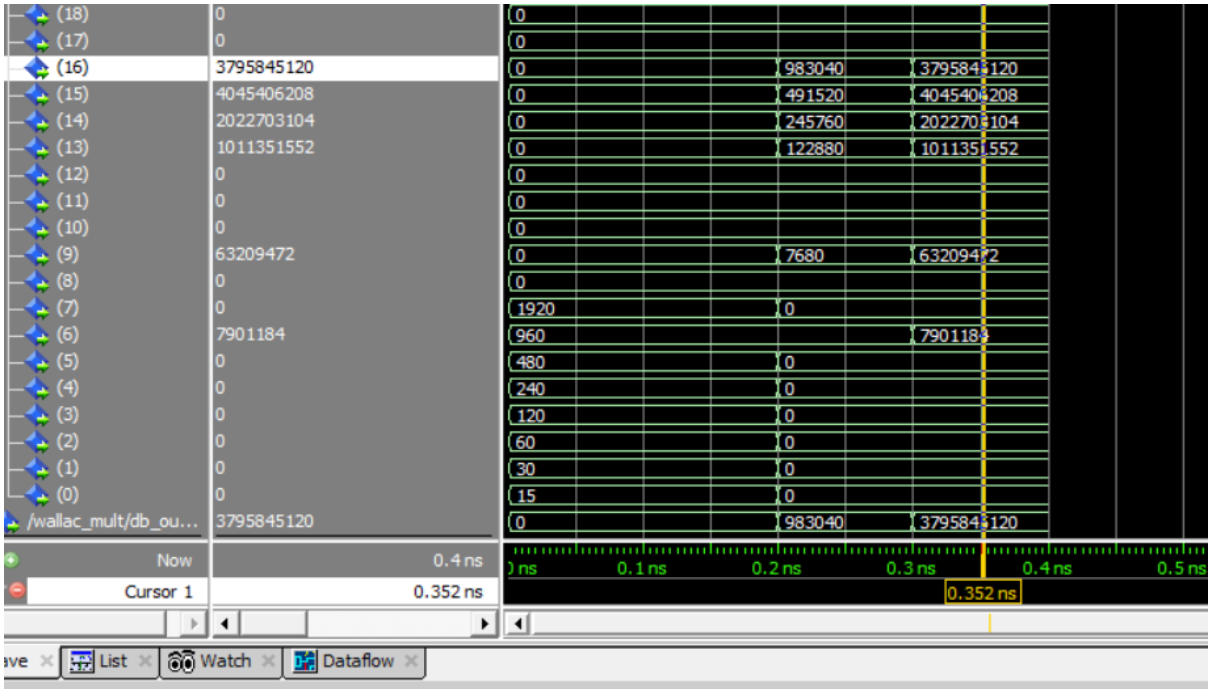
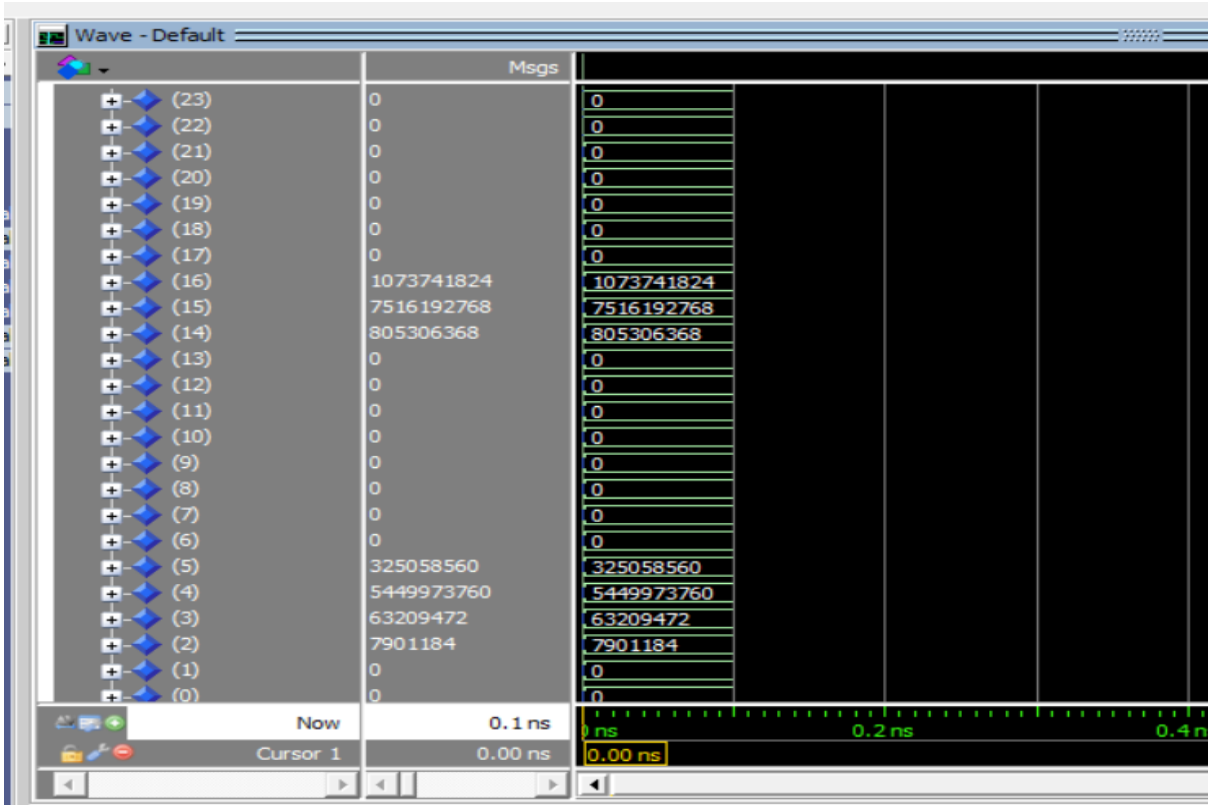
Results Of Wallace Multiplier Simulations

