a, b => b(7)  , c => s7)  ;
109 s_09: and16 port map ( a => a, b => b(8)  , c => s8)  ;
110 s_10: and16 port map ( a => a, b => b(9)  , c => s9)  ;
111 s_11: and16 port map ( a => a, b => b(10) , c => s10) ;
112 s_12: and16 port map ( a => a, b => b(11) , c => s11) ;
113 s_13: and16 port map ( a => a, b => b(12) , c => s12) ;
114 s_14: and16 port map ( a => a, b => b(13) , c => s13) ;
115 s_15: and16 port map ( a => a, b => b(14) , c => s14) ;
116 s_16: and16 port map ( a => a, b => b(15) , c => s15) ;
117
118 ----------------------------------------------------------------------------------------------
```

```
-----/////////////////-----Stage 1----------/////////////////----------------------------

-------Stage 1 Row 1 -------------------

   loop1: for i in 0 to 12 generate
 begin
    s1_1(i)  <= s0(i)  ;
 end generate loop1;

S1_1_13 : HA_sum port map (a => s0(13),b =>  s1(12),c =>  s1_1(13))              ;
S1_1_14 : FA_sum port map (a => s0(14),b =>  s1(13),c =>  s2(12), d => s1_1(14) ) ;
S1_1_15 : FA_sum port map (a => s0(15),b =>  s1(14),c =>  s2(13), d => s1_1(15) ) ;
S1_1_16 : FA_sum port map (a => s1(15),b =>  s2(14),c =>  s3(13), d => s1_1(16) ) ;
S1_1_17 : FA_sum port map (a => s2(15),b =>  s3(14),c =>  s4(13), d => s1_1(17) ) ;
S1_1_18 : FA_sum port map (a => s3(15),b =>  s4(14),c =>  s5(13), d => s1_1(18) ) ;

s1_1(19) <= s4(15)  ;
s1_1(20) <= s5(15)  ;
s1_1(21) <= s6(15)  ;
s1_1(22) <= s7(15)  ;
s1_1(23) <= s8(15)  ;
s1_1(24) <= s9(15)  ;
s1_1(25) <= s10(15) ;
s1_1(26) <= s11(15) ;
s1_1(27) <= s12(15) ;
s1_1(28) <= s13(15) ;
s1_1(29) <= s14(15) ;
s1_1(30) <= s15(15) ;

--------------Stage 1 Row 2---------------------------

loop2: for i in 1 to 12 generate
begin
  s1_2(i)  <= s1(i-1)  ;
end generate loop2;


s1_2(13) <=s2(11) ;


S1_2_14 : HA_carry port map (a => s0(13),b =>  s1(12),c =>  s1_2(14))              ;
S1_2_15 : FA_carry port map (a => s0(14),b =>  s1(13),c =>  s2(12), d => s1_2(15) ) ;
S1_2_16 : FA_carry port map (a => s0(15),b =>  s1(14),c =>  s2(13), d => s1_2(16) ) ;
S1_2_17 : FA_carry port map (a => s1(15),b =>  s2(14),c =>  s3(13), d => s1_2(17) ) ;
S1_2_18 : FA_carry port map (a => s2(15),b =>  s3(14),c =>  s4(13), d => s1_2(18) ) ;
```

```
S1_2_19 : FA_carry port map (a => s3(15),b =>  s4(14),c =>  s5(13), d => s1_2(19) ) ;


s1_2(20) <= s6(14)  ;
s1_2(21) <= s7(14)  ;
s1_2(22) <= s8(14)  ;
s1_2(23) <= s9(14)  ;
s1_2(24) <= s10(14) ;
s1_2(25) <= s11(14) ;
s1_2(26) <= s12(14) ;
s1_2(27) <= s13(14) ;
s1_2(28) <= s14(14) ;
s1_2(29) <= s15(14) ;

--------------------Stage 1 Row 3-------------------------

loop3: for i in 2 to 12 generate
begin
  s1_3(i)  <= s2(i-2)  ;
end generate loop3;


s1_3(13) <= s3(10) ;

S1_3_14 : HA_sum port map (a => s3(11),b =>  s4(10),c =>  s1_3(14) )             ;
S1_3_15 : FA_sum port map (a => s3(12),b =>  s4(11),c =>  s5(10), d => s1_3(15) ) ;
S1_3_16 : FA_sum port map (a => s4(12),b =>  s1(11),c =>  s6(10), d => s1_3(16) ) ;
S1_3_17 : FA_sum port map (a => s5(12),b =>  s6(11),c =>  s7(10), d => s1_3(17) ) ;

s1_3(18) <= s6(12)  ;
s1_3(19) <= s5(14)  ;

s1_3(20) <= s7(13)  ;
s1_3(21) <= s8(13)  ;
s1_3(22) <= s9(13)  ;
s1_3(23) <= s10(13) ;
s1_3(24) <= s11(13) ;
s1_3(25) <= s12(13) ;
s1_3(26) <= s13(13) ;
s1_3(27) <= s14(13) ;
s1_3(28) <= s15(13) ;

