```vhdl

--------------------------------Multiplexer (Main Selector of Operation---------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
--ENTITY
entity mux_main is
   port(in_add, in_sub, in_mul, in_div, fl_add, fl_sub, fl_mul, fl_div : in unsigned(7 downto 0);
         in_add_fg,in_sub_fg,in_mul_fg,in_div_fg,fl_add_fg,fl_sub_fg,fl_mul_fg,fl_div_fg,clk:in std_logic;
      op_code:in unsigned( 2 downto 0 );
      res:out unsigned(7 downto 0);
      suc_flag:out std_logic);
end mux_main;
--ARCHITECTURE
architecture Behavioral of mux_main is
begin
process(clk)
begin
--POSITIVE EDGE TRIGGERED
   if (clk='1') then
--SELECTING RESULT WITH OPCODE
      case op_code is
          when "000" =>res<=in_add;
          when "001" =>res<=in_sub;
          when "010" =>res<=in_mul;
          when "011" =>res<=in_div;
          when "100" =>res<=fl_add;
          when "101" =>res<=fl_sub;
          when "110" =>res<=fl_mul;
          when others =>res<=fl_div;
      end case;
--SELCTING SUCESS FLAGS USING OPCODE
      case op_code is
          when "000" =>suc_flag <=in_add_fg;
          when "001" =>suc_flag<=in_sub_fg;
          when "010" =>suc_flag<=in_mul_fg;
          when "011" =>suc_flag<=in_div_fg;
          when "100" =>suc_flag<=fl_add_fg;
          when "101" =>suc_flag<=fl_sub_fg;
          when "110" =>suc_flag<=fl_mul_fg;
          when others =>suc_flag<=fl_div_fg;
         end case;
   end if;
end process;
--END ARCHITECTURE
end Behavioral;


```