**ECE 501 : Contemporary Digital Systems and VHDL**

**Assignment –3: 4-Bit Adders and 16-Bit Adders**

Roshni Uppala

1011735230

1. Objective :

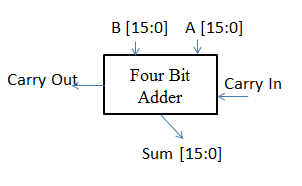
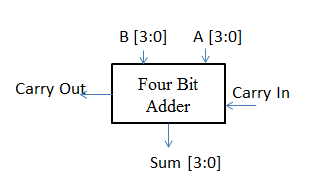
To build one bit adder, four but ripple carry adder and a 16 bit adder circuit using Verilog HDL ( structural code ) .

1. Design Specification :

* Inputs :
* One bit adder : Two one bit vectors ( A , B) and a one bit carry- in( Cin).
* 4-bit ripple carry adder : Two 4-bit vectors ( A[3:0] and B[3:0]) and a one bit carry- in( Cin) which is used to concatenate bit slices.
* 16 bit ripple carry adder : Two 16-bit vectors (A[15:0] and B[15:0]) and a one bit carry- in( Cin) which is used to concatenate bit slices.
* Outputs :
* One bit adder : A 1- bit sum , S and a 1-bit carry out( used to concatenate bit slices and to indicate overflow) .
* 4-bit ripple carry adder : A 4-bit sum, S[3:0] and a 1-bit carry out( used to concatenate bit slices and to indicate overflow) .
* 16-bit ripple carry adder : A 16-bit sum , S[15:0] and a 1-bit carry out ( used to concatenate bit slices and to indicate overflow) .
* Functional behavior :

Forms S=A+B+Carry-in.

Generates carry-out as required.

* Timing : 

Operates asynchronously.

Generates stable sum and carry-out.

* Other considerations :

None.

1. Design Structure :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | | | Output | |
| A | B | Cin | SUM | COUT |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Table 3.1 : 1-Bit full adder

* Register Transfer level of the design
* 1bit adder :
* Sum = a’b’cin + a’bcin’ + ab’cin’ + abcin

Carry-out = a’bcin + ab’cin + abcin’ + abcin

Inputs – a,b,cin

Outputs – sum, cout

* 4 bit ripple carry adder :
* FA0—full adder for bit 0

Inputs– A[0], B[0], carry-in

Outputs– S[0], c0

* FA1– full adder for bit 1

Inputs– A[1], B[1], carry-in

Outputs– S[1], c1

* FA2– full adder for bit 2

Inputs– A[2], B[2], carry-in

Outputs– S[2], c2

* FA3– full adder for bit 3

Inputs– A[3], B[3], carry-in

Outputs– S[3], carry-out

* 16 bit ripple carry adder :
* SFA0

Inputs : A[3:0], B[3:0],cin

Outputs : Sum[3:0] , c0

* SFA1

Inputs: A[7:4] , B[7:4] , c0

Outputs: Sum [7:4],c1

* SFA2

Inputs : A[11:8], B[11:8] ,c1

Outputs: Sum [11:8] , c2

* SFA3

Inputs : A[15:12], B[15:12] , c2

Outputs: Sum[15:12],c3

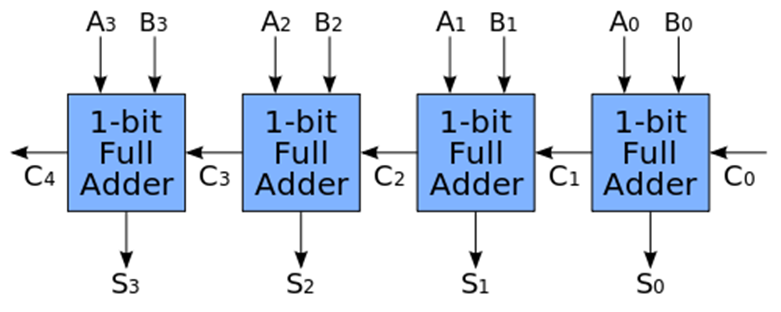
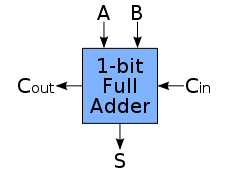


