算術論理演算回路

入出力ポート

	ポート名	ビット数	
入力	a	16	オペランド（２項演算の場合は第２オペランド）
入力	b	16	２項演算の場合の第１オペランド
入力	f	5	演算の種類の選択
出力	s	16	演算結果

f が整数演算のとき，a，b，s は２の補数として取り扱います.

ネット

ネット名	値
x	a に 16'h8000 を加算した値
y	b に 16'h8000 を加算した値

ソースコード

```
`include "defs.v"

module alu(a, b, f, s);

 input [15:0] a, b;
 input [4:0]   f;
 output [15:0] s;
 reg  [15:0]    s;
 wire [15:0] x, y;

 assign x = a + 16'h8000;
 assign y = b + 16'h8000;

 always @(a or b or x or y or f)
   case(f)
     `ADD : s = b + a;
     `SUB : s = b - a;
     `MUL : s = b * a;
     `SHL : s = b << a;
     `SHR : s = b >> a;
     `BAND: s = b & a;
     `BOR : s = b | a;
     `BXOR: s = b ^ a;
     `AND : s = b && a;
     `OR  : s = b || a;
     `EQ  : s = b == a;
     `NE  : s = b != a;
     `GE  : s = y >= x;
     `LE  : s = y <= x;
     `GT  : s = y > x;
     `LT  : s = y < x;
     `NEG : s = -a;
     `BNOT : s = ~a;
     `NOT : s = !a;
     default : s = 16'hxxxx;
   endcase

endmodule
```

注１

ビット列は符号なし整数として扱われるので，a と b の大小比較を２の補数として行うために，16'h8000 を加算した x と y を導入．つまり，a と b の最上位ビットを反転したものが x と y になる．符号なし整数 x と y の大小比較は２の補数 a と b の大小比較と一致する．

注２

Verilog2001 の場合，signed が使えるので，x と y を使う必要はない．a,b,s を，singed をつけて（input signed, output signed, reg signed など）宣言する．ただし，シミュレータ，論理合成ツールによっては signed の扱いにバグがあるので注意が必要．（例えば，Icarus Verilog には signed の処理にバグがある）