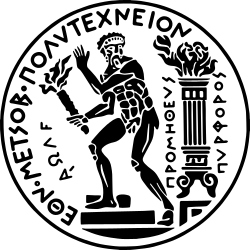
**Εθνικό Μετσόβιο Πολυτεχνείο** 

**Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών**

**Υπολογιστών**

Εξάμηνο: 8ο Ακ. Έτος: 2024- 2025

# «Ψηφιακά Συστήματα VLSI»

Χαράλαμπος Παπαδόπουλος (ΑΜ: 03120199)

Νικόλαος Παπακωνσταντόπουλος (ΑΜ: 03120069)

Ομάδα: 45

# BCD Parallel Adder

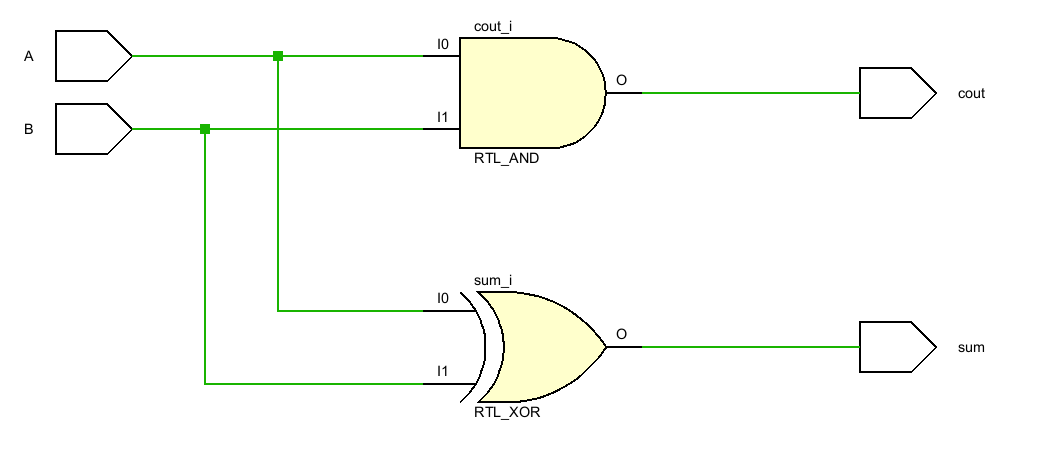
Ζητούμενο αυτής της άσκησης ήταν η υλοποίηση ενός BCD Parallel Adder 4 ψηφίων, ο οποίος παίρνει 2 Binary Coded Decimal αριθμούς των 4 ψηφίων (δηλαδή από 0 έως 9999) και τους αθροίζει κάνοντας διόρθωση στην έξοδο μέσω του .

Η άσκηση ακολούθησε μία bottom up λογική, δηλαδή σε κάθε ερώτημα μας ζητούταν να σχεδιάσουμε ένα κύκλωμα το οποίο θα χρησιμοποιούταν ως δομική μονάδα σε ένα πιο περίπλοκο κύκλωμα μέσω structural αρχιτεκτονικής.

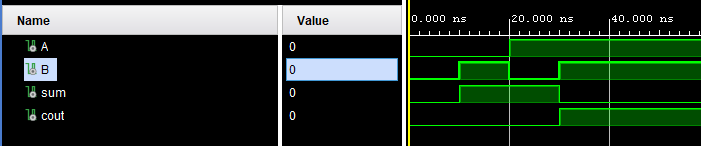
Παρατίθενται οι κώδικες κάθε υποερωτήματος, μαζί με τα αντίστοιχα RTL, testbench και critical paths.