Fig 3.1: Left – 1 bit full adder ; right -- 4-bit ripple carry adder

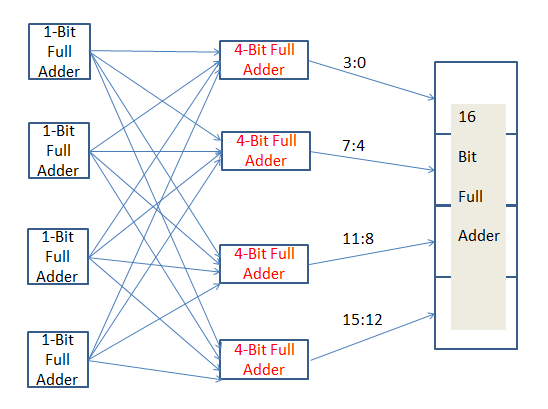


Fig 3.2 : 16-Bit ripple carry adder

1. Design Entry :

* 1 BIT ADDER :

//1-bit full adder

// saved as fulladderonebit.v

module fulladderonebit(a,b,cin, sum, cout);

input a, b, cin;

output sum, cout;

assign sum = !a && !b && cin || !a && b && !cin || a && !b &&!cin || a && b && cin;

assign cout = !a &&b && cin || a && !b && cin || a && b&& !cin || a && b && cin;

endmodule

* 4- BIT RIPPLE CARRY ADDER:

//four-bit –ripple carry adder

//uses fulladderonebit.v

//saved as fourbitadder.v

module fourbitadder(A, B, cin, SUM, cout);

input [3:0] A,B;

input cin;

output [3:0] SUM;

output cout;

wire c0,c1,c2;

fulladderonebit FA3(A[3], B[3], c2, SUM[3], cout);

fulladderonebit FA2(A[2], B[2], c1, SUM[2], c2);

fulladder onebitFA1(A[1], B[1], c0, SUM[1], c1);

fulladder onebitFA0(A[0], B[0], cin, SUM[0], c0);

endmodule

* 16- BIT RIPPLE CARRY ADDER:

//16 bit ripple carry adder

//uses fourbitadder.v

//uses onebitadder.v

//saved as sixteenbitadder.v

module sixteenbitadder(A, B, cin, SUM, cout);

input [15:0] A,B;

input cin;

output [15:0] SUM;

output cout;

wire c0,c1,c2;

fourbitadder FA3(A[15:12], B[15:12], c2, SUM[15:12], cout);

fourbitadder FA2(A[11:8], B[11:8], c1, SUM[11:8], c2);

fourbitadder FA1(A[7:4], B[7:4], c0, SUM[7:4], c1);

fourbitadder FA0(A[3:0], B[3:0], cin, SUM[3:0], c0);

endmodule

The one bit adder is used in the four bit ripple carry adder and these two are further extended to form a 16-bit ripple carry adder.

