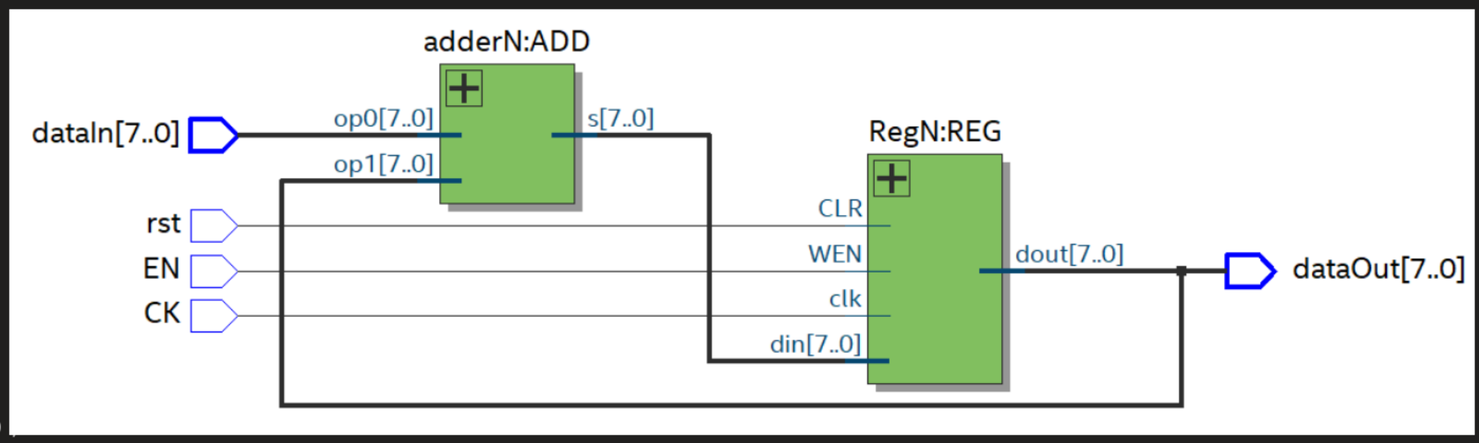
Accumulator

Tuesday, April 27, 2021 5:05 PM




```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity AccumN is
	generic(k : natural := 8);
	port (CK : in std_logic;
		EN : in std_logic;
			rst : in std_logic;
			dataIn : in std_logic_vector(k-1 downto 0);
			dataOut : out std_logic_vector(k-1 downto 0));
end AccumN;
architecture Structural of AccumN is
signal s_adderOut : std_logic_vector(k-1 downto 0);
signal s_regOut : std_logic_vector(k-1 downto 0);
begin
-- k bit register
REG: entity work.RegN(Behavioral)
		generic map(n => k)
		port map (din => s_adderOut,
			dout => s_regOut,
			WEN  => EN,
			CLR => rst,
			clk => CK);

-- k bit adder with no carry interface

ADD: entity work.AdderN(Arithm)
		generic map(n => k)
		 port map(op0 => dataIn,
			op1 => s_regOut,
			 s => s_adderOut);
	dataOut <= s_regOut;

end Structural;
```

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity RegN is
	generic(n : natural := 8);
	port (clk : in std_logic;
		WEN : in std_logic;
			CLR : in std_logic;
			din : in std_logic_vector(n-1 downto 0);
			dout : out std_logic_vector(n-1 downto 0));
end RegN;
architecture Behavioral of RegN is
begin
	process(clk)
	begin
		if rising_edge(clk) then
			if (CLR = '1') then
				dout <= (others => '0');
			elsif (WEN = '1') then
				dout <= din;
			end if;
		end if;
	end process;
end Behavioral;
```

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;
-- Generic Adder with no carry interface.
entity adderN is
	generic(n : natural := 8);
	port (op0:	in std_logic_vector(n-1 downto 0);
		op1:	in std_logic_vector(n-1 downto 0);
			s:		out std_logic_vector(n-1
downto 0));
end adderN;
architecture Arithm of AdderN is
begin
	s <= std_logic_vector(unsigned(op0)+unsigned(op1));

end Arithm;
```