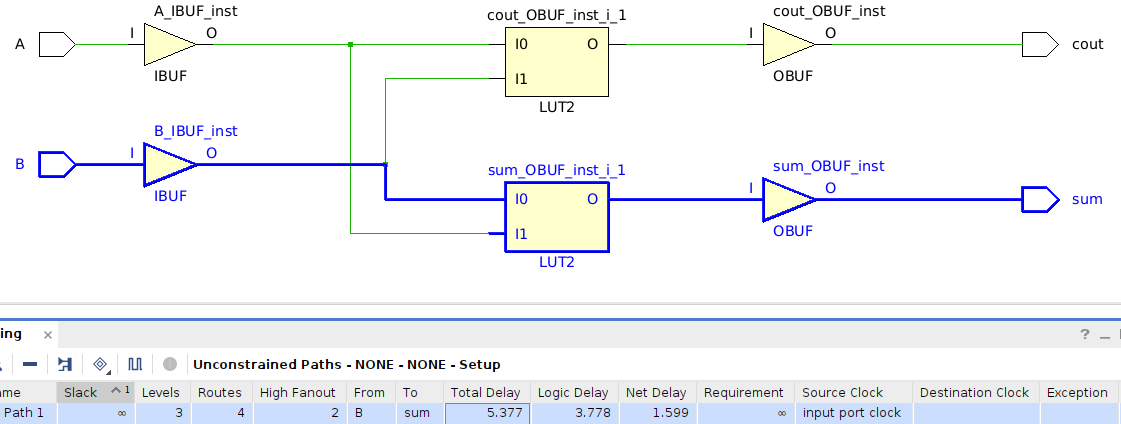
1. *Half Adder*

|  |
| --- |
| **library** **IEEE**;  **use** **IEEE.STD\_LOGIC\_1164.ALL**;  **entity** **half\_adder** **is**  **Port** (  A, B : **in** **std\_logic**;  sum, cout : **out** **std\_logic**  );  **end** **half\_adder**;  **architecture** **dataflow** **of** **half\_adder** **is**  **begin**  sum <= A **XOR** B;  cout <= A **AND** B;  **end** **dataflow**; |



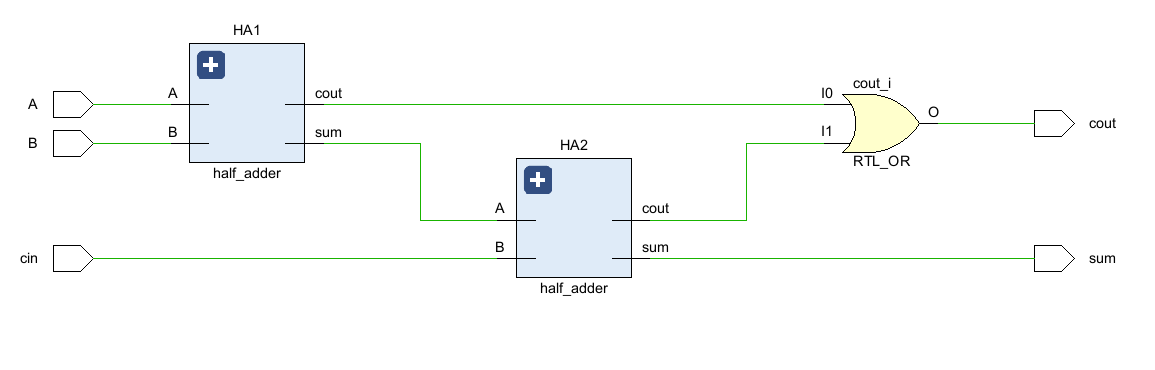
|  |
| --- |
| **library** **IEEE**;  **use** **IEEE.STD\_LOGIC\_1164.ALL**;  **entity** **half\_adder\_tb** **is**  **end** **half\_adder\_tb**;  **architecture** **test** **of** **half\_adder\_tb** **is**  **component** **half\_adder**  **Port** (  A, B : **in** **std\_logic**;  sum, cout : **out** **std\_logic**  );  **end** **component**;    **signal** A, B : **std\_logic**;  **signal** sum, cout : **std\_logic**;    **begin**  uut: half\_adder  **port** **map** (  A => A,  B => B,  sum => sum,  cout => cout  );    **process**  **begin**  A <= '0'; B <= '0'; **wait** **for** **10** ns;  A <= '0'; B <= '1'; **wait** **for** **10** ns;  A <= '1'; B <= '0'; **wait** **for** **10** ns;  A <= '1'; B <= '1'; **wait** **for** **10** ns;    **wait**;  **end** **process**;    **end** **test**; |