--------------Stage 1 Row 4 --------------------------

loop4: for i in 3 to 12 generate
```

```
209 begin
210   s1_4(i)  <= s3(i-3)  ;
211 end generate loop4;
212
213 s1_4(13) <= s4(9) ;
214 s1_4(14) <= s5(9) ;
215
216 S1_4_15 : HA_carry port map (a => s3(11),b =>  s4(10),c =>  s1_4(15) )             ;
217 S1_4_16 : FA_carry port map (a => s3(12),b =>  s4(11),c =>  s5(10), d => s1_4(16) ) ;
218 S1_4_17 : FA_carry port map (a => s4(12),b =>  s1(11),c =>  s6(10), d => s1_4(17) ) ;
219 S1_4_18 : FA_carry port map (a => s5(12),b =>  s6(11),c =>  s7(10), d => s1_4(18) ) ;
220
221 s1_4(19) <= s6(13)  ;
222
223 s1_4(20) <= s8(12)  ;
224 s1_4(21) <= s9(12)  ;
225 s1_4(22) <= s10(12) ;
226 s1_4(23) <= s11(12) ;
227 s1_4(24) <= s12(12) ;
228 s1_4(25) <= s13(12) ;
229 s1_4(26) <= s14(12) ;
230 s1_4(27) <= s15(12) ;
231
232 --------Stage 1 Row 5 -----------------------
233
234 loopz4: for i in 4 to 12 generate
235 begin
236   s1_5(i)  <= s4(i-4)  ;
237 end generate loopz4;
238 --
239 s1_5(13) <= s5(8) ;
240 s1_5(14) <= s6(8) ;
241
242 S1_5_15 : HA_sum port map (a => s6(9),b =>  s7(8),c =>  s1_5(15) )  ;
243 S1_5_16 : HA_sum port map (a => s7(9),b =>  s8(8),c =>  s1_5(16) )  ;
244
245 s1_5(17) <= s8(9)   ;
246 s1_5(18) <= s7(11)  ;
247 s1_5(19) <= s7(12)  ;
248
249 s1_5(20) <= s9(11)  ;
250 s1_5(21) <= s10(11) ;
251 s1_5(22) <= s11(11) ;
252 s1_5(23) <= s12(11) ;
253 s1_5(24) <= s13(11) ;
```

```
s1_5(25) <= s14(11) ;
s1_5(26) <= s15(11) ;

--------Stage 1 Row 6 ----------------------
loop5: for i in 5 to 12 generate
begin
  s1_6(i)  <= s5(i-5)  ;
end generate loop5;

s1_6(13) <= s6(7)  ;
s1_6(14) <= s7(7)  ;
s1_6(15) <= s8(7)  ;

S1_6_16 : HA_carry port map (a => s6(9),b =>  s7(8),c =>  s1_6(16) )  ;
S1_6_17 : HA_carry port map (a => s7(9),b =>  s8(8),c =>  s1_6(17) )  ;

s1_6(18) <= s8(10) ;
s1_6(19) <= s8(11) ;

s1_6(20) <= s10(10) ;
s1_6(21) <= s11(10) ;
s1_6(22) <= s12(10) ;
s1_6(23) <= s13(10) ;
s1_6(24) <= s14(10) ;
s1_6(25) <= s15(10) ;

-----------------------------------------------
----------Row 7---------------------------------
loop7x: for i in 6 to 12 generate
begin
  s1_7(i)  <= s6(i-6)  ;
end generate loop7x;

s1_7(13) <= s7(6)  ;
s1_7(14) <= s8(6)  ;
s1_7(15) <= s9(6)  ;

s1_7(16) <= s9(7)  ;
s1_7(17) <= s9(8)  ;
s1_7(18) <= s9(9)  ;
s1_7(19) <= s9(10) ;

