```
// File: problem_2_5_2_basic_gates.v
// John Hubbard, 03 Oct 2014
// hw2 assignment

module mux(f, a, b, select);
    input a, b, select;
    output f;
    assign f = (select & a) | ((~select) & b);
endmodule


module problem_2_5_2_basic_gates_bdd(a, b, f0, f1, f2, f3, f4, f5);
    input a, b;
    output f0, f1, f2, f3, f4, f5;
    wire notA;

    mux func_notA(notA, 1'b0, 1'b1, a);

    mux func0_and(f0, b, 1'b0, a);
    mux func1_or(f1, 1'b1, b, a);
    mux func2_xor(f2, notA, a, b);
    mux func3_nand(f3, 1'b0, 1'b1, f0); // f0 == ab
    mux func4_nor(f4, 1'b0, notA, b);
    mux func5_xnor(f5, a, notA, b);

endmodule

module problem_2_5_2_basic_gates_aig(a, b, f0, f1, f2, f3, f4, f5);
    input a, b;
    output f0, f1, f2, f3, f4, f5;
    wire notA, notB;
    wire BnotA, AnotB;
    wire part1, part2;

    not helper1(notA, a);
    not helper2(notB, b);
    and helper3(BnotA, b, notA);
    and helper4(AnotB, a, notB);

    and func0_and(f0, a, b);
    not func1_or(f1, f4); // f4 == not(a + b)

    // xor: ~a.b + a.~b:
    // =  ~ nor(~a.b + a.~b)
    // =  ~ and( ~(~a.b), ~(a.~b))

    not h5(part1, BnotA);
    not h6(part2, AnotB);
    and h7(f5, part1, part2); // f5 is xnor, which is just nor, before inverting
    not h8(f2, f5);           // f2 is nor

    not func3_nand(f3, f0); // f0 == ab
    and func4_nor(f4, notA, notB);
```

```
endmodule
```