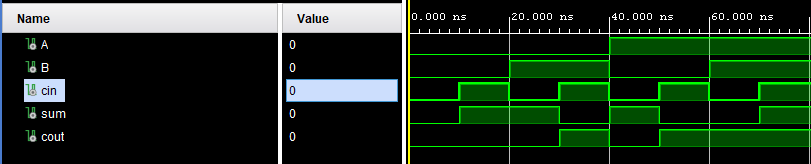


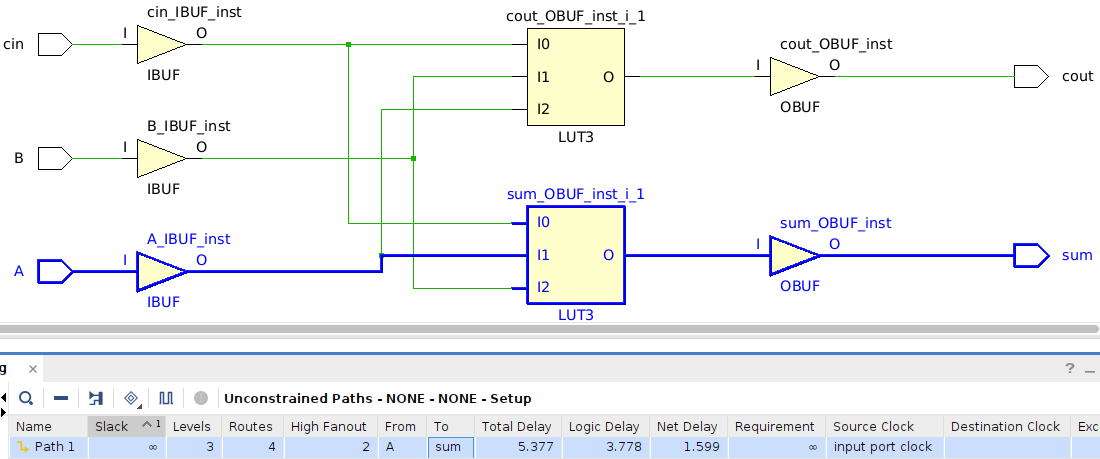
1. Full adder

|  |
| --- |
| **library** **IEEE**;  **use** **IEEE.STD\_LOGIC\_1164.ALL**;  **entity** **full\_adder** **is**  **Port** (  A, B, cin : **in** **std\_logic**;  sum, cout : **out** **std\_logic**  );  **end** **full\_adder**;  **architecture** **structural** **of** **full\_adder** **is**  **component** **half\_adder**  **Port** (  A, B : **in** **std\_logic**;  sum, cout : **out** **std\_logic**  );  **end** **component**;  **signal** sum1, c1, c2 : **std\_logic**;  **begin**  HA1: half\_adder  **port** **map** (  A => A,  B => B,  sum => sum1,  cout => c1  );    HA2: half\_adder  **port** **map** (  A => sum1,  B => cin,  sum => sum,  cout => c2  );    cout <= c1 **OR** c2;  **end** **structural**; |



|  |
| --- |
| **library** **IEEE**;  **use** **IEEE.STD\_LOGIC\_1164.ALL**;  **entity** **full\_adder\_tb** **is**  **end** **full\_adder\_tb**;  **architecture** **testbench** **of** **full\_adder\_tb** **is**    **component** **full\_adder**  **Port** (  A, B, cin : **in** **std\_logic**;  sum, cout : **out** **std\_logic**  );  **end** **component**;  **signal** A, B, cin : **std\_logic** := '0';  **signal** sum, cout : **std\_logic**;  **begin**    uut: full\_adder  **port** **map** (  A => A,  B => B,  cin => cin,  sum => sum,  cout => cout  );    **process**  **begin**  A <= '0'; B <= '0'; Cin <= '0'; **wait** **for** **10** ns;  A <= '0'; B <= '0'; Cin <= '1'; **wait** **for** **10** ns;  A <= '0'; B <= '1'; Cin <= '0'; **wait** **for** **10** ns;  A <= '0'; B <= '1'; Cin <= '1'; **wait** **for** **10** ns;  A <= '1'; B <= '0'; Cin <= '0'; **wait** **for** **10** ns;  A <= '1'; B <= '0'; Cin <= '1'; **wait** **for** **10** ns;  A <= '1'; B <= '1'; Cin <= '0'; **wait** **for** **10** ns;  A <= '1'; B <= '1'; Cin <= '1'; **wait** **for** **10** ns;  **wait**;  **end** **process**;  **end** **testbench**; |