s1_7(20) <= s11(9) ;
s1_7(21) <= s12(9) ;
s1_7(22) <= s13(9) ;
```

```
299  s1_7(23) <= s14(9) ;
300  s1_7(24) <= s15(9) ;
301
302  ----------Row 8----------------------------------
303
304  loop8x: for i in 7 to 12 generate
305  begin
306    s1_8(i)  <= s7(i-7)  ;
307  end generate loop8x;
308
309  s1_8(13) <= s8(5)  ;
310  s1_8(14) <= s9(5)  ;
311  s1_8(15) <= s10(5)  ;
312
313  s1_8(16) <= s10(6)  ;
314  s1_8(17) <= s10(7)  ;
315  s1_8(18) <= s10(8) ;
316  s1_8(19) <= s10(9) ;
317
318
319  s1_8(20) <= s12(8) ;
320  s1_8(21) <= s13(8) ;
321  s1_8(22) <= s14(8) ;
322  s1_8(23) <= s15(8) ;
323
324  ----------Row 9----------------------------------
325
326  s1_9(8)  <= s8(0)  ;
327  s1_9(9)  <= s8(1)  ;
328  s1_9(10)  <= s8(2)  ;
329  s1_9(11)  <= s8(3)  ;
330  s1_9(12) <= s8(4)  ;
331
332  s1_9(13) <=  s9(4)  ;
333  s1_9(14) <= s10(4)  ;
334  s1_9(15) <= s11(4)  ;
335
336  s1_9(16) <= s11(5)  ;
337  s1_9(17) <= s11(6)  ;
338  s1_9(18) <= s11(7) ;
339  s1_9(19) <= s11(8) ;
340
341  s1_9(20) <= s13(7) ;
342  s1_9(21) <= s14(7) ;
343  s1_9(22) <= s15(7) ;
```

```
344
345  ----------Row 10--------------------------------
346
347  s1_10(9)   <= s9(0)  ;
348  s1_10(10)  <= s9(1)  ;
349  s1_10(11)  <= s9(2)  ;
350  s1_10(12)  <= s9(3)  ;
351
352  s1_10(13) <= s10(3)  ;
353  s1_10(14) <= s11(3)  ;
354  s1_10(15) <= s12(3)  ;
355
356  s1_10(16) <= s12(4)  ;
357  s1_10(17) <= s12(5)  ;
358  s1_10(18) <= s12(6) ;
359  s1_10(19) <= s12(7) ;
360  s1_10(20) <= s14(6) ;
361  s1_10(21) <= s15(6) ;
362
363
364  ----------Row 11--------------------------------
365  s1_11(10)  <= s10(0)  ;
366  s1_11(11)  <= s10(1)  ;
367  s1_11(12)  <= s10(2)  ;
368
369  s1_11(13) <= s11(2)  ;
370  s1_11(14) <= s12(2)  ;
371  s1_11(15) <= s13(2)  ;
372
373  s1_11(16) <= s13(3)  ;
374  s1_11(17) <= s13(4)  ;
375  s1_11(18) <= s13(5) ;
376  s1_11(19) <= s13(6) ;
377
378  s1_11(20) <= s15(5) ;
379  ----------Row 12--------------------------------
380
381  s1_12(11)  <= s11(0)  ;
382  s1_12(12)  <= s11(1)  ;
383
384  s1_12(13) <= s12(1)  ;
385
386  s1_12(14) <= s13(1)  ;
387  s1_12(15) <= s14(1)  ;
388  s1_12(16) <= s14(2)  ;
```

```
s1_12(17) <= s14(3)  ;
s1_12(18) <= s14(4) ;
s1_12(19) <= s14(5) ;

----------Row 13--------------------------------

s1_13(12)  <= s12(0)  ;

s1_13(13) <= s13(0)  ;
s1_13(14) <= s14(0)  ;

s1_13(15) <= s15(0)  ;
s1_13(16) <= s15(1)  ;
s1_13(17) <= s15(2)  ;
s1_13(18) <= s15(3) ;
s1_13(19) <= s15(4) ;



-------------------------------------------------

--------\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
------------------------STAGE 2-----------------------------------------------------------------

loopx: for i in 0 to 8 generate
begin
  s2_1(i)  <= s1_1(i)  ;
end generate loopx;

S2_1_9 : HA_sum port map (a => s1_1(9),b =>  s1_2(9),c =>  s2_1(9) )  ;

loopa: for i in 10 to 22 generate
begin
S2_1_FA : FA_sum port map (a => s1_1(i),b =>  s1_2(i),c =>s1_3(i) , d =>  s2_1(i) )  ;
end generate loopa;


loopb: for i in 23 to 30 generate
begin
  s2_1(i)  <= s1_1(i)  ;
end generate loopb;


---------------Row 2 ----------------------------------------------------------------

```