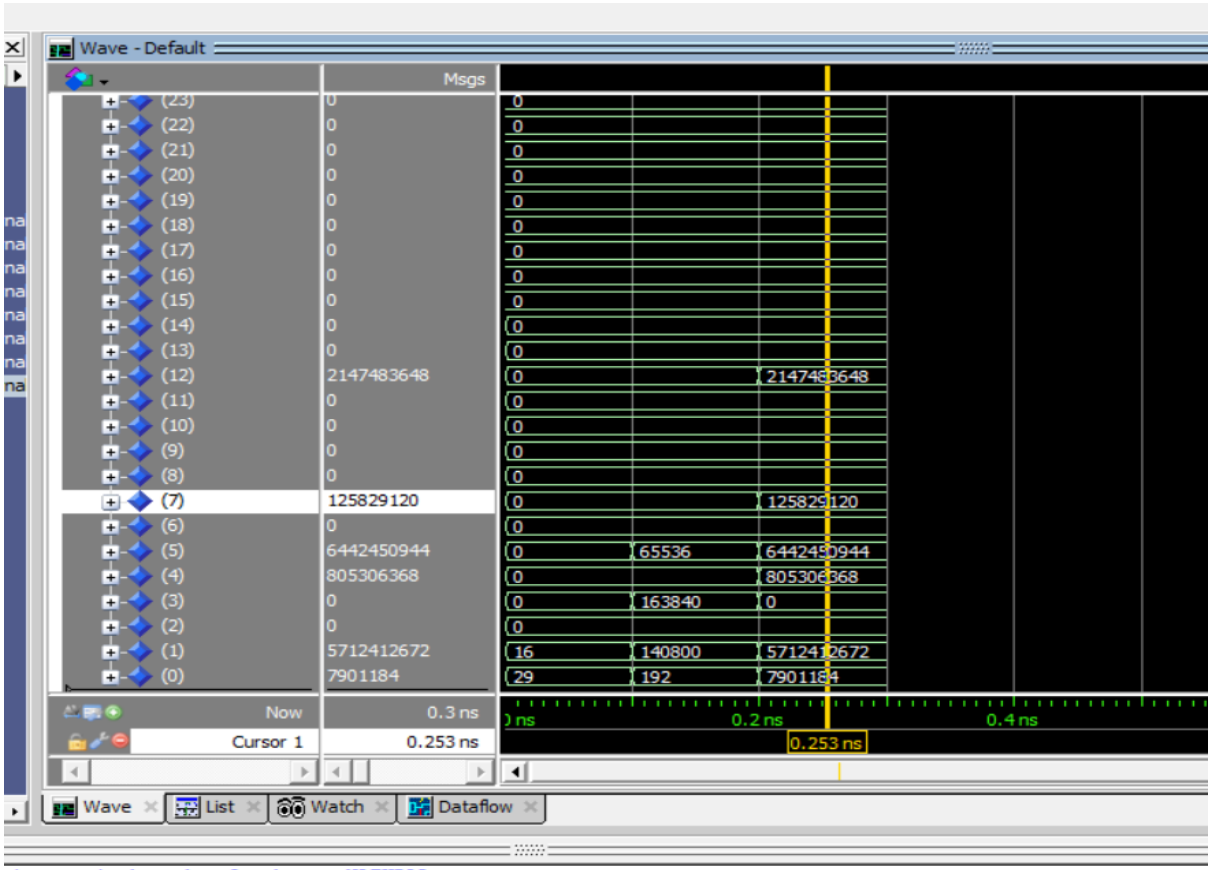
Partial Product Generation



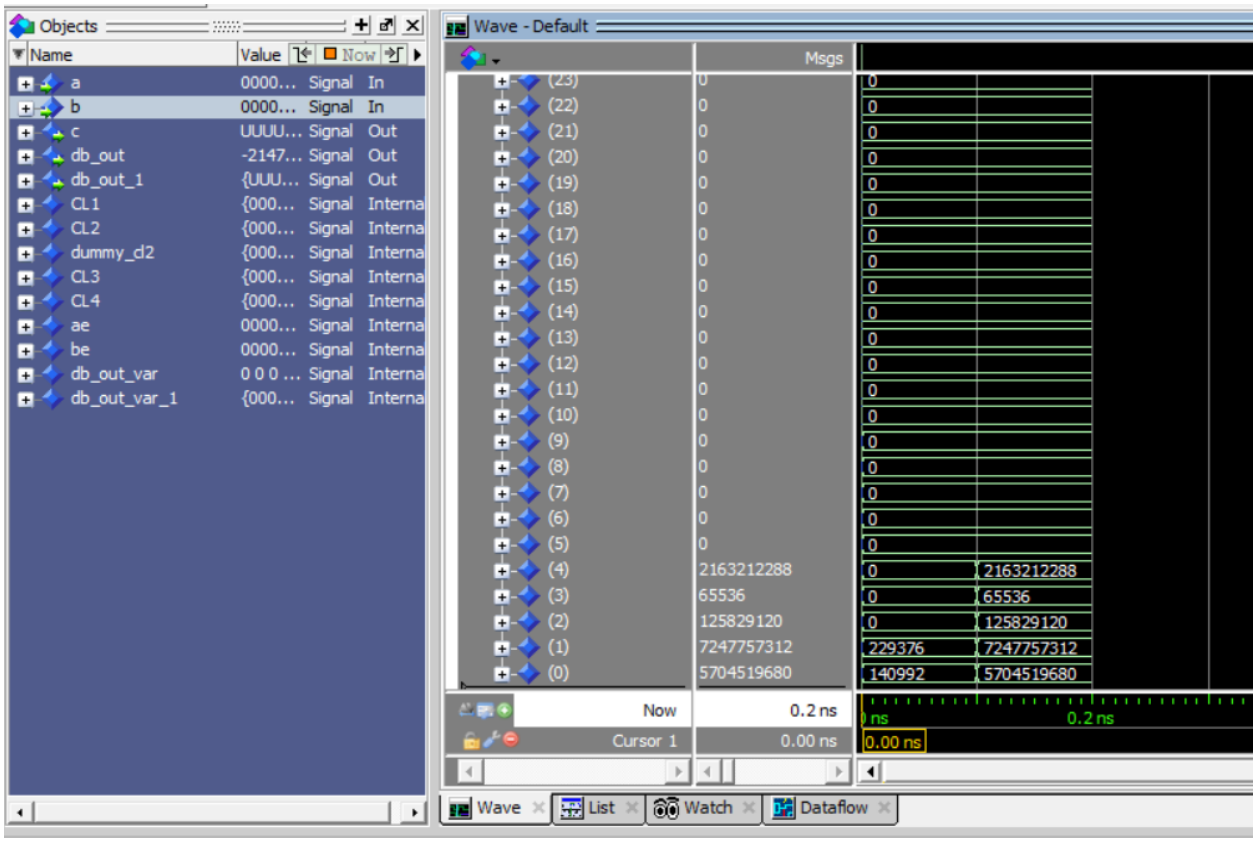
Layer 1 reduction



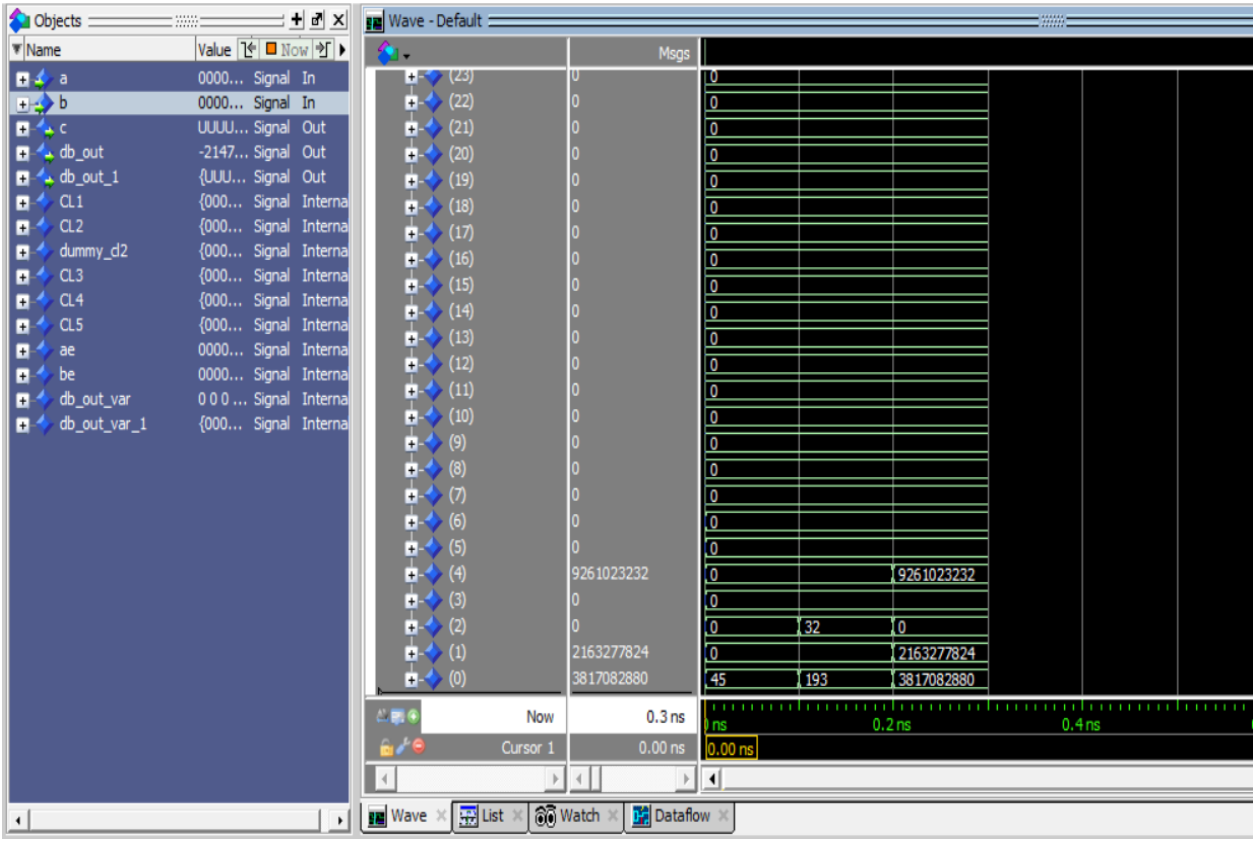
Layer 2 Reduction



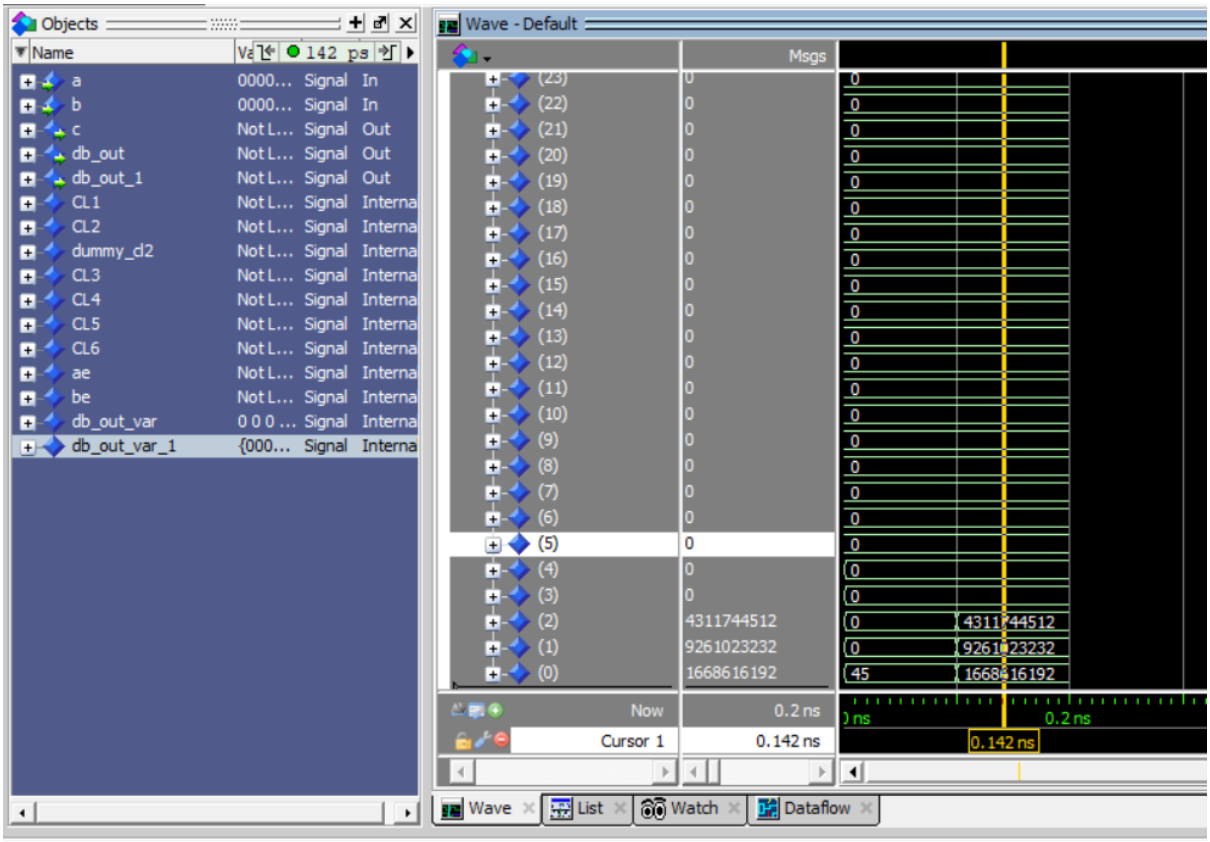
Layer 3 Reduction



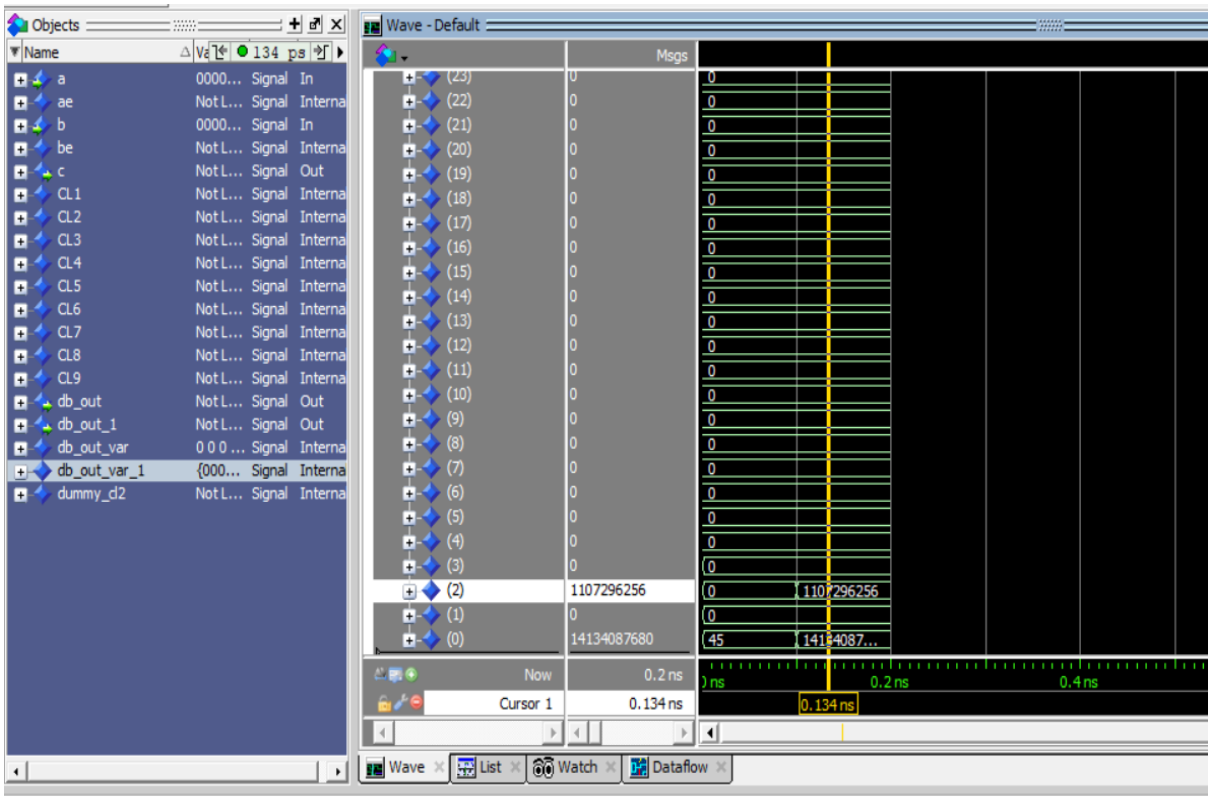
