```verilog
module RegFile_Alu (RdestRegLoc, RsrcRegLoc, Clk, En, Rst, Imm, Imm_s, OpCode, RdestOut, Flags);

   input [3:0] RdestRegLoc, RsrcRegLoc;
   input Clk, En, Rst, Imm_s; //Imm_s is 1 when we want to use the Imm value, else its 0 if we want to use RegSrc
   input [15:0] Imm; //the 16 bit immediate value

   output [15:0] RdestOut; //feeds in to Reg load and is outputed from the ALU

   input [4:0] OpCode;
   output [4:0] Flags;

   wire [15:0] RsrcOut, AluOutput;
   reg [15:0] AluSrcIn;

   RegFile myReg(
      .RdestRegLoc(RdestRegLoc),
      .RsrcRegLoc(RsrcRegLoc),
      .Clk(Clk),
      .En(En),
      .Rst(Rst),
      .Load(AluOutput),
      .RdestOut(RdestOut),
      .RsrcOut(RsrcOut)
   );

   ALU myALU(
      .Rsrc(AluSrcIn),
      .Rdest(RdestOut),
      .OpCode(OpCode),
      .Flags(Flags),
      .Out(AluOutput)
   );

   always @(*) begin
      if (~Imm_s)
         AluSrcIn <= RsrcOut;

      else begin
         AluSrcIn <= Imm;
      end
   end

endmodule

```