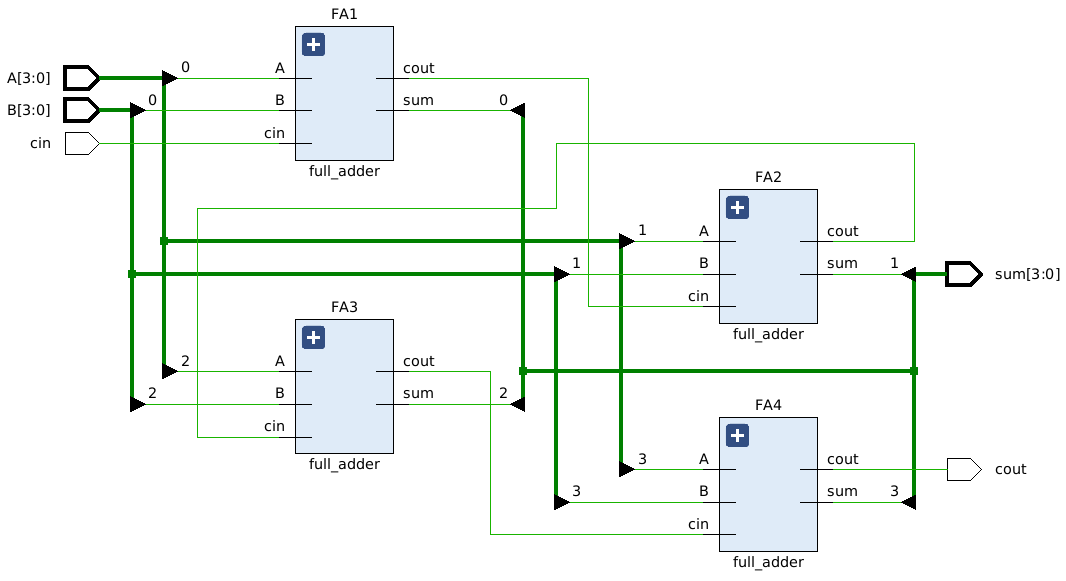




(ίσο με πριν, από είσοδο ως sum)

1. Parallel Adder

|  |
| --- |
| **library** **IEEE**;  **use** **IEEE.STD\_LOGIC\_1164.ALL**;  **entity** **parallel\_adder** **is**  **Port** (  A, B : **in** **std\_logic\_vector**(**4**-**1** **downto** **0**);  cin : **in** **std\_logic**;  sum : **out** **std\_logic\_vector**(**4**-**1** **downto** **0**);  cout : **out** **std\_logic**  );  **end** **parallel\_adder**;  **architecture** **structural** **of** **parallel\_adder** **is**  **component** **full\_adder** **is**  **Port** (  A, B, cin : **in** **std\_logic**;  sum, cout : **out** **std\_logic**  );  **end** **component**;  **signal** c1, c2, c3 : **std\_logic**;  **begin**  FA1: full\_adder  **port** **map**(  A => A(**0**),  B => B(**0**),  cin => cin,  cout => c1,  sum => sum(**0**)  );    FA2: full\_adder  **port** **map**(  A => A(**1**),  B => B(**1**),  cin => c1,  cout => c2,  sum => sum(**1**)  );  FA3: full\_adder  **port** **map**(  A => A(**2**),  B => B(**2**),  cin => c2,  cout => c3,  sum => sum(**2**)  );    FA4: full\_adder  **port** **map**(  A => A(**3**),  B => B(**3**),  cin => c3,  cout => cout,  sum => sum(**3**)  );  **end** **structural**; |



**library** **IEEE**;

**use** **IEEE.STD\_LOGIC\_1164.ALL**;

**entity** **parallel\_adder\_tb** **is**

**end** **parallel\_adder\_tb**;

**architecture** **testbench** **of** **parallel\_adder\_tb** **is**

**component** **parallel\_adder**

**Port** (

A, B : **in** **std\_logic\_vector**(**3** **downto** **0**);

cin : **in** **std\_logic**;

sum : **out** **std\_logic\_vector**(**3** **downto** **0**);

cout : **out** **std\_logic**

);

**end** **component**;

**signal** A, B : **std\_logic\_vector**(**3** **downto** **0**);

**signal** cin : **std\_logic**;

**signal** sum : **std\_logic\_vector**(**3** **downto** **0**);

**signal** cout : **std\_logic**;

**begin**

uut: parallel\_adder

**port** **map** (

A => A,

B => B,

cin => cin,

sum => sum,

cout => cout

);

**process**

**begin**

A <= "0000"; B <= "0000"; cin <= '0'; **wait** **for** **10** ns; -- 0 + 0

A <= "0001"; B <= "0001"; cin <= '0'; **wait** **for** **10** ns; -- 1 + 1

A <= "0010"; B <= "0011"; cin <= '0'; **wait** **for** **10** ns; -- 2 + 3

A <= "0110"; B <= "0010"; cin <= '1'; **wait** **for** **10** ns; -- 6 + 2 + 1

A <= "1010"; B <= "0101"; cin <= '0'; **wait** **for** **10** ns; -- 10 + 5

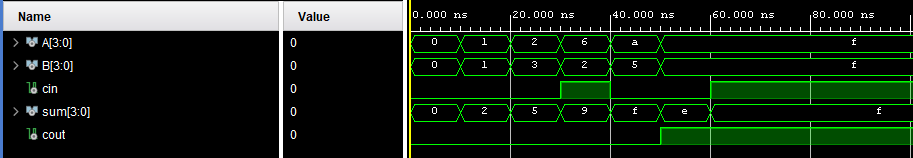
A <= "1111"; B <= "1111"; cin <= '0'; **wait** **for** **10** ns; -- 15 + 15