```
434 loopr2: for i in 1 to 8 generate
435 begin
436   s2_2(i)  <= s1_2(i)  ;
437 end generate loopr2;
438
439 s2_2(9) <= s1_3(9) ;
440
441 S2_2_10 : HA_carry port map (a => s1_1(9),b =>  s1_2(9),c =>  s2_2(10) )  ;
442
443 loopr22: for i in 10 to 22 generate
444 begin
445 S2_2_Fcarry : FA_carry port map (a => s1_1(i),b =>  s1_2(i),c =>s1_3(i) , d =>  s2_2(i+1) )  ;
446 end generate loopr22;
447
448
449
450 loopr23: for i in 24 to 29 generate
451 begin
452   s2_2(i)  <= s1_2(i)  ;
453 end generate loopr23;
454
455 ---------------Row 3-----------------------------------------
456 loopr3: for i in 2 to 8 generate
457 begin
458   s2_3(i)  <= s1_3(i)  ;
459 end generate loopr3;
460
461 s2_3(9) <= s1_4(9) ;
462
463
464 S2_3_10 : HA_sum port map (a => s1_4(10),b =>  s1_5(10),c =>  s2_3(10) )  ;
465
466 loopr3a: for i in 11 to 21 generate
467 begin
468 S2_3_FA : FA_sum port map (a => s1_4(i),b =>  s1_5(i),c =>s1_6(i) , d =>  s2_3(i) )  ;
469 end generate loopr3a;
470
471 s2_3(22) <= s1_4(22) ;
472 s2_3(23) <= s1_2(23) ;
473
474 loopr3b: for i in 24 to 28 generate
475 begin
476   s2_3(i)  <= s1_3(i)  ;
477 end generate loopr3b;
478 ---------------------------------------------------------------
```

```
---------------Row 4-------------------------------------------
loopr4: for i in 3 to 8 generate
begin
  s2_4(i)  <= s1_4(i)  ;
end generate loopr4;

s2_4(9) <= s1_5(9) ;
s2_4(10) <= s1_6(10) ;


S2_4_11 : HA_carry port map (a => s1_4(10),b =>  s1_5(10),c =>  s2_4(11) )  ;

loopr4a: for i in 11 to 21 generate
begin
S2_4_Fcarry : FA_carry port map (a => s1_4(i),b =>  s1_5(i),c =>s1_6(i) , d =>  s2_4(i+1) )  ;
end generate loopr4a;

s2_4(23) <= s1_3(23) ;

loopr4b: for i in 24 to 27 generate
begin
  s2_4(i)  <= s1_4(i)  ;
end generate loopr4b;

-------------------------------------------------
-------------------------Row 5--------------------
loopr5: for i in 4 to 8 generate
begin
  s2_5(i)  <= s1_5(i)  ;
end generate loopr5;

s2_5(9) <= s1_6(9) ;
s2_5(10) <= s1_7(10) ;


S2_5_11 : HA_sum port map (a => s1_7(11),b =>  s1_8(11),c =>  s2_5(11) )  ;

loopr5a: for i in 12 to 20 generate
begin
S2_5_FA : FA_sum port map (a => s1_7(i),b =>  s1_8(i),c =>s1_9(i) , d =>  s2_5(i) )  ;
end generate loopr5a;

s2_5(21) <= s1_7(21) ;
s2_5(22) <= s1_5(22) ;
s2_5(23) <= s1_4(23) ;
```

```
524
525 loopr5b: for i in 24 to 26 generate
526 begin
527   s2_5(i)  <= s1_5(i)  ;
528 end generate loopr5b;
529
530 ---------------------------------------
531 ----------Row 6---------------------
532
533 loopr6: for i in 5 to 8 generate
534 begin
535   s2_6(i)  <= s1_6(i)  ;
536 end generate loopr6;
537
538 s2_6(9) <= s1_7(9) ;
539 s2_6(10) <= s1_8(10) ;
540 s2_6(11) <= s1_9(11) ;
541
542
543 S2_6_12 : HA_carry port map (a => s1_7(11),b =>  s1_8(11),c =>  s2_6(12) )  ;
544
545 loopr6a: for i in 12 to 20 generate
546 begin
547 S2_6_Fcarry : FA_carry port map (a => s1_7(i),b =>  s1_8(i),c =>s1_9(i) , d =>  s2_6(i+1) )  ;
548 end generate loopr6a;
549
550 s2_6(22) <= s1_6(22) ;
551 s2_6(23) <= s1_5(23) ;
552 s2_6(24) <= s1_6(24) ;
553 s2_6(25) <= s1_6(25) ;
554 ---------------------------------------------------------------
555 -------------Row 7-----------------------------------------------
556
557 s2_7(6)  <= s1_7(6)  ;
558 s2_7(7)  <= s1_7(7)  ;
559 s2_7(8)  <= s1_7(8)  ;
560
561 s2_7(9) <= s1_8(9)    ;
562 s2_7(10) <= s1_9(10)  ;
563 s2_7(11) <= s1_10(11) ;
564
565
566 S2_7_12 : HA_sum port map (a => s1_10(12),b =>  s1_11(12),c =>  s2_7(12) )  ;
567
568 loopr7a: for i in 13 to 19 generate
```

