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

entity bit_slice is
  port(
    a,b,Less,Ainvert,carryIn  : in std_logic;
    Op                        : in std_logic_vector (2 downto 0);
    result,carryOut           : out std_logic
  );
end bit_slice;

architecture stdBitSlice of bit_slice is
  signal aMux, bMux, andOut, orOut  : std_logic;
  signal AxorB, AandB, sum, CandXor : std_logic;
begin
  --Inverter muxes
  aMux <= a xor Ainvert;
  bMux <= b xor Op(2);              -- Assuming Binvert is MSB of Op

  --Basic gates
  andOut <= aMux and bMux;
  orOut <= aMux or bMux;

  --Following is adder block
  AxorB <= aMux xor bMux;
  AandB <= aMux and bMux;
  CandXor <= AxorB and carryIn;
  sum <= AxorB xor carryIn ;
  carryOut <= AandB or CandXor;

  --Big mux
  result <= (not Op(1) and not Op(0) and andOut) or
            (not Op(1) and Op(0) and orOut) or
            (Op(1) and not Op(0) and sum) or
            (Op(1) and Op(0) and Less);

end stdBitSlice;

```