1. Reports :
2. Compilation report :

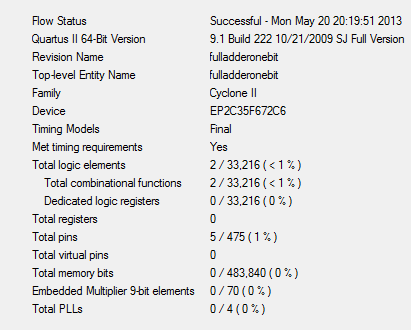


Fig 3.3 : 1-bit Full adder compilation result

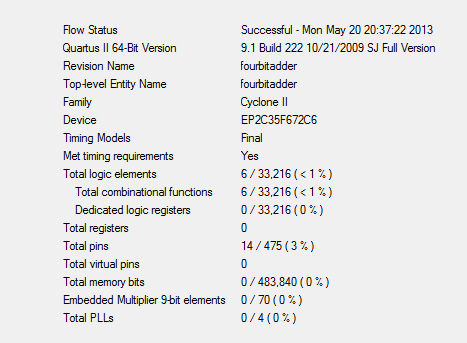


Fig 3.4 : 4-bit ripple carry adder compilation result

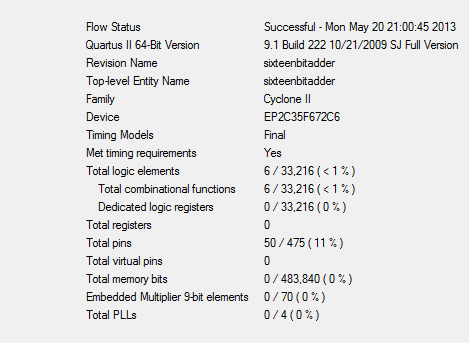


Fig 3.5 : 16-Bit ripple carry adder

1. RTL report :

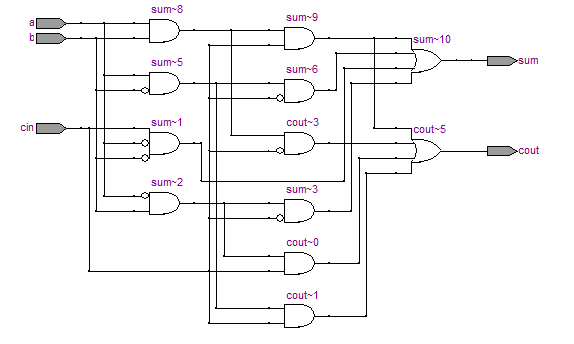


Fig 3.6 : 1-bit full adder

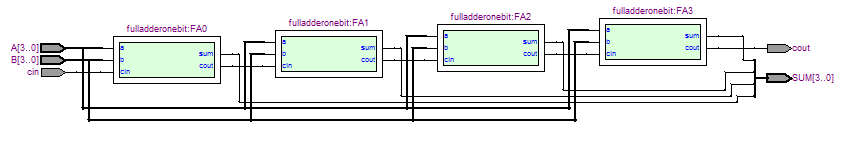
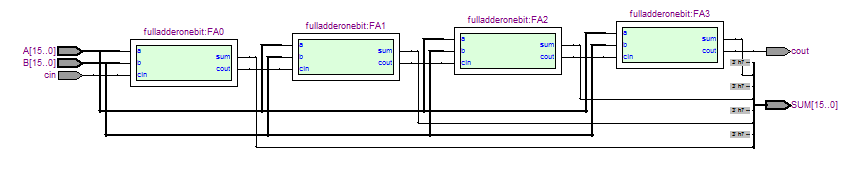
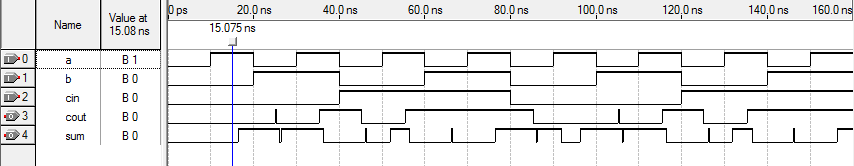


Fig 3.7 : 4-bit ripple carry full adder

Fig 3.8 : 16 bit ripple carry full adder

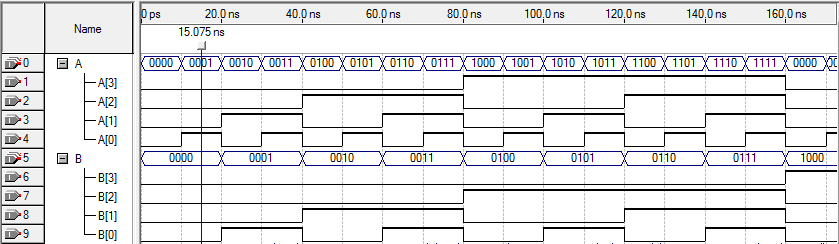
1. Simulation report :