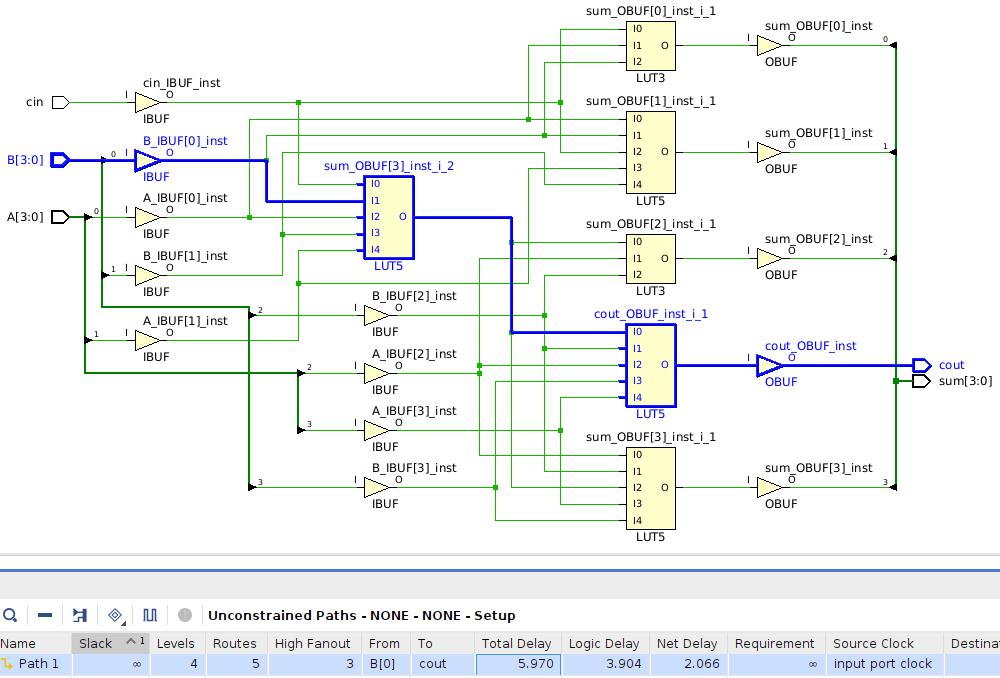
A <= "1111"; B <= "1111"; cin <= '1'; **wait** **for** **10** ns; -- 15 + 15 + 1

**wait**;

**end** **process**;

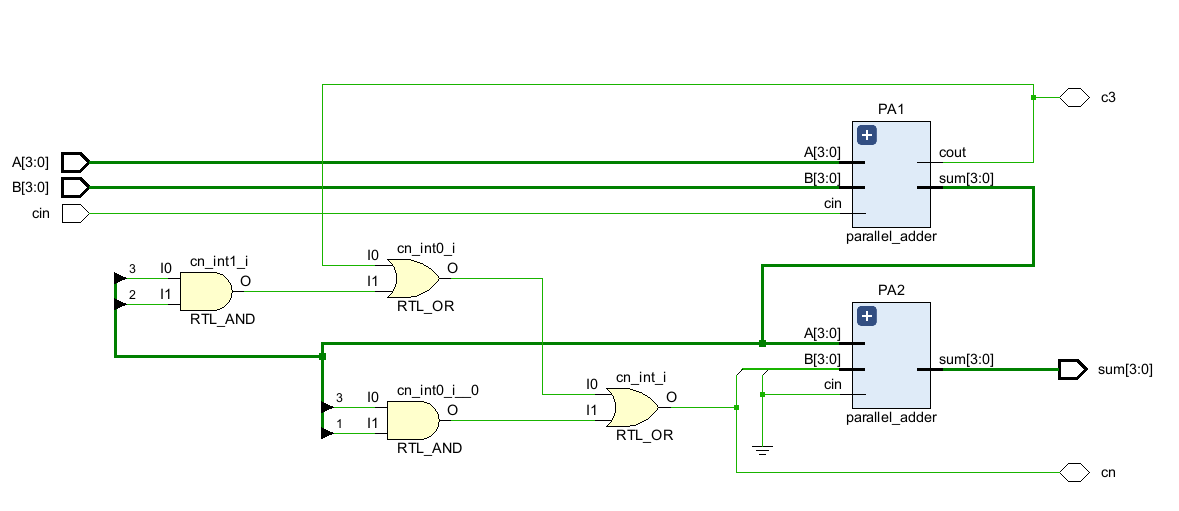
**end** **testbench**;



