Code Modification:

1. All the modified code has comment with //Code Modified.
2. Insert the basic operation ADD and OR’s opcode:

ADD = 6'b100000, OR = 6'b100101.

3. IMemory[0] = 32'h8c210003;

IMemory[1] = 32'hac020000;

IMemory[2] = 32'h00642820;

This code stand for:

lw $1 3($1); // IMemory[0] = 32'h8c210003;

sw $2 0($0); // IMemory[1] = 32'hac020000;

Add $3 $2 $5; // IMemory[2] = 32'h00642820;

This is not our target code, so we need to change it to what we want.

add $5,$2,$1 -> 32'b000000|00010|00001|00101|00000|100000

IMemory[0] = 32'h00412820

lw $3,4($5) -> 32'b100011|00101|00011|0000000000000100

IMemory[1] = 32'h8ca30004;

lw $2,0($2) -> 32'b100011|00010|00010|0000000000000000

IMemory[2] = 32'h8c420000;

or $3,$5,$3 -> 32'b000000|00101|00011|00011|00000|100101

IMemory[3] = 32'h00a31825;

sw $3,0($5) -> 32'b101011|00101|00011|000000000000000

IMemory[4] = 32'haca30000;

Then we got the none-nop code: (This is in the mipspipe\_harzard.v)

IMemory[0] = 32'h00412820;//add $5,$2,$1

IMemory[1] = 32'h8ca30004;//lw $3,4($5)

IMemory[2] = 32'h8c420000;//lw $2,0($2)

IMemory[3] = 32'h00a31825;//or $3,$5,$3

IMemory[4] = 32'haca30000;//sw $3,0($5)

4. After we got the code, we can see IMemory[0] and [1] has the data dependency, so add two nops instruction between them, also [1] and [3] has data dependency, since they have [2] between them, so only one nop will be needed. Same between [3] and [4]. After optimized, the code will be look like this:

IMemory[0] = 32'h00412820;//add $5,$2,$1

IMemory[1] = nop;//insert bubble

IMemory[2] = nop;//insert bubble

IMemory[3] = 32'h8ca30004;//lw $3,4($5)

IMemory[4] = 32'h8c420000;//lw $2,0($2)

IMemory[5] = nop;//lw $2,0($2) has taken 1 lie, so only need 1 bubble here

IMemory[6] = 32'h00a31825;//or $3,$5,$3

IMemory[7] = nop;// insert bubble

IMemory[8] = nop;// insert bubble

IMemory[9] = 32'haca30000;//sw $3,0($5)

5. Rewrite the DMemory, since will need to access those addresses:

DMemory[2] = 32'hfffffff0; //0($2):32'hfffffff0

DMemory[5] = 32'hffffffff; //4($5):32'hffffffff

6. We have opcode of add and or, but also need to implement the operation of add and or:

37: EXMEMALUOut <= Ain | Bin; // or operation