```
begin
S2_7_FA : FA_sum port map (a => s1_10(i),b =>  s1_11(i),c =>s1_12(i) , d =>  s2_7(i) )  ;
end generate loopr7a;

s2_7(20) <= s1_10(20) ;
s2_7(21) <= s1_8(21) ;

s2_7(22) <= s1_7(22) ;
s2_7(23) <= s1_6(23) ;

s2_7(24) <= s1_7(24) ;
-------------------------------------------------
--------------Row 8 ------------------------

s2_8(7)  <= s1_8(7)  ;
s2_8(8)  <= s1_8(8)  ;

s2_8(9) <= s1_9(9)    ;
s2_8(10) <= s1_10(10) ;
s2_8(11) <= s1_11(11) ;

s2_8(12) <= s1_12(12) ;

S2_8_13 : HA_carry port map (a => s1_10(12),b =>  s1_11(12),c =>  s2_8(13) )  ;

loopr8a: for i in 13 to 19 generate
begin
S2_8_FA : FA_carry port map (a => s1_10(i),b =>  s1_11(i),c =>s1_12(i) , d =>  s2_8(i+1) )  ;
end generate loopr8a;

s2_8(21) <= s1_9(21) ;
s2_8(22) <= s1_8(22) ;
s2_8(23) <= s1_7(23) ;
-----------------------------------------------
--------------------Row 9----------------------

s2_9(8)  <= s1_9(8)   ;
s2_9(9) <= s1_10(9)   ;
s2_9(10) <= s1_11(10) ;
s2_9(11) <= s1_12(11) ;

loopr9a: for i in 12 to 19 generate
begin
s2_9(i) <= s1_13(i) ;
end generate loopr9a;
```

```
s2_9(20) <= s1_11(20) ;
s2_9(21) <= s1_10(21) ;
s2_9(22) <= s1_9(22)  ;
s2_9(23) <= s1_8(23)  ;

-------------------------------------------------------------------------------------------------
----\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
-----STAGE 3 ----------------------------------------------

-------Row 1-----------------
loops3a: for i in 0 to 5 generate
begin
s3_1(i) <= s2_1(i) ;
end generate loops3a;

S3_1_6 : HA_sum port map (a => s2_1(6),b =>  s2_2(6),c =>  s3_1(6) )  ;

loops3b: for i in 7 to 25 generate
begin
S3_1_FA : FA_sum port map (a => s2_1(i),b =>  s2_2(i),c =>s2_3(i) , d =>  s3_1(i) )  ;
end generate loops3b;

loops3c: for i in 26 to 30 generate
begin
s3_1(i) <= s2_1(i) ;
end generate loops3c;

----------Row 2 --------------------------
loops32a: for i in 1 to 5 generate
begin
s3_2(i) <= s2_2(i) ;
end generate loops32a;

s3_2(6) <= s2_3(6) ;

S3_2_7 : HA_carry port map (a => s2_1(6),b =>  s2_2(6),c =>  s3_2(7) )  ;

loops32b: for i in 7 to 25 generate
begin
S3_2_Fcarry: FA_carry port map (a => s2_1(i),b =>  s2_2(i),c =>s2_3(i) , d =>  s3_2(i+1) )  ;
end generate loops32b;


s3_2(27) <= s2_2(27) ;
```

```
659 s3_2(28) <= s2_2(28) ;
660 s3_2(29) <= s2_2(29) ;
661
662 ----------Row 3-------------------------------------
663 loops33a: for i in 2 to 5 generate
664 begin
665 s3_3(i) <= s2_3(i) ;
666 end generate loops33a;
667
668 s3_3(6) <= s2_4(6) ;
669
670 S3_3_7 : HA_sum port map (a => s2_4(7),b =>  s2_5(7),c =>  s3_3(7) )  ;
671
672 loops33b: for i in 8 to 24 generate
673 begin
674 S3_3_FA: FA_sum port map (a => s2_4(i), b =>  s2_5(i), c =>s2_6(i), d =>  s3_3(i))  ;
675 end generate loops33b;
676
677 s3_3(25) <= s2_4(25) ;
678 s3_3(26) <= s2_2(26) ;
679 s3_3(27) <= s2_3(27) ;
680 s3_3(28) <= s2_3(28) ;
681 -------------------
682 ------------Row 4
683
684 loops34a: for i in 3 to 5 generate
685 begin
686 s3_4(i) <= s2_4(i) ;
687 end generate loops34a;
688
689 s3_4(6) <= s2_5(6) ;
690 s3_4(7) <= s2_6(7) ;
691
692 S3_4_8 : HA_carry port map (a => s2_4(7),b =>  s2_5(7),c =>  s3_4(8) )  ;
693
694 loops34b: for i in 8 to 24 generate
695 begin
696 S3_4_FC: FA_carry port map (a => s2_4(i), b =>  s2_5(i), c =>s2_6(i), d =>  s3_4(i+1))  ;
697 end generate loops34b;
698
699 s3_4(26) <= s2_3(26) ;
700 s3_4(27) <= s2_4(27) ;
701
702
703 ------------Row 5
```