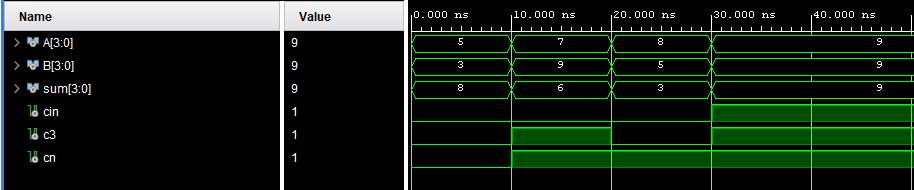


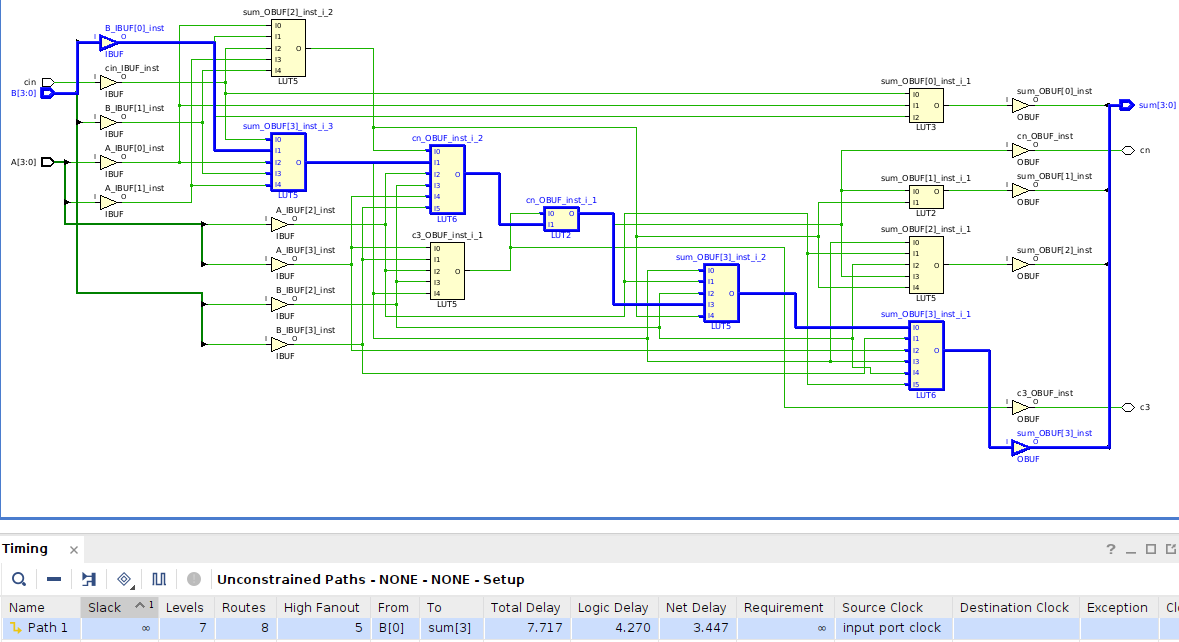
1. BCD Full Adder

|  |
| --- |
| **library** **IEEE**;  **use** **IEEE.STD\_LOGIC\_1164.ALL**;  **entity** **BCD\_full\_adder** **is**  **Port** (  A, B : **in** **std\_logic\_vector**(**4**-**1** **downto** **0**);  cin : **in** **std\_logic**;  sum : **out** **std\_logic\_vector**(**4**-**1** **downto** **0**);  c3, cn : **inout** **std\_logic**  );  **end** **BCD\_full\_adder**;  **architecture** **structural** **of** **BCD\_full\_adder** **is**  **signal** sum1, B1 : **std\_logic\_vector**(**3** **downto** **0**);  **signal** cn\_int : **std\_logic**;  **signal** or\_input : **std\_logic\_vector**(**2** **downto** **0**);  **component** **parallel\_adder** **is**  **Port** (  A, B : **in** **std\_logic\_vector**(**4**-**1** **downto** **0**);  cin : **in** **std\_logic**;  sum : **out** **std\_logic\_vector**(**4**-**1** **downto** **0**);  cout : **out** **std\_logic**  );  **end** **component**;  **begin**  PA1: parallel\_adder  **port** **map**(  A => A,  B => B,  sum => sum1,  cin => cin,  cout => c3  );  B1 <= ('0' & cn\_int & cn\_int & '0'); -- second input of second PA  PA2: parallel\_adder  **port** **map**(  A => sum1,  B => B1,  sum => sum,  cin => '0'  );    cn\_int <= ( c3 **OR** (sum1(**3**) **AND** sum1(**2**))**OR** (sum1(**3**) **and** sum1(**1**)) );  cn <= cn\_int;  **end** **structural**; |



