```vhdl
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY ALUMUX IS
    PORT (
    ALUOP_IN        :  IN STD_LOGIC_VECTOR(2 DOWNTO 0);
    AND_BUFFER_IN   :  IN STD_LOGIC_VECTOR(31 DOWNTO 0):=
    "00000000000000000000000000000000";
    SRL_BUFFER_IN   :  IN STD_LOGIC_VECTOR(31 DOWNTO 0):=
    "00000000000000000000000000000000";
    SRA_BUFFER_IN   :  IN STD_LOGIC_VECTOR(31 DOWNTO 0):=
    "00000000000000000000000000000000";
    SLL_BUFFER_IN   :  IN STD_LOGIC_VECTOR(31 DOWNTO 0):=
    "00000000000000000000000000000000";
    OOR_BUFFER_IN   :  IN STD_LOGIC_VECTOR(31 DOWNTO 0):=
    "00000000000000000000000000000000";
    ADD_BUFFER_IN   :  IN STD_LOGIC_VECTOR(31 DOWNTO 0):=
    "00000000000000000000000000000000";
    SUB_BUFFER_IN   :  IN STD_LOGIC_VECTOR(31 DOWNTO 0):=
    "00000000000000000000000000000000";
    CARRY_ADD_IN    :  IN STD_LOGIC;
    CARRY_SUB_IN    :  IN STD_LOGIC;
    ZERO_ADD_IN     :  IN STD_LOGIC;
    ZERO_SUB_IN     :  IN STD_LOGIC;
    OVER_SUB_IN     :  IN STD_LOGIC;
    OVER_ADD_IN     :  IN STD_LOGIC;
    CLOCK_IN        :  IN STD_LOGIC;
    RESULT_MUX      : OUT STD_LOGIC_VECTOR(31 DOWNTO 0):=
    "00000000000000000000000000000000";
    ZERO_MUX        : OUT STD_LOGIC := '0';
    CARRY_MUX       : OUT STD_LOGIC := '0';
    OVER_MUX        : OUT STD_LOGIC := '0');
END ALUMUX;

ARCHITECTURE STRUCTURAL OF ALUMUX IS

BEGIN
  PROCESS(CLOCK_IN)
    BEGIN
      IF (CLOCK_IN = '1') THEN
      CASE ALUOP_IN is
        WHEN "000" => -- ADDER
           RESULT_MUX <= ADD_BUFFER_IN;
           CARRY_MUX  <= CARRY_ADD_IN;
           ZERO_MUX   <= ZERO_ADD_IN;
           OVER_MUX   <= OVER_ADD_IN;
        WHEN "001" => -- SUBBER
           RESULT_MUX <= SUB_BUFFER_IN;
           CARRY_MUX  <= CARRY_SUB_IN;
           ZERO_MUX   <= ZERO_SUB_IN;
           OVER_MUX   <= OVER_SUB_IN;
        WHEN "010" => -- ANDDER
           RESULT_MUX <= AND_BUFFER_IN;
        WHEN "011" => -- ORRER
           RESULT_MUX <= OOR_BUFFER_IN;
        WHEN "100" => -- shift_left_logical
           RESULT_MUX <= SLL_BUFFER_IN;
        WHEN "101" => -- shift_Right_logical
           RESULT_MUX <= SRL_BUFFER_IN;
        WHEN "110" => -- shift_LEFT_arithmatic
           RESULT_MUX <= SLL_BUFFER_IN;
        WHEN "111" => -- shift_right_arithmatic
           RESULT_MUX <= SRA_BUFFER_IN;
        WHEN OTHERS => -- SHOULD NOT HAPPEN
      END CASE;
    END IF;
END PROCESS;
END STRUCTURAL;
```