datapath.vhd

```
----------------------------------------
--      Author:      Roger Grayson
--      Date: Feb, 2003
--      Purpose:Datapath for my implementation
--                   of the Coulston computer
----------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_signed.conv_integer;
use work.my_stuff.all;       --my package

entity DataPath is
    port(   control_word: in std_logic_vector(20 downto 0);
            status: out std_logic_vector(6 downto 0);
            clk, reset: in std_logic);
end DataPath;

architecture behavior of DataPath is

signal A_bus,B_bus,R_bus: std_logic_vector(31 downto 0);
signal RFoutA,RFoutB,MARout,MBRout,TSBout,Data:std_logic_vector(31 downto 0);
signal PCout,IRout,MARMout,MBRMout,RAMinout,ALUout:std_logic_vector(31 downto 0);
signal RFa,RFb,RFr,RFWE,PC,IR,MAR,MARM,MBR,MBRM,TSB,CS,RE,WE,BBUS:std_logic;
signal ALUop,ABUS,RBUS:std_logic_vector(1 downto 0);
constant four: std_logic_vector(31 downto 0) := x"00000001";
signal IR15_0:std_logic_vector(31 downto 0);
signal RegA,RegB,Wreg:std_logic_vector(4 downto 0);
signal eq:std_logic;
begin
    IR15_0 <= x"0000" & IRout(15 downto 0);
    --control word reassignments
    RFa <= control_word(20);
    RFb <= control_word(19);
    Rfr <= control_word(18);
    RFWE <= control_word(17);
    ALUop <= control_word(16 downto 15);
    PC <= control_word(14);
    IR <= control_word(13);
    MAR <= control_word(12);
    MARM <= control_word(11);
    MBR <= control_word(10);
    MBRM <= control_word(9);
    TSB <= control_word(8);
    CS <= control_word(7);
    RE <= control_word(6);
    WE <= control_word(5);
    ABUS <= control_word(4 downto 3);
    BBUS <= control_word(2);
    RBUS <= control_word(1 downto 0);
    --mux port mappings
    Bbusmux:muxNx2x1 port map(RFoutB,PCout,BBUS,B_bus);
    Abusmux:muxNx4x1 port map(RFoutA,IR15_0,four,PCout,ABUS,A_bus);
```

```
    MARMux:muxNx2x1 port map(B_bus,R_bus,MARM,MARMout);
    MBMMux:muxNx2x1 port map(A_bus,Data,MBRM,MBRMout);
    Rbusmux:muxNx4x1 port map(ALUout,IRout,IRout,MBRout,RBUS,R_bus);
    RFamux:muxNx2x1 generic map (5) port map(IRout(25 downto 21),IRout(20 downto 16),RFa,
RegA);
    RFbmux:muxNx2x1 generic map (5) port map(IRout(25 downto 21),IRout(20 downto 16),RFb,
RegB);
    RFrmux:muxNx2x1 generic map (5) port map(IRout(15 downto 11),IRout(20 downto 16),RFr,
Wreg);
    --register file port mapping
    regfile:reg_file port map(clk,reset,RFWE,Wreg,RegA,RegB,R_bus,RFoutA,RFoutB);
    --ALU port mapping
    myALU:ALU port map(ALUop,eq,IRout,A_bus,B_bus,ALUout);
    --RAM port mapping
    myRAM:ram port map(MARout(9 downto 0),RAMinout,RE,WE,CS,reset);
    --PC, IR, MAR, and MBR registers
    PCreg:reg port map(clk,reset,PC,R_bus,PCout);
    IRreg:reg port map(clk,reset,IR,R_bus,IRout);
    MAreg:reg port map(clk,reset,MAR,MARMout,MARout);
    MBreg:reg port map(clk,reset,MBR,MBRMout,MBRout);
    --tri-state buffer
    TSBpm:tsb port map(MBRout,TSB,TSBout);
    Data <= (TSBout or RAMinout);
    --assign status word
    status <= eq & IRout(31 downto 26);
end behavior;
```