```
----------------------------------------------------------------------------------
----------------
-- Company:
-- Engineer:
--
-- Create Date: 12/15/2020 04:54:50 PM
-- Design Name:
-- Module Name: DCDI - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
----------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;



entity DCDI is
  Port (DataIN: in std_logic_vector(31 downto 0);
        RegWr: out std_logic;
        AdrD: out std_logic_vector(3 downto 0);
        MxD: out std_logic_vector(1 downto 0);
        SSalt: out std_logic_vector(1 downto 0);
        CSalt: out std_logic;
        MemWr: out std_logic;
        OpUAL: out std_logic_vector(3 downto 0);
        MxA: OUT STD_LOGIC;
        MxB: out std_logic;
        AdrSA: out std_logic_vector(3 downto 0);
```

```
        AdrSB: out std_logic_vector(3 downto 0);
        SelC: out std_logic
        );
end DCDI;

architecture Behavioral of DCDI is
signal vectorCMD: std_logic_vector(9 downto 0);
signal CODOP: std_logic_vector(7 downto 0);
begin
CODOP<= DataIN(31 downto 24);
AdrSA <= DataIN(19 downto 16);
AdrSB <= DataIN(15 downto 12);
AdrD <= DataIN(23 downto 20);
OpUAL <= DataIN(28 downto 25);
codopProc: process(CODOP)
begin
    case CODOP is
        when b"0000_0000" => vectorCMD <= b"00_0000_0000"; --NOP
        when b"0100_0000" => vectorCMD <= b"10_0000_0000"; --MOVA
        when b"0000_0010" => vectorCMD <= b"10_0000_0000"; --ADD
        when b"0000_0101" => vectorCMD <= b"10_0000_0000"; --SUB
        when b"0000_1000" => vectorCMD <= b"10_0000_0000"; --AND
        when b"0000_1001" => vectorCMD <= b"10_0000_0000"; --OR
        when b"0000_1010" => vectorCMD <= b"10_0000_0000"; --XOR
        when b"0000_1011" => vectorCMD <= b"10_0000_0000"; --NOT
        when b"0000_1101" => vectorCMD <= b"10_0000_0000"; --SHR
        when b"0000_1110" => vectorCMD <= b"10_0000_0000"; --SHL
        when b"0010_0010" => vectorCMD <= b"10_0000_0011"; --ADDI
        when b"0010_0101" => vectorCMD <= b"10_0000_0011"; --SUBI
        when b"0010_1000" => vectorCMD <= b"10_0000_0010"; --ANDI
        when b"0010_1001" => vectorCMD <= b"10_0000_0010"; --ORI
        when b"0010_1010" => vectorCMD <= b"10_0000_0010"; --XORI
        when b"0110_0000" => vectorCMD <= b"00_0010_0011"; --BZ
        when b"0101_0000" => vectorCMD <= b"00_0011_0011"; --BNZ
        when b"0110_1001" => vectorCMD <= b"00_0111_0000"; --HALT

    end case;
end process;

RegWr <= vectorCmd(9);
```

```
MxD <= vectorCmd(8 downto 7);
SSalt <= vectorCmd(6 downto 5);
CSalt <= vectorCmd(4);
MemWr <= vectorCmd(3);
MxA <= vectorCmd(2);
MxB <= vectorCmd(1);
SelC <= vectorCmd(0);

end Behavioral;
```