```
704
705 loops35a: for i in 4 to 5 generate
706 begin
707 s3_5(i) <= s2_5(i) ;
708 end generate loops35a;
709
710 s3_5(6) <= s2_6(6) ;
711 s3_5(7) <= s2_7(7) ;
712
713 S3_5_8 : HA_sum port map (a => s2_7(8),b =>  s2_8(8),c =>  s3_5(8) )  ;
714
715 loops35b: for i in 9 to 23 generate
716 begin
717 S3_5_Fsum: FA_sum port map (a => s2_7(i), b =>  s2_8(i), c =>s2_9(i), d =>  s3_5(i))  ;
718 end generate loops35b;
719
720 s3_5(24) <= s2_7(24) ;
721 s3_5(25) <= s2_5(25) ;
722 s3_5(26) <= s2_4(26) ;
723 ------------------------------------------
724 ----------------Row 6--------------------
725
726 s3_6(5) <= s2_6(5) ;
727
728
729 s3_6(6) <= s2_7(6) ;
730 s3_6(7) <= s2_8(7) ;
731
732 s3_6(8) <= s2_9(8) ;
733
734 S3_6_9 : HA_carry port map (a => s2_7(8),b =>  s2_8(8),c =>  s3_6(9) )  ;
735
736 loops36b: for i in 9 to 23 generate
737 begin
738 S3_6_FC: FA_carry port map (a => s2_7(i), b =>  s2_8(i), c =>s2_9(i), d =>  s3_6(i+1))  ;
739 end generate loops36b;
740
741 s3_6(25) <= s2_6(25) ;
742 s3_6(26) <= s2_5(26) ;
743 ------------------------------------------------------------------------------------------
744 -------\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
745 ------------STAGE 4
746
747 -------Row 1-----------------
748 loops4a: for i in 0 to 3 generate
```

```
749 begin
750 s4_1(i) <= s3_1(i) ;
751 end generate loops4a;
752
753 S4_1_4 : HA_sum port map (a => s3_1(4),b =>  s3_2(4),c =>  s4_1(4) )  ;
754
755 loops4b: for i in 5 to 27 generate
756 begin
757 S4_1_FA : FA_sum port map (a => s3_1(i),b =>  s3_2(i),c =>s3_3(i) , d =>  s4_1(i) )  ;
758 end generate loops4b;
759
760 loops4c: for i in 28 to 30 generate
761 begin
762 s4_1(i) <= s3_1(i) ;
763 end generate loops4c;
764
765 -------Row 2----------------
766 loops42: for i in 1 to 3 generate
767 begin
768 s4_2(i) <= s3_2(i) ;
769 end generate loops42;
770
771 s4_2(4) <= s3_3(4) ;
772
773 S4_2_5 : HA_carry port map (a => s3_1(4),b =>  s3_2(4),c =>  s4_2(5) )  ;
774
775 loops42a: for i in 5 to 27 generate
776 begin
777 S4_2_FC : FA_carry port map (a => s3_1(i), b =>  s3_2(i), c =>s3_3(i) , d =>  s4_2(i+1) )  ;
778 end generate loops42a;
779
780
781 s4_2(29) <= s3_2(29) ;
782 ---------------------------------------
783 -----Row 3
784
785
786 s4_3(2) <= s3_3(2) ;
787 s4_3(3) <= s3_3(3) ;
788
789 s4_3(4) <= s3_4(4) ;
790
791 S4_3_5 : HA_sum port map (a => s3_4(5),b =>  s3_5(5),c =>  s4_3(5) )  ;
792
793 loops43: for i in 6 to 26 generate
```

