```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;

entity MIPS is
  port(
    rst : in std_logic;
    clk : in std_logic
    );

end MIPS;

-- purpose: put all the components together
architecture str of MIPS is

  component Datapath
    port(
      rst : in std_logic;
      clk : in std_logic;

      -- Inputs
      ALUSrcA  : in std_logic;
      ALUSrcB  : in std_logic_vector( 1 downto 0 );
      ALUCtrl  : in std_logic_vector( 2 downto 0 );
      RegWrite : in std_logic;
      RegDst   : in std_logic_vector( 1 downto 0 );
      PCSource : in std_logic_vector( 1 downto 0 );
      PCLoad   : in std_logic;
      IorD     : in std_logic;
      MemRead  : in std_logic;
      MemWrite : in std_logic;
      MemtoReg : in std_logic_vector( 1 downto 0 );
      IRWrite  : in std_logic;
      Extend   : in std_logic;
      ReadSrc  : in std_logic;
      PCWriteCond : in std_logic_vector( 2 downto 0 );

      -- Outputs
      OpCode : out std_logic_vector( 5 downto 0 );
      funct  : out std_logic_vector( 5 downto 0 );
      BranchType : out std_logic
      );

  end component;

  component Control
    port(
      rst : in std_logic;
      clk : in std_logic;

      -- Outputs
      ALUSrcA  : out std_logic;
      ALUSrcB  : out std_logic_vector( 1 downto 0 );
      ALUOp    : out std_logic_vector( 2 downto 0 );
      RegWrite : out std_logic;
      RegDst   : out std_logic_vector( 1 downto 0 );
      PCSource : out std_logic_vector( 1 downto 0 );
      PCLoad   : out std_logic;
      IorD     : out std_logic;
      MemRead  : out std_logic;
      MemWrite : out std_logic;
      MemtoReg : out std_logic_vector( 1 downto 0 );
      IRWrite  : out std_logic;
      Extend   : out std_logic;
```

```
      ReadSrc  : out std_logic;
      PCWriteCond : out std_logic_vector( 2 downto 0 );

      -- Inputs
      OpCode : in std_logic_vector( 5 downto 0 );
      BranchType   : in std_logic;
      funct  : in std_logic_vector( 5 downto 0 )
      );

  end component;

  component ALUControl
    port(
      -- Inputs
      ALUOp : in std_logic_vector( 2 downto 0 );
      funct : in std_logic_vector( 5 downto 0 );

      -- Outputs
      ALUCtrl : out std_logic_vector( 2 downto 0 )
      );

  end component;

-- Internal signals
  signal ALUSrcA                        : std_logic;
  signal ALUSrcB                        : std_logic_vector( 1 downto 0 );
  signal RegWrite                       : std_logic;
  signal RegDst                 : std_logic_vector( 1 downto 0 );
  signal PCSource                       : std_logic_vector( 1 downto 0 );
  signal PCLoad                 : std_logic;
  signal IorD                   : std_logic;
  signal MemRead                        : std_logic;
  signal MemWrite                       : std_logic;
  signal MemtoReg                       : std_logic_vector( 1 downto 0 );
  signal IRWrite                        : std_logic;
  signal ALUOp                  : std_logic_vector(2 downto 0 );
  signal OpCode                 : std_logic_vector(5 downto 0);
  signal funct                  : std_logic_vector(5 downto 0);
  signal ALUCtrl                        : std_logic_vector(2 downto 0);
  signal Extend                 : std_logic;
  signal PCWriteCond                    : std_logic_vector( 2 downto 0 );
  signal BranchType                     : std_logic;
  signal ReadSrc                        : std_logic;

begin  -- bhv

  -- Control
  Control_Ins : Control port map(
    rst      => rst,
    clk      => clk,
    ALUSrcA  => ALUSrcA,
    ALUSrcB  => ALUSrcB,
    ALUOp    => ALUOp,
    RegWrite => RegWrite,
    RegDst   => RegDst,
    PCSource => PCSource,
    PCLoad   => PCLoad,
    IorD     => IorD,
    MemRead  => MemRead,
    MemWrite => MemWrite,
    MemtoReg => MemtoReg,
    IRWrite  => IRWrite,
    OpCode   => OpCode,
    PCWriteCond  => PCWriteCond,
```

```
    BranchType   => BranchType,
    Extend       => Extend,
    funct        => funct,
    ReadSrc              => ReadSrc);

  -- ALU Control
  ALUControl_Ins : ALUControl port map(
    ALUOp   => ALUOp,
    funct   => funct,
    ALUCtrl => ALUCtrl);

  -- Datapath
  Datapath_Ins : Datapath port map(
    rst      => rst,
    clk      => clk,
    ALUSrcA  => ALUSrcA,
    ALUSrcB  => ALUSrcB,
    ALUCtrl  => ALUCtrl,
    RegWrite => RegWrite,
    RegDst   => RegDst,
    PCSource => PCSource,
    PCLoad   => PCLoad,
    IorD     => IorD,
    MemRead  => MemRead,
    MemWrite => MemWrite,
    MemtoReg => MemtoReg,
    IRWrite  => IRWrite,
    OpCode   => OpCode,
    funct    => funct,
    PCWriteCond  => PCWriteCond,
    BranchType   => BranchType,
    Extend       => Extend,
    ReadSrc              => ReadSrc);

end str;
```