Appendix J

Rake Finger Used in Rake Receiver by Using VHDL Code

Program Description

Input:

- Reset signal.
- Frame synchronization signal.
- Clock signal.
- In-phase digital signal (6 bits).
- Quadrature digital signal (6 bits).
- Initial scrambling code (24 bits).
- Spreading factor (3 bits).
- OVSF code number (9 bits).

Output:

- Frame synchronization signal.
- Clock signal of I channel.
- Clock signal of Q channel.
- Equalized I data (14 bits).
- Equalized Q data (14 bits).

```
LIBRARY ieee; USE ieee.std_logic_1164.all; USE
ieee.std_logic_signed.all; USE ieee.std_logic_arith.all;

ENTITY RakeFinger_Master IS
    Port (  reset : IN std_logic;
        FrameSync : IN std_logic;
        clk : IN std_logic;
        I_channel : IN std_logic_vector(5 downto 0);
        Q_channel : IN std_logic_vector(5 downto 0);
        Group_3 : IN std_logic_vector(23 downto 0);
        SF : IN std_logic_vector(2 downto 0);
        OVSF_No : IN std_logic_vector(8 downto 0);

        FrameSync_EqDPCH : OUT std_logic;
        clk_EqData : OUT std_logic;
        clk_EqDPCCH : OUT std_logic;
        EqDPDCH_I : OUT std_logic_vector(13 downto 0);
        EqDPCCH_Q : OUT std_logic_vector(13 downto 0)

        -- From Path Searcher --
        --SL0_FrameSync : OUT std_logic;
        --SL1_FrameSync : OUT std_logic
        );
END RakeFinger_Master ;

ARCHITECTURE behavior OF RakeFinger_Master IS
    COMPONENT Decimator_4
    PORT (  reset : IN std_logic;
        clk : IN std_logic;        -- input clk : 15.36 MHz
        FrmSync : IN std_logic;

        clkout : OUT std_logic;
        clkCodeOut : OUT std_logic;
        FrmSyncOut : INOUT std_logic
        );
    END COMPONENT;

    COMPONENT Scram_Gen_Reset
    PORT (  clk : IN std_logic;                -- timing of 38400 chips per frame.
        reset : IN std_logic;
        FrameSync : IN std_logic;
        Group_3 : IN std_logic_vector(23 downto 0);

        Code_I : OUT std_logic_vector(0 to 1);
        Code_Q : OUT std_logic_vector(0 to 1)
        );
    END COMPONENT;

    COMPONENT OVSF_Gen
    PORT (  clk : IN std_logic;            -- timing of 38400 chips per frame.
        reset : IN std_logic;
        FrameSync : IN std_logic;
        SF : IN std_logic_vector(2 downto 0);
        OVSF_No : IN std_logic_vector(8 downto 0);

        OVSF_Code : OUT std_logic_vector(0 to 1)
        );
    END COMPONENT ;

    COMPONENT Despreader
    PORT (  reset : IN std_logic;
        clk : IN std_logic;
        FrameSync : IN std_logic;
```

```
    I_channel : IN std_logic_vector(5 downto 0);
    Q_channel : IN std_logic_vector(5 downto 0);
    SF : IN std_logic_vector(2 downto 0);
    ScCode_I : IN std_logic_vector(0 to 1);
    ScCode_Q : IN std_logic_vector(0 to 1);
        OVSF_Code : IN std_logic_vector(0 to 1);

    -- Output for DPDCH --
    clk_out : OUT std_logic;
    Frame_out : OUT std_logic;
    dataout_I : OUT std_logic_vector(6 downto 0);
    dataout_Q : OUT std_logic_vector(6 downto 0);

    -- Output for DPCCH --
    DPCCH_clk_out : OUT std_logic;
    DPCCH_Frame_out : OUT std_logic;
    DPCCH_I : OUT std_logic_vector(6 downto 0);
    DPCCH_Q : OUT std_logic_vector(6 downto 0)
    );
END COMPONENT;

COMPONENT Ch_estimator
PORT (  clk:in std_logic;
    reset:in std_logic;
    FrameSync:in std_logic;
    Q_data_Re:in std_logic_vector(6 downto 0);
    Q_data_Im:in std_logic_vector(6 downto 0);

    EstImpRe:out std_logic_vector(7 downto 0);
    EstImpIm:out std_logic_vector(7 downto 0)
    );
END COMPONENT;

COMPONENT ChannelEqualizer
PORT (  reset : IN std_logic;
    SF : IN std_logic_vector(2 downto 0);
    -- Inputs from DPDCH --
    Data_clk_In : IN std_logic;
    FrameSync_In : IN std_logic;
    DataIn_I : IN std_logic_vector(6 downto 0);
    DataIn_Q : IN std_logic_vector(6 downto 0);
    -- Inputs from DPCCH --
    DPCCH_clk_In : IN std_logic;
    DPCCH_FrameSync_In : IN std_logic;
    DPCCH_I : IN std_logic_vector(6 downto 0);
    DPCCH_Q : IN std_logic_vector(6 downto 0);
    -- Inputs from Channel Estimator --
    EstImpRe : IN std_logic_vector(7 downto 0);
    EstImpIm : IN std_logic_vector(7 downto 0);

    Data_clk_Out : OUT std_logic;
    DPCCH_clk_Out : OUT std_logic;
    FrameSync_Out : OUT std_logic;
    DataOut_I : OUT std_logic_vector(13 downto 0);
    DataOut_Q : OUT std_logic_vector(13 downto 0)
    );
END COMPONENT;

--COMPONENT Multipath_Searcher
--Port (    reset : IN std_logic;
--  FrameSync : IN std_logic;
--  clk : IN std_logic;
--  I_channel : IN std_logic_vector(5 downto 0);
