8重3-1多路复用器

```
library IEEE;
use IEEE.std_logic_1164.all;
entity mux83_1 is
port(d0,d1,d2:in STD_LOGIC_VECTOR(7 downto 0);
        sel :in STD_LOGIC_VECTOR(1 downto 0);
        dout:out STD_LOGIC_VECTOR(7 downto 0));
end mux83_1;
architecture rtl of mux83_1 is
begin
    dout <=
    d0 when sel="00" else
    d1 when sel="01" else
    d2;
end rtl;
```

参数化多路复用器及定制为8-1多路复用器

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity index_add is
generic (n: integer := 3);
port(cin: in std_logic_vector (2**n-1 downto 0);
     sel: in std_logic_vector (n-1 downto 0);
    cout:out std_logic);
end index_add;

architecture rtl of index_add is
begin
    cout <= cin(conv_integer(sel));
end rtl;
```

控制信号产生逻辑的VHDL程序

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity Instruction_logic is
    port(MOVA, MOVB, MOVC, ALU, NOTR, SH, JMP, JZ, Z, JC, C, INR, OUTR, NOP,
HALT: in std_logic;
         IR: in std_logic_vector(7 downto 0);
         SM: in std_logic;
         RA, WA, MADD: out std_logic_vector(1 downto 0);
         S: out std_logic_vector(3 downto 0);
         PCLD, PCIN, WE,XL,DL,ALUM,FBUS,FR,FL,LDIR,CFE,ZFE,SME: out std_logic);
end Instruction_logic;
```

```vhdl
architecture rtl of Instruction_logic is
begin
    RA  <= IR(1 downto 0);
    WA  <= IR(3 downto 2);
    MADD  <= "00" when SM='0' and HALT='1'   else
             "01" when MOVB='1' and HALT='1' else
             "10" when MOVC='1' and HALT='1' else "11";
    S  <= IR(7 downto 4);
    PCLD  <= SM and (not NOP) and (JMP or (JZ and Z) or (JC and C)) and HALT;
    PCIN  <= SM and not(JMP or (JZ and Z) or (JC and C)) and HALT;
    WE  <= not(SM and (MOVA or MOVC or ALU or SH or NOTR or INR)) and HALT;
    XL  <= SM and MOVB and HALT;
    DL  <= not(SM and MOVB) and HALT;
    ALUM  <= not ALU and HALT;
    FBUS  <= SM and (not SH) and HALT;
    FR  <= '1' when SM='1' and IR(1 downto 0)="00" and HALT='1' else '0';
    FL  <= '1' when SM='1' and IR(1 downto 0)="11" and HALT='1' else '0';
    LDIR  <= not SM and HALT;
    CFE  <= '0';
    ZFE  <= '0';
    SME  <= '0';
end rtl;
```