Layer 4 Reduction



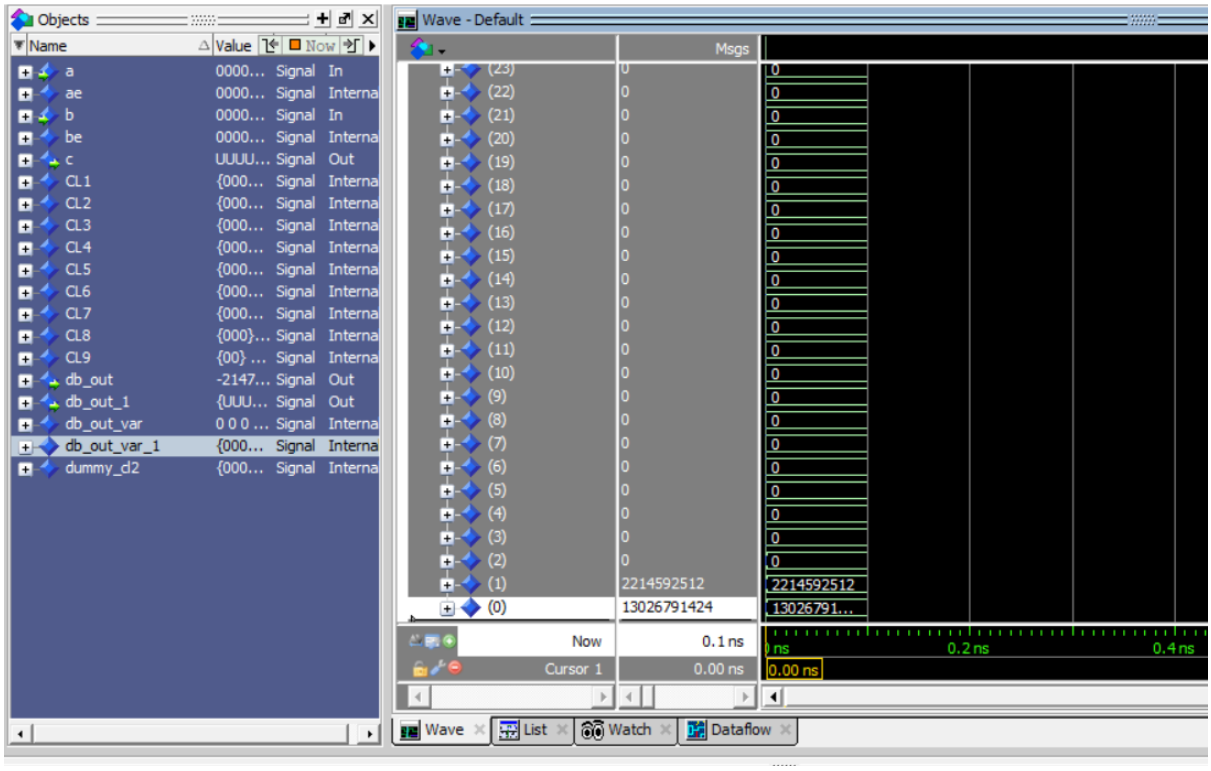
Layer 5 Reduction



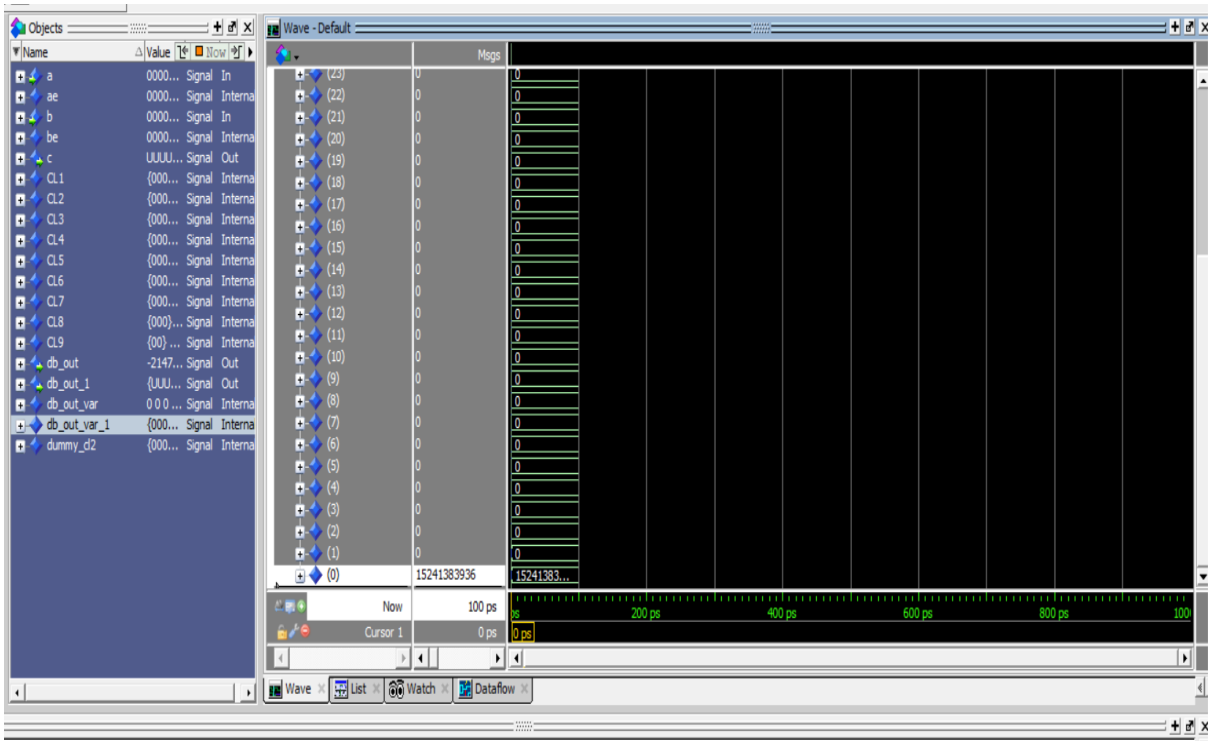
Layer 6 Reduction



Layer 7 Reduction



Layer 8 Reduction



Final Result With Adder

Pipelined Version of Wallace Multiplier

