hamming.c

```c
#include <ap_int.h>
void hamming(ap_uint<1> a,
  ap_uint<1> b, ap_uint<16> &h){
#pragma HLS INTERFACE ap_none port=a,b,h

  static ap_uint<16> tmp = 0;
  ap_uint<1> c = a^b;
  if(c.get_bit(0))
    tmp++;
  h = tmp;
}
```

hamming.v

```verilog
module hamming(clk, rst, a, b, h);
  input clk, rst, a, b;
  output reg [15:0]h;

  always@(posedge clk or posedge rst)
    if(rst)
      h <= 0;
    else
      h <= h + a^b;
endmodule
```

hamming_netlist.v

```verilog
module hamming(clk, rst, a, b, h);
  input clk, rst, a, b;
  output [15:0]h;
...
DFF \c_reg[0] (.D(n3), .CLK(clk),
 .RST(rst), .Q(h[0]) );
...
XNOR U1 ( .A(a), .B(b), .Z(n2) );
NANDN U3 ( .A(n2), .B(n3), .Z(n3) );
...
endmodule
```