

**FFT TOP MODULE**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_SIGNED.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity fft is

Port (

clock : in std\_logic;

-- reset : in std\_logic;

--fft\_start\_i : in std\_logic;

fft\_o : out std\_logic\_vector(31 downto 0)

);

end fft;

architecture Behavioral of fft is

COMPONENT dds\_compiler\_0

PORT (

aclk : IN STD\_LOGIC;

s\_axis\_config\_tvalid : IN STD\_LOGIC;

s\_axis\_config\_tdata : IN STD\_LOGIC\_VECTOR(15 DOWNTO 0);

m\_axis\_data\_tvalid : OUT STD\_LOGIC;

m\_axis\_data\_tdata : OUT STD\_LOGIC\_VECTOR(15 DOWNTO 0)

);

END COMPONENT;

COMPONENT my\_fft

PORT (

aclk : IN STD\_LOGIC;

s\_axis\_config\_tdata : IN STD\_LOGIC\_VECTOR(23 DOWNTO 0);

s\_axis\_config\_tvalid : IN STD\_LOGIC;

s\_axis\_config\_tready : OUT STD\_LOGIC;

s\_axis\_data\_tdata : IN STD\_LOGIC\_VECTOR(31 DOWNTO 0);

s\_axis\_data\_tvalid : IN STD\_LOGIC;

s\_axis\_data\_tready : OUT STD\_LOGIC;

s\_axis\_data\_tlast : IN STD\_LOGIC;

m\_axis\_data\_tdata : OUT STD\_LOGIC\_VECTOR(31 DOWNTO 0);

m\_axis\_data\_tuser : OUT STD\_LOGIC\_VECTOR(15 DOWNTO 0);

m\_axis\_data\_tvalid : OUT STD\_LOGIC;

m\_axis\_data\_tready : IN STD\_LOGIC;

m\_axis\_data\_tlast : OUT STD\_LOGIC;

event\_frame\_started : OUT STD\_LOGIC;

event\_tlast\_unexpected : OUT STD\_LOGIC;

event\_tlast\_missing : OUT STD\_LOGIC;

event\_status\_channel\_halt : OUT STD\_LOGIC;

event\_data\_in\_channel\_halt : OUT STD\_LOGIC;

event\_data\_out\_channel\_halt : OUT STD\_LOGIC

);

END COMPONENT;

----------------fft signal-------------------

signal fft\_data\_in : std\_logic\_vector(31 downto 0);

signal fft\_data\_valid\_in : std\_logic;

signal fft\_data\_last\_in : std\_logic;

signal fft\_data\_out : std\_logic\_vector(31 downto 0);

signal fft\_user\_out : std\_logic\_vector(15 downto 0);

signal fft\_data\_valid\_out : std\_logic;

signal fft\_data\_last\_out : std\_logic;

--------------dds signal-----------------------

signal dds\_phase\_in :std\_logic\_vector(15 downto 0) := x"00FF";

signal dds\_phase\_valid\_in : std\_logic := '1';

signal dds\_valid\_out : std\_logic;

signal dds\_out : std\_logic\_vector(15 downto 0);

------------------------------------------

begin

fft\_o <= fft\_data\_out;

-------------------fft--------------------

fft : my\_fft

PORT MAP (

aclk => clock,

s\_axis\_config\_tdata =>(others => '0'),

s\_axis\_config\_tvalid => '0',

s\_axis\_config\_tready => open,

s\_axis\_data\_tdata => fft\_data\_in,

s\_axis\_data\_tvalid => fft\_data\_valid\_in,

s\_axis\_data\_tready => open,

s\_axis\_data\_tlast => fft\_data\_last\_in,

m\_axis\_data\_tdata => fft\_data\_out,

m\_axis\_data\_tuser => fft\_user\_out,

m\_axis\_data\_tvalid => fft\_data\_valid\_out,

m\_axis\_data\_tready => '1',

m\_axis\_data\_tlast => fft\_data\_last\_out,

event\_frame\_started => open,

event\_tlast\_unexpected => open,

event\_tlast\_missing => open,

event\_status\_channel\_halt => open,

event\_data\_in\_channel\_halt => open,

event\_data\_out\_channel\_halt => open

);

--------------------------------------------

---------------------dds--------------------

dds : dds\_compiler\_0

PORT MAP (

aclk => clock,

s\_axis\_config\_tvalid => dds\_phase\_valid\_in ,

s\_axis\_config\_tdata => dds\_phase\_in ,

m\_axis\_data\_tvalid => dds\_valid\_out ,

m\_axis\_data\_tdata => dds\_out

);

fft11 : process (clock)

--variable fft\_counter : integer range 0 to 1024 ;

begin

if (rising\_edge (clock)) then

if fft\_counter < 1024 then

fft\_data\_valid\_in <= '1' ;

fft\_data\_in <= x"0000" & dds\_out ;

fft\_counter := fft\_counter + 1 ;

end if;

-else

-fft\_data\_last\_in <= '0';

end if;

end process;

end Behavioral;

