A=1, B=0, CIN=0 therefore Sum-=1

These are glitches

Fig 3.9 : 1- bit full adder simulation



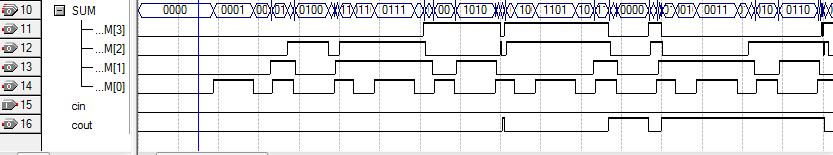


Fig 3.10: 4-bit ripple carry adder simulation result

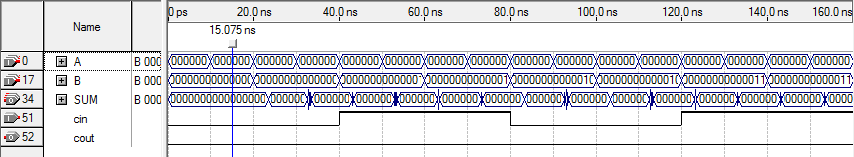


Fig 3.11: 16-bit ripple carry adder simulation result

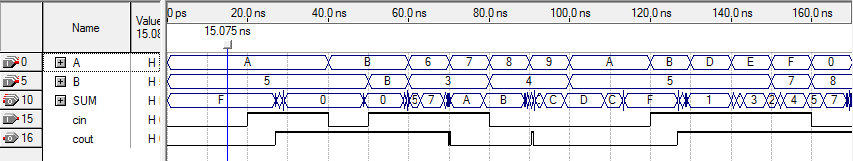
1. Design Verification / Test Plan :

* Test Plan
* Considering a given truth table ( as of table 3.2 and table 3.3 )
* Simulating the above Verilog codes for 4-bit and 16-bit ripple carry adder .
* Input values are assigned as of the truth table in table 3.2 and table 3.3
* Comparing the output values obtained from the simulation and the truth table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Input | | | Output | |
|  | A(Hex) | B(Hex) | Carry-in | Sum(Hex) | Carry-out |
| 1 | A | 5 | 0 | F | 0 |
| 2 | A | 5 | 1 | 0 | 1 |
| 3 | B | 5 | 0 | 0 | 1 |

Table 3.2 : 4-bit AND 16-bit ripple carry adder testing truth table

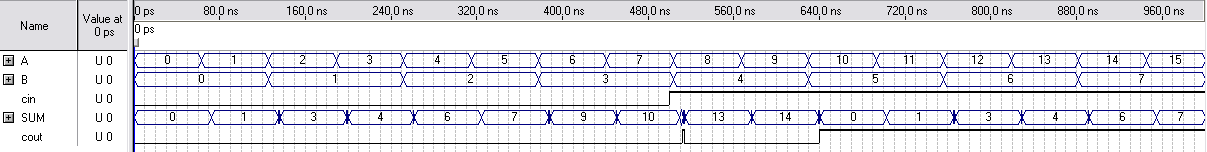
* Test Simulation report :



These are because the transition of A and B are so quick that it forces into the adder

Its seen that A +5 = F. The truth table in test plan satisfies

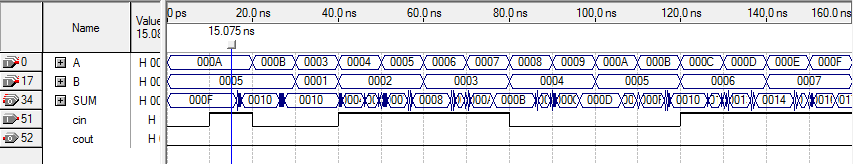
Fig 3.10 : 4-bit ripple carry adder test simulation result



These are glitches

Here it can be seen that 3+4 = 7

Fig 3.13: 16- bit ripple carry adder test simulation result 1



The truth table satisfies .

Fig 3.12: 16-bit ripple carry adder test simulation result 2