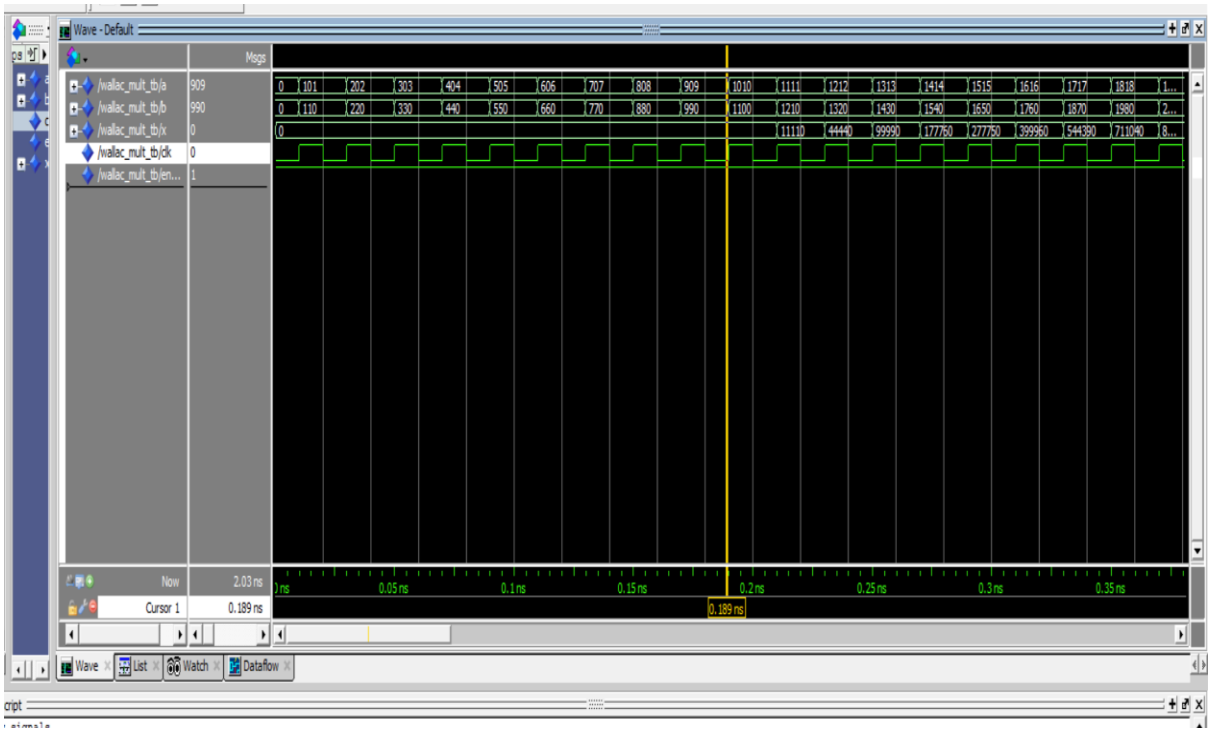
Code Modification:

```
process (clk) begin

if (clk'event and clk = '1') then
 CL1_latched <=  CL1 ;
 CL2_latched <=  CL2 ;
 CL3_latched <=  CL3 ;
 CL4_latched <=  CL4 ;
 CL5_latched <=  CL5 ;
 CL6_latched <=  CL6 ;
 CL7_latched <=  CL7 ;
