

```
module WB
```

```
//input
```

```
    input [2:0] Wbaddress,  
    input [1:0] control, // 1: RegWrite , 0:MemtoReg  
    input [15:0] ALUresult,  
    input [15:0] LDresult,
```

```
//output
```

```
    output RegWrite_,  
    output [2:0] Wbaddress_,  
    output [15:0] Wbdata_
```

```
////////////////////////////////////////////////////////////////
```

```
// 機能
```

```
    WB フェーズの機能を提供する。 レジスタ書き込みタイミングや書き込み処理は ID に委託。
```

```
////////////////////////////////////////////////////////////////
```

```
//input 詳細
```

```
    Wbaddress      3 bit 書き込み用アドレス  
    control        2bit 制御コード。 WB で使用するの以下の2つ  
                   第 1bit RegWrite   書き込み命令  
                   第 0bit MemtoReg 読み出しデータ(LDresult)を使用するか  
                                   (0: ALUresult 1: LDresult)  
  
    ALUresult      16bit ALU の演算結果  
    LDresult       16bit メモリの読み出し結果
```

```
//output 詳細
```

```
    RegWrite_      1bit レジスタ書き込み制御信号  
    Wbaddress_     3bit レジスタ書き込み用アドレス  
    Wbdata_        1bit レジスタ書き込みデータ
```

```
////////////////////////////////////////////////////////////////
```

```
// コンパイル結果
```

```
Flow Status Successful - Thu Apr 27 13:32:54 2017
```

```
Quartus II 32-bit Version      13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
```

```
Revision Name    WB
```

```
Top-level Entity Name  WB
```

Family Cyclone IV E
Device EP4CE30F23I7
Timing Models Final
Total logic elements 16 / 28,848 (< 1 %)
Total combinational functions 16 / 28,848 (< 1 %)
Dedicated logic registers 0 / 28,848 (0 %)
Total registers 0
Total pins 57 / 329 (17 %)
Total virtual pins 0
Total memory bits 0 / 608,256 (0 %)
Embedded Multiplier 9-bit elements 0 / 132 (0 %)
Total PLLs 0 / 4 (0 %)

```
////////////////////////////////////////  
module WB(  
//    input clock,  
        input [2:0] WBaddress,  
        input [1:0] control, // 1: RegWrite , 0:MemtoReg  
        input [15:0] ALUresult,  
        input [15:0] LDresult,  
  
        output RegWrite_,  
        output [2:0] WBaddress_,  
        output [15:0] WBdata_  
);  
  
    assign RegWrite_ = control[1];  
    assign WBaddress_ = WBaddress;  
    assign WBdata_ = (control[0]==1)? LDresult : ALUresult;  
  
endmodule
```