```
794 begin
795 S4_3_FA : FA_sum port map (a => s3_4(i), b =>  s3_5(i), c =>s3_6(i) , d =>  s4_3(i) )  ;
796 end generate loops43;
797
798 s4_3(27) <= s3_4(27) ;
799 s4_3(28) <= s3_2(28) ;
800
801 ---------------------------------------
802 -----Row 4
803
804
805 s4_4(3) <= s3_4(3) ;
806
807 s4_4(4) <= s3_5(4) ;
808 s4_4(5) <= s3_6(5) ;
809
810 S4_4_6 : HA_carry port map (a => s3_4(5),b =>  s3_5(5),c =>  s4_4(6) )  ;
811
812 loops44: for i in 6 to 26 generate
813 begin
814 S4_4_FA : FA_carry port map (a => s3_4(i), b =>  s3_5(i), c =>s3_6(i) , d =>  s4_4(i+1) )  ;
815 end generate loops44;
816
817 s4_4(28) <= s3_3(28) ;
818
819
820 ---------------------------------------------------------
821 --\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
822 -----------STAGE 5
823
824 -------Row 1-----------------
825 loops5a: for i in 0 to 2 generate
826 begin
827 s5_1(i) <= s4_1(i) ;
828 end generate loops5a;
829
830 S5_1_3 : HA_sum port map (a => s4_1(3),b =>  s4_2(3),c =>  s5_1(3) )  ;
831
832 loops5b: for i in 4 to 28 generate
833 begin
834 S5_1_FA : FA_sum port map (a => s4_1(i),b =>  s4_2(i), c =>s4_3(i) , d =>  s5_1(i) )  ;
835 end generate loops5b;
836
837 s5_1(29) <= s4_1(29) ;
838 s5_1(30) <= s4_1(30) ;
```

```
839  -------Row 2-----------------
840  loops52a: for i in 1 to 2 generate
841  begin
842  s5_2(i) <= s4_2(i) ;
843  end generate loops52a;
844
845  s5_2(3) <= s4_3(3) ;
846
847  S5_2_4 : HA_carry port map (a => s4_1(3),b =>  s4_2(3),c =>  s5_2(4) )  ;
848
849  loops52b: for i in 4 to 28 generate
850  begin
851  S5_2_FA : FA_carry port map (a => s4_1(i),b =>  s4_2(i),c =>s4_3(i) , d =>  s5_2(i+1) )  ;
852  end generate loops52b;
853
854  -----------Row 3 --------
855  s5_3(2) <= s4_3(2) ;
856
857  loops53: for i in 3 to 28 generate
858  begin
859  s5_3(i) <= s4_4(i) ;
860  end generate loops53;
861
862  s5_3(29) <= s4_2(29) ;
863
864  ---------------------------------------------------
865  ----------\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\-------\
866  ---------------Stage 6
867  --------------------------Row 1
868
869  s6_1(0) <= s5_1(0) ;
870  s6_1(1) <= s5_1(1) ;
871
872  S6_1_2 : HA_sum port map (a => s5_1(2),b =>  s5_2(2),c =>  s6_1(2) )  ;
873
874  loops6b: for i in 3 to 29 generate
875  begin
876  S6_1_FA : FA_sum port map (a => s5_1(i),b =>  s5_2(i),c =>s5_3(i) , d =>  s6_1(i) )  ;
877  end generate loops6b;
878
879
880  s6_1(30) <= s5_1(30) ;
881
882  ------------------------------Row 2
883
```

