VLSI Design

Dadda Multiplier 8×8

EE 671 : Assignment #4

Simranjeet Singh - 183076005

November 12, 2018

Dadda Reduction

Dadda Multiplier 8×8 reduction is shown in the following Figure 1. Table represent the number of half adder and full adder required to implement this multiplier.

Table 1: Number of adder

Adder	Number
Full Adder	48
Half Adder	8

Given delay

As per the given in the Assignment problem statement.

Table 2: Delay parameter

Adder	Sum Delay	Carry Delay
Full Adder	120 ps	80 ps
Half Adder	70 ps	40 ps
MUX	_	50 ps

	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
row_00_st_0										O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)
row_01_st_0									(0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)
row_02_st_0								O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	
row_03_st_0							O (0)	(0)	(0)	O (0)	(0)	O (0)	O (0)	O (0)	O (0)		
row_04_st_0						O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)			
row_05_st_0					O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	(0)	(0)	O (0)				
row_06_st_0				O (0)	O (0)	O (0)	(0)	(0)	(0)	(0)	O (0)	O (0)					
row_07_st_0			O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	0 ,						
row_08_st_0		O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)	14						
row_00_st_1		O (0)	O (0)	O (0)	O (0)	(0)	(0)	(0)	(4)	(0)	0 (0	(0)	O (0)	O (0)	O (0)	(0)	O (0)
row_01_st_1			O (0)	O (0)	(0)	O (0)	O (0)	O (4)	O (7)	O (4)	O (0	O (0)	(0)	O (0)	O (0)	O (0)	O (0)
row_02_st_1				O (0)	O (0)	O (0)	O (0)	O (8)	O (8)	O (7)	0	0 (0)	0 (0)	O (0)	O (0)	O (0)	O (0)
row_03_st_1					O (0)	0 (0)	(8)	(8)	(8)	(8)	0 (4)	0 (0)	(0)	(0)	O (0)	O (0)	O (0)
row_04_st_1						O (0)	O (8)	O (12)	O (12)	O (12)	O (7)	O (0)	O (0)	O (0)	O (0)	O (0)	O (0)
row_05_st_1						O (8)	O (12)	O (12)	O (12)	O (12)	O (2)	O (7)	O (0)	(0)	O (0)	O (0)	O (0)
row_00_st_2		O (0)	O (0)	(0)	(0)	(8)	(12)	(16)	(15)	(8)	O (8)	(4)	(0)	(0)	(0)	(0)	(0)
row_01_st_2			O (0)	O (0)	O (8)	O (12)	O (16)	O (20)	O (2 0)	O (19)	O (12)	O (8)	O (4)	O (0)	O (0)	O (0)	O (0)
row_02_st_2				O (0)	O (12)	O (20)	O (20)	O (20)	O (2 0)	O (20)	O (15)	O (12)	O (7)	O (0)	(0)	O (0)	O (0)
row_03_st_2				(8)	(16)	(20)	(24)	(24)	(2 4)	O (24)	O (24)	(19)	(12)	(7)	O (0)	O (0)	O (0)
row_00_st_3		O (0)	(0)	(8)	O (16)	O (20)	O (24)	(24)	(2 4)	O (23)	O (20)	O (15)	(8)	(4)	(0)	O (0)	O (0)
row_01_st_3			O (0)	O (12)	O (24)	O (28)	O (28)	O (28)	O (2 8)	O (24)	O (24)	O (19)	O (12)	O (7)	O (0)	(0)	O (0)
row_02_st_3			O (8)	O (20)	O (28)	O (32)	O (32)	O (32)	O (32)	O (32)	O (27)	O (24)	O (19)	O (12)	O (7)	O (0)	O (0)
row_00_st_4		O (0)	(20)	(32)	(40)	O (40)	0 (40)	(40)	(4 0)	(35)	(32)	(27)	(20)	(15)	(4)	(0)	O (0)
row_01_st_4		(16)	(28)	(36)	(40)	(44)	(44)	(44)	O (44)	O (44)	O (39)	O (36)	O (31)	O (24)	O (19)	O (7)	O (0)

Simulation Result

Result Shown in the below figure is tested by applying the input $(FF \times FF) + 0F0F = 10D10$. The complete result is available at 810 ps

Observation

I have tested the code on different test vector to check the delay model. The worst delay i was finding on the following case:

 $(FF \times 0F) + 00FF = 000F0 - 910 \text{ ps}$

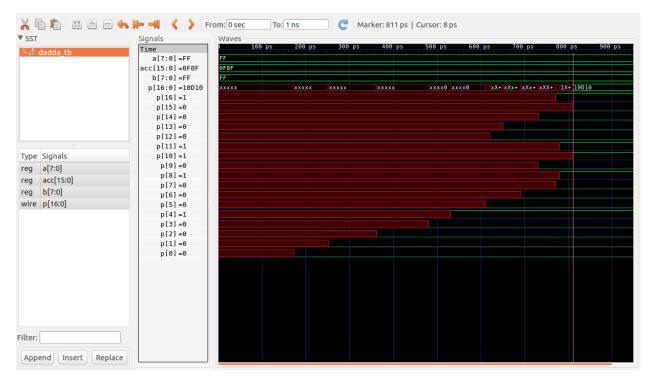


Figure 2: Dadda Multiplier Simulation

November 12, 2018 Page 3