 CL8_latched <=  CL8 ;
 CL9_latched <=  CL9 ;
 ae(31 downto 0) <= a;
 be(31 downto 0) <= b;
end if ;

end process ;
```

Test Bench:

```
entity wallac_mult_tb is

end entity ;

architecture behave of wallac_mult_tb is


signal a, b: std_logic_vector(31 downto 0) := (others => '0') ;


signal x : std_logic_vector(63 downto 0) ;

signal clk : std_logic := '0' ;
signal enable_clk : std_logic := '1' ;

begin
dut: wallac_mult port map (a => a, b=> b, c => x, clk => clk) ;

clk <= (not clk ) and enable_clk after 10 ps ;


process (clk) begin

if(clk'event and clk = '1') then
a <= std_logic_vector(to_unsigned((to_integer(unsigned(a))+101), 32)) ;
b <= std_logic_vector(to_unsigned((to_integer(unsigned(b))+110), 32)) ;

if (to_integer(unsigned(a)) > 10000) then
enable_clk <= '0' ;
else
enable_clk <= '1' ;
end if ;
end if ;
end process ;
 end architecture ;
```

Result:


Wave - Default
Msgs
/wallac_mult_tb/a
/wallac_mult_tb/b
/wallac_mult_tb/x
/wallac_mult_tb/clk
/wallac_mult_tb/en...
909
990
0
0
1
Now
2.03 ns
Cursor 1
0.189 ns
0.05 ns
0.1 ns
0.15 ns
0.2 ns
0.25 ns
0.3 ns
0.35 ns
0.189 ns
Wave
List
Watch
Dataflow