wireとreg

wire文は信号線の名前を付ける目的で使います。

wire [15:0] a,b;

信号線への出力や接続はassign文を使います

assign a=b;

reg[15:] accum;

レジスタの宣言にはreg文を使います。

reg文はalways文で<=(ブロッキング代入文)を使って、クロックの立ち上がり(立下り)に同期して値を保存します。

always @(posedge clk)

  if(we) accum <= datain; こちらは立ち上がりに同期

always @(negedge clk)

   if(we) accume <= datain; こちらは立下りに同期

wire文とreg文使用上の注意

- wire文もreg文のコード中のどこにでも書けます。verilogにはscopeがないからです。書いた場所より下で有効になります。しかしmodule文の最初の方で宣言することをお勧めします。
- reg文はalways @(posedge)で立ち上がり、always @(negedge)で立ち上がり動作のD-F.F(レジスタ)を生成します。他にもD-ラッチを生成したり、wire文の代わりにも使えるのですが、この授業では使いません。プロ以外は止めた方が無難です。
- reg文で生成したレジスタは、初期状態が決まるようにリセットしておいた方がいいです。これには非同期と同期があります。→ always文を参照してください。
- reg文でメモリも書けます→　mem文を参照してください。