VHDL Code

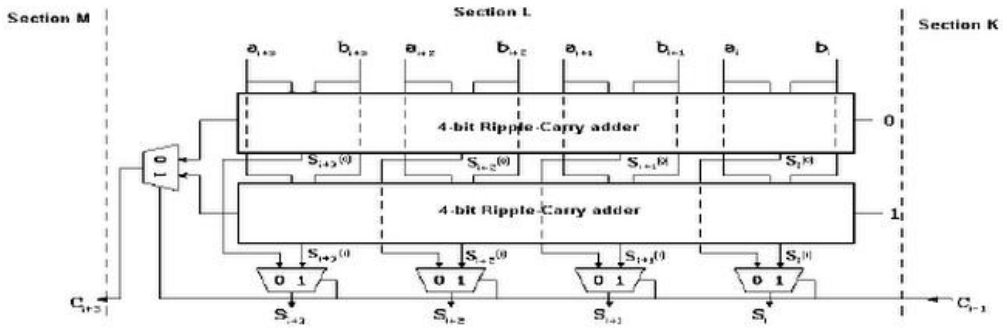
For

Carry Save Adder

Done By

Atchyuth Sonti

Carry Select Adder Example 4-bit Adder



- Is composed of two four-bit ripple carry adders per section
- Both sum and carry bits are calculated for the two alternatives of the input carry, "0" and "1"

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity main is
port( a:in std_logic_vector(3 downto 0);
     b:in std_logic_vector(3 downto 0);
                carry:in std_logic;
                outsum:out std_logic_vector(3 downto 0);
                outcarry:out std_logic);
end main;

architecture Behavioral of main is
signal c0,c1,sum0,sum1:std_logic_vector(3 downto 0);
signal carry0,carry1:std_logic;
component fulladder is
port(a,b,cin :in std_logic;
        sum,cout: out std_logic);
end component;
begin
 carry0<=carry;
 carry1<=not carry;
 fa01:fulladder port map(a(0),b(0),carry0,sum0(0),c0(0));
```

```
fa02:fulladder port map(a(1),b(1),c0(0),sum0(1),c0(1));
fa03:fulladder port map(a(2),b(2),c0(1),sum0(2),c0(2));
fa04:fulladder port map(a(3),b(3),c0(2),sum0(3),c0(3));
fa11:fulladder port map(a(0),b(0),carry1,sum1(0),c1(0));
fa12:fulladder port map(a(1),b(1),c1(0),sum1(1),c1(1));
fa13:fulladder port map(a(2),b(2),c1(1),sum1(2),c1(2));
fa14:fulladder port map(a(3),b(3),c1(2),sum1(3),c1(3));
process(carry)
begin
if(carry='0') then
outcarry<=c0(3);
outsum<=sum0;
else
outcarry<=c1(3);
outsum<=sum1;
end if;
end process;
end Behavioral;
```

Sub Program For Full Adder

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity fulladder is
    Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           cin : in  STD_LOGIC;
           sum : out  STD_LOGIC;
           cout : out  STD_LOGIC);
end fulladder;

architecture Behavioral of fulladder is
```

begin

sum<= a xor b xor cin;

cout<=(a and b)or(b and cin)or(cin and a);

end Behavioral;