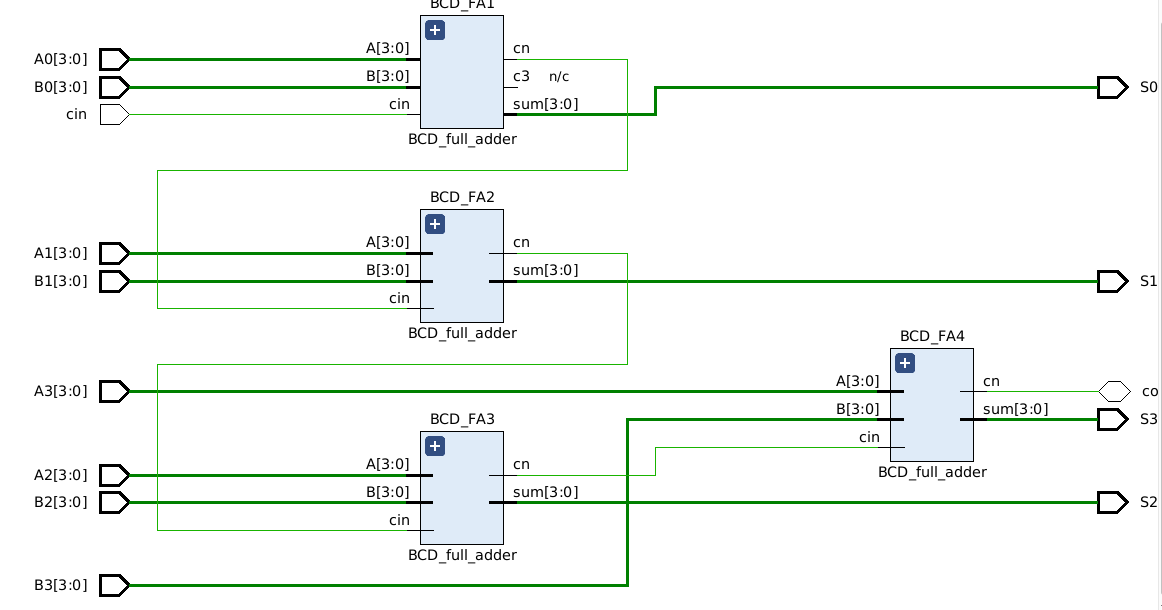
|  |
| --- |
| **library** **IEEE**;  **use** **IEEE.STD\_LOGIC\_1164.ALL**;  **entity** **BCD\_full\_adder\_tb** **is**  **end** **BCD\_full\_adder\_tb**;  **architecture** **test** **of** **BCD\_full\_adder\_tb** **is**  **component** **BCD\_full\_adder**  **Port** (  A, B : **in** **std\_logic\_vector**(**3** **downto** **0**);  cin : **in** **std\_logic**;  sum : **out** **std\_logic\_vector**(**3** **downto** **0**);  c3, cn: **inout** **std\_logic**  );  **end** **component**;    **signal** A, B, sum : **std\_logic\_vector**(**3** **downto** **0**);  **signal** cin, c3, cn : **std\_logic**;    **begin**  uut: BCD\_full\_adder  **port** **map** (  A => A,  B => B,  cin => cin,  sum => sum,  c3 => c3,  cn => cn  );  **process**  **begin**  A <= "0101"; -- 5  B <= "0011"; -- 3  cin <= '0';  **wait** **for** **10** ns;    A <= "0111"; -- 7  B <= "1001"; -- 9  cin <= '0';  **wait** **for** **10** ns;    A <= "1000"; -- 8  B <= "0101"; -- 5  cin <= '0';  **wait** **for** **10** ns;  A <= "1001"; -- 9  B <= "1001"; -- 9  cin <= '1';  **wait** **for** **10** ns;  **wait**;  **end** **process**;  **end** **test**; |





