```
--
-- MIPs Processor Developement
-- Eric W. Johnson
-- Valparaiso University
--
--
-- Modified by Jeffrey D. Will


library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use work.mips_types.all;

entity control is
   port( INSTR: in std_logic_vector(5 DOWNTO 0);
         REGDST: OUT std_logic;
         BRANCH: OUT std_logic;
         MEMREAD: OUT std_logic;
         MEMTOREG: OUT std_logic;
         ALUOP: OUT std_logic_vector(1 DOWNTO 0);
         MEMWRITE: OUT std_logic;
         ALUSRC: OUT std_logic;
         REGWRITE: OUT std_logic);
end control;

architecture behavioral of control is
begin
  PROCESS( INSTR )
     variable temp : std_logic_vector ( 7 downto 0 );
     variable instr_int : integer range 0 to 64;
  BEGIN
     temp := "00000000";
     temp := temp + instr;
     instr_int := conv_Integer(temp);
     case instr_int is

       -- ******************************************
       -- Students:  YOU MUST FILL IN THESE CONTROL LINES
       -- FOR EACH INDIVIDUAL INSTRUCTION... they are
       -- set to all zeroes below, but the zeroes are
       -- just placeholders...
       -- ******************************************
    when 0 => REGDST <= '1';
                  BRANCH <= '0';
                  MEMREAD <= '0';
                  MEMTOREG <= '0';
                  ALUOP <= "10";            -- ALUOP ORDER OF BITS????
                  MEMWRITE <= '0';
                  ALUSRC <= '0';
                  REGWRITE <= '1';
        when 35 => REGDST <= '0';
                  BRANCH <= '0';
                  MEMREAD <= '1';
                  MEMTOREG <= '1';
                  ALUOP <= "00";
                  MEMWRITE <= '0';
                  ALUSRC <= '1';
                  REGWRITE <= '1';
        when 43 => REGDST <= '0';
                  BRANCH <= '0';
                  MEMREAD <= '0';
                  MEMTOREG <= '0';
                  ALUOP <= "00";
                  MEMWRITE <= '1';
                  ALUSRC <= '1';
                  REGWRITE <= '0';
        when 4 => REGDST <= '0';
```

```
70                      BRANCH <= '1';
71                      MEMREAD <= '0';
72                      MEMTOREG <= '0';
73                      ALUOP <= "01";
74                      MEMWRITE <= '0';
75                      ALUSRC <= '0';
76                      REGWRITE <= '0';
77              when others => REGDST <= '0';
78                      BRANCH <= '0';
79                      MEMREAD <= '0';
80                      MEMTOREG <= '0';
81                      ALUOP <= "00";
82                      MEMWRITE <= '0';
83                      ALUSRC <= '0';
84                      REGWRITE <= '0';
85
86          END CASE;
87
88        END PROCESS;
89
90      end behavioral;
91
```