```
884  s6_2(1) <= s5_2(1) ;
885  s6_2(2) <= s5_3(2) ;
886
887  S6_2_3 : HA_carry port map (a => s5_1(2),b =>  s5_2(2),c =>  s6_2(3) )  ;
888
889  loops62: for i in 3 to 29 generate
890  begin
891  S6_1_FC : FA_carry port map (a => s5_1(i),b =>  s5_2(i),c =>s5_3(i) , d =>  s6_2(i+1) )  ;
892  end generate loops62;
893
894  ---------------------------------------------------------------------------------
895
896
897  prod1(31) <= '0' ;
898  prod1(30 downto 0) <= s6_1 ;
899
900  prod2(31) <= '0' ;
901  prod2(30 downto 1) <= s6_2 ;
902  prod2(0) <= '0' ;
903
904  end Behave;
905
906  ---------------------------------------------------
```

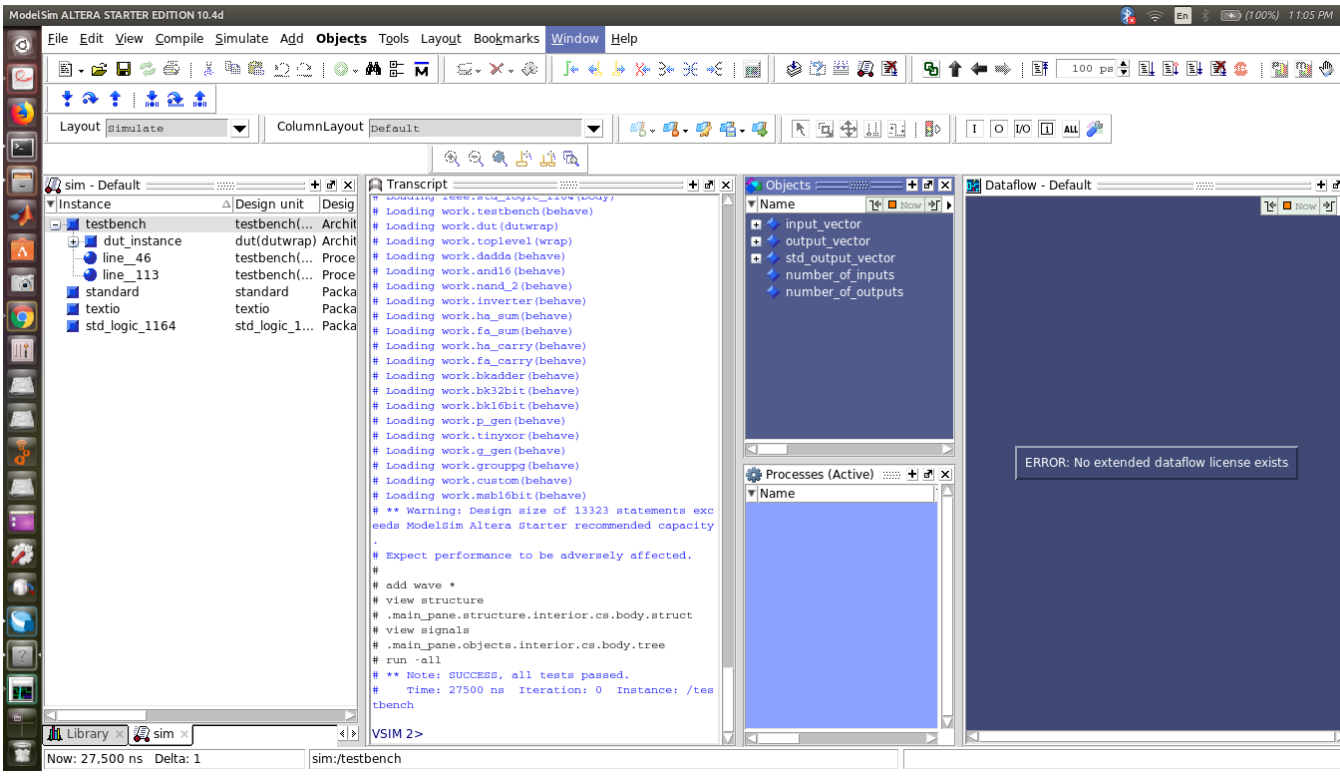

Figure 2: Succes of all test cases

Delay Analysis

Time taken in Brent-Kung Adder Black dots in my Brent-kung cell for P,G generation in carry generation stage which take **300ps** (Custom + Inverter).
So total time taken in **carry generation** is $11 \times 300 =$ **3300ps**
Time taken in pre-processing stage in **intial p,g generation** (nand + Inverter) is $150 + 100 =$ **250 ps**
Finally, time taken in **sum bits generation** (post-processing stage) is one xor which is **200 ps**.

Hence the time which we guarantee that the addition by Brent Kung will be complete = 3300 + 250 +200 = **3750 ps**

Dadda Multiplier
Time taken for partial product generation (Nand + inverter) = 150 + 100 = **250ps**
Time taken for Full adders in 6 stages of Dadda = 400 * 6 = **2400ps**
Total time taken for guaranteed multiplication =
3750 + 250 + 2400 = **6400 ps**
Critical Path will be determined by the critical path of the brent-kung adder which is 31st bit generation of th sum.

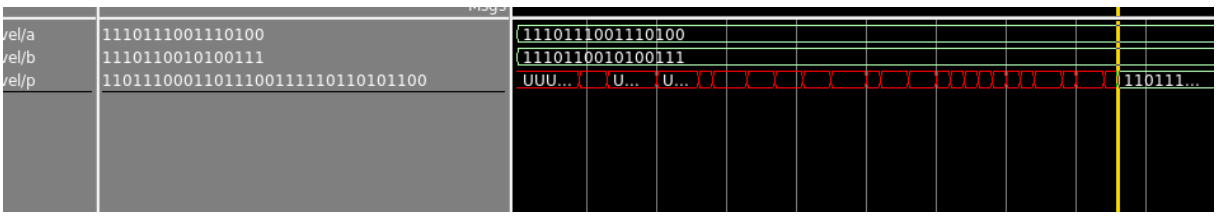

Figure 3: Delay of a test case