1. BCD Parallel Adder

|  |
| --- |
| **library** **IEEE**;  **use** **IEEE.STD\_LOGIC\_1164.ALL**;  **entity** **BCD\_parallel\_adder** **is**  **Port** (  A0, B0 : **in** **std\_logic\_vector**(**4**-**1** **downto** **0**);  A1, B1 : **in** **std\_logic\_vector**(**4**-**1** **downto** **0**);  A2, B2 : **in** **std\_logic\_vector**(**4**-**1** **downto** **0**);  A3, B3 : **in** **std\_logic\_vector**(**4**-**1** **downto** **0**);  cin : **in** **std\_logic**;  S0 : **out** **std\_logic\_vector**(**4**-**1** **downto** **0**);  S1 : **out** **std\_logic\_vector**(**4**-**1** **downto** **0**);  S2 : **out** **std\_logic\_vector**(**4**-**1** **downto** **0**);  S3 : **out** **std\_logic\_vector**(**4**-**1** **downto** **0**);  cout : **inout** **std\_logic**  );  **end** **BCD\_parallel\_adder**;  **architecture** **Behavioral** **of** **BCD\_parallel\_adder** **is**  **signal** cout\_0, cout\_1, cout\_2 : **std\_logic**;  **component** **BCD\_full\_adder** **is**  **Port** (  A, B : **in** **std\_logic\_vector**(**4**-**1** **downto** **0**);  cin : **in** **std\_logic**;  sum : **out** **std\_logic\_vector**(**4**-**1** **downto** **0**);  c3, cn : **inout** **std\_logic**  );  **end** **component**;  **begin**  BCD\_FA1: BCD\_full\_adder  **port** **map**(  A => A0,  B => B0,  sum => S0,  cin => cin,  cn => cout\_0  );    BCD\_FA2: BCD\_full\_adder  **port** **map**(  A => A1,  B => B1,  sum => S1,  cin => cout\_0,  cn => cout\_1  );  BCD\_FA3: BCD\_full\_adder  **port** **map**(  A => A2,  B => B2,  sum => S2,  cin => cout\_1,  cn => cout\_2  );    BCD\_FA4: BCD\_full\_adder  **port** **map**(  A => A3,  B => B3,  sum => S3,  cin => cout\_2,  cn => cout  );    **end** **Behavioral**; |



|  |
| --- |
| **library** **IEEE**;  **use** **IEEE.STD\_LOGIC\_1164.ALL**;  **use** **IEEE.NUMERIC\_STD.ALL**;  **use** **IEEE.STD\_LOGIC\_UNSIGNED**;  **entity** **BCD\_parallel\_adder\_tb** **is**  **end** **BCD\_parallel\_adder\_tb**;  **architecture** **test** **of** **BCD\_parallel\_adder\_tb** **is**  **component** **BCD\_parallel\_adder** **is**  **Port** (  A0, B0 : **in** **std\_logic\_vector**(**4**-**1** **downto** **0**);  A1, B1 : **in** **std\_logic\_vector**(**4**-**1** **downto** **0**);  A2, B2 : **in** **std\_logic\_vector**(**4**-**1** **downto** **0**);  A3, B3 : **in** **std\_logic\_vector**(**4**-**1** **downto** **0**);  cin : **in** **std\_logic**;  S0 : **out** **std\_logic\_vector**(**4**-**1** **downto** **0**);  S1 : **out** **std\_logic\_vector**(**4**-**1** **downto** **0**);  S2 : **out** **std\_logic\_vector**(**4**-**1** **downto** **0**);  S3 : **out** **std\_logic\_vector**(**4**-**1** **downto** **0**);  cout : **inout** **std\_logic**  );  **end** **component**;  **signal** A0, B0, A1, B1, A2, B2, A3, B3, S0, S1, S2, S3 : **std\_logic\_vector**(**4**-**1** **downto** **0**);  **signal** cin, cout : **std\_logic**;  **begin**  uut: BCD\_parallel\_adder  **port** **map** (  A0 => A0,  B0 => B0,  A1 => A1,  B1 => B1,  A2 => A2,  B2 => B2,  A3 => A3,  B3 => B3,  cin => cin,  S0 => S0,  S1 => S1,  S2 => S2,  S3 => S3,  cout => cout  );    **process**  **type** test\_array **is** **array**(**0** **to** **2**) **of** **integer** **range** **0** **to** **15**;  **constant** tests : test\_array := (  **0**, **5**, **9**  );  **begin**  **for** i **in** **0** **to** **2** **loop**  **for** x **in** **0** **to** **2** **loop**  **for** y **in** **0** **to** **2** **loop**    B3 <= **std\_logic\_vector**(to\_unsigned(tests(i), **4**));  B2 <= **std\_logic\_vector**(to\_unsigned(tests(i), **4**));  B1 <= **std\_logic\_vector**(to\_unsigned(tests(i), **4**));  B0 <= **std\_logic\_vector**(to\_unsigned(tests(i), **4**));  A3 <= **std\_logic\_vector**(to\_unsigned(tests(x), **4**));  A2 <= **std\_logic\_vector**(to\_unsigned(tests(x), **4**));  A1 <= **std\_logic\_vector**(to\_unsigned(tests(y), **4**));  A0 <= **std\_logic\_vector**(to\_unsigned(tests(y), **4**));    cin <= '0';  **wait** **for** **10** ns;  cin <= '1';  **wait** **for** **10** ns;  **end** **loop**;  **end** **loop**;  **end** **loop**;  **wait**;  **end** **process**;  **end** **test**; |