--  Q_channel : IN std_logic_vector(5 downto 0);
```

```
    --  clk_ChipCodeGen : IN std_logic;
    --  ScCode_I : IN std_logic_vector(0 to 1);
    --  ScCode_Q : IN std_logic_vector(0 to 1);

    --  SL0_FrameSync : OUT std_logic;
    --  SL1_FrameSync : OUT std_logic
    --  );
    --END COMPONENT;

    --(Output of Decimator)--
    signal FrameSync_Chip : std_logic := '0';
    signal clk_Chip : std_logic := '0';     -- ChipTiming for Despreader : 38400 chips/frame.
    signal clk_ChipCodeGen : std_logic := '0';      -- ChipTiming for CodeGenerator : 3.84 MHz

    --(Output of Sc&OVSF Generator)--
    signal ScCode_I : std_logic_vector(0 to 1);
    signal ScCode_Q : std_logic_vector(0 to 1);
    signal OVSF_Code : std_logic_vector(0 to 1);

    --(Output of Despreader)--
    -- Output for DPDCH --
    signal DPDCH_clk : std_logic := '0';
    signal FrameSync_Data : std_logic := '0';
    signal DPDCH_I : std_logic_vector(6 downto 0);
    signal DPDCH_Q : std_logic_vector(6 downto 0);
    -- Output for DPCCH --
    signal FrameSync_DPCCH : std_logic := '0';
    signal DPCCH_clk : std_logic;
    signal DPCCH_I : std_logic_vector(6 downto 0);
    signal DPCCH_Q : std_logic_vector(6 downto 0);

    --(Output of ChannelEstimator)--
    signal Imp_I : std_logic_vector(7 downto 0):="00000000";
    signal Imp_Q : std_logic_vector(7 downto 0):=(others=>'0');

    FOR Decimator: Decimator_4 USE ENTITY work.Decimator_4(behavior);
    FOR ScCode_Generator: Scram_Gen_Reset USE ENTITY work.Scram_Gen_Reset(rtl);
    FOR OVSF_Generator: OVSF_Gen USE ENTITY work.OVSF_Gen(behavior);
    FOR Symbol_Despreader: Despreader USE ENTITY work.Despreader(behavior);
    FOR Channel_Estimator: Ch_estimator USE ENTITY work.Ch_Estimator(behavior);
    FOR Channel_Equalizer: ChannelEqualizer USE ENTITY work.ChannelEqualizer(behavior);
    --FOR Path_Searcher: Multipath_Searcher USE ENTITY work.Multipath_Searcher(behavior);

BEGIN
    Decimator: Decimator_4
        PORT MAP(reset,
             clk,
             FrameSync,
             clk_Chip,
             clk_ChipCodeGen,
             FrameSync_Chip
             );

    ScCode_Generator: Scram_Gen_Reset
        PORT MAP (clk => clk_ChipCodeGen,
              reset => reset,
              FrameSync => FrameSync,
              Group_3 => Group_3,
              Code_I => ScCode_I,
              Code_Q => ScCode_Q
              );

    OVSF_Generator: OVSF_Gen
        PORT MAP (clk => clk_ChipCodeGen,
```

```
            reset => reset,
            FrameSync => FrameSync,
            SF => SF,
            OVSF_No => OVSF_No,
            OVSF_Code => OVSF_Code
            );

    Symbol_Despreader: Despreader
        PORT MAP (reset => reset,
            clk => clk_Chip,
            FrameSync => FrameSync_Chip,
            I_channel => I_channel,
            Q_channel => Q_Channel,
            SF => SF,
            ScCode_I => ScCode_I,
            ScCode_Q => ScCode_Q,
            OVSF_Code => OVSF_Code,
            clk_out => DPDCH_clk,
            Frame_out => FrameSync_Data,
            dataout_I => DPDCH_I,
            dataout_Q => DPDCH_Q,
            DPCCH_clk_out => DPCCH_clk,
            DPCCH_Frame_out => FrameSync_DPCCH,
            DPCCH_I => DPCCH_I,
            DPCCH_Q => DPCCH_Q
            );

    Channel_Estimator: Ch_estimator
        PORT MAP (clk => DPCCH_clk,
            reset => reset,
            FrameSync => FrameSync_DPCCH,
            Q_data_Re => DPCCH_I,
            Q_data_Im => DPCCH_Q,
            EstImpRe => Imp_I,
            EstImpIm => Imp_Q
            );

    Channel_Equalizer: ChannelEqualizer
        PORT MAP (reset => reset,
            SF => SF,
           -- Inputs from DPDCH --
            Data_clk_In => DPDCH_clk,
            FrameSync_In => FrameSync_Data,
            DataIn_I => DPDCH_I,
            DataIn_Q => DPDCH_Q,
           -- Inputs from DPCCH --
            DPCCH_clk_In => DPCCH_clk,
            DPCCH_FrameSync_In => FrameSync_DPCCH,
            DPCCH_I => DPCCH_I,
            DPCCH_Q => DPCCH_Q,
           -- Inputs from Channel Estimator --
            EstImpRe => Imp_I,
            EstImpIm => Imp_Q,

            Data_clk_Out => clk_EqData,
            DPCCH_clk_Out => clk_EqDPCCH,
            FrameSync_Out => FrameSync_EqDPCH,
            DataOut_I => EqDPDCH_I,
            DataOut_Q => EqDPCCH_Q
            );
    --Path_Searcher:
    Multipath_Searcher PORT MAP (reset,FrameSync,clk,I_channel,Q_channel,
    clk_ChipCodeGen,ScCode_I,ScCode_Q,SL0_FrameSync,SL1_FrameSync);
END behavior;
```