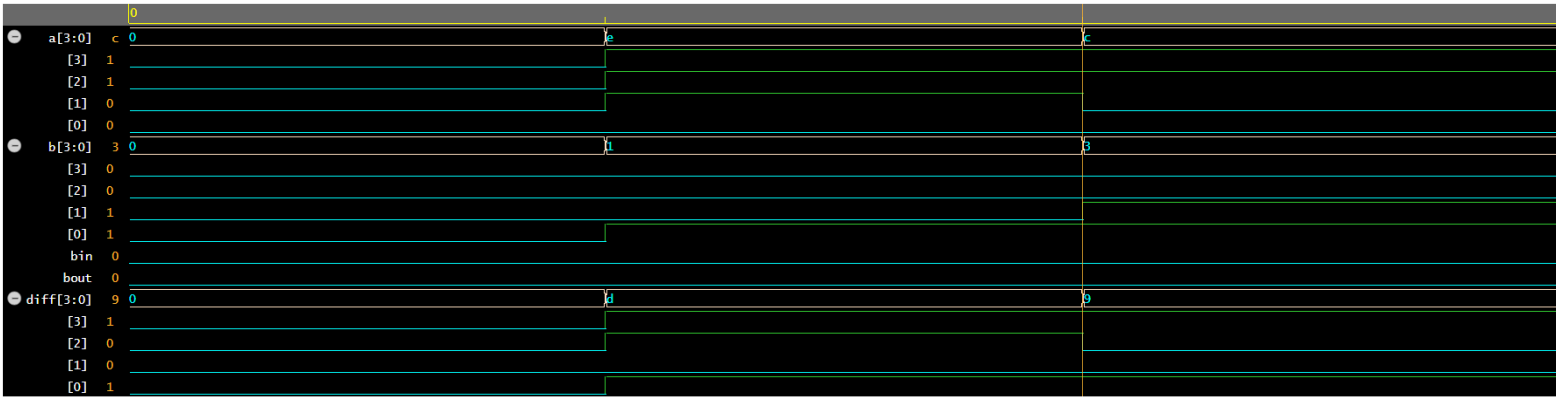
Design:

```
module halfsubstractor(a, b, diff, borrow);
  input a, b;
  output diff, borrow;

  assign diff = a ^ b;
  assign borrow = ~a & b;
endmodule

module fullsubstractor(a, b, c, diff, borrow);
  input a, b, c;
  output diff, borrow;
  wire n1, n2, n3;

  halfsubstractor first(.a(a), .b(b), .diff(n1), .borrow(n2));
  halfsubstractor second(.a(n1), .b(c), .diff(diff), .borrow(n3));
  assign borrow = n3 | n2;
endmodule

module rcs_4bit(a, b, bin, diff, bout);
  input [3:0]a;
  input [3:0]b;
  input bin;
  output [3:0]diff;
  output bout;
  wire n1, n2, n3;

  fullsubstractor first(.a(a[0]), .b(b[0]), .c(bin), .diff(diff[0]),
.borrow(n1));

  fullsubstractor second(.a(a[1]), .b(b[1]), .c(n1), .diff(diff[1]),
.borrow(n2));

  fullsubstractor third(.a(a[2]), .b(b[2]), .c(n2), .diff(diff[2]),
.borrow(n3));

  fullsubstractor fourth(.a(a[3]), .b(b[3]), .c(n3), .diff(diff[3]),
.borrow(bout));

endmodule
```

testbrench:

```
`timescale 1ns/1ns
module rcs_tb();
  reg [3:0]a, b;
  reg bin;
  wire [3:0]diff;
  wire bout;
  wire n1, n2, n3;

  rcs_4bit rcs_test(.a(a), .b(b), .bin(bin), .diff(diff), .bout(bout));
  initial begin
    a = 4'b0000;
    b = 4'b0000;
    bin = 1'b0;
    #1

    a = 4'b1110;
    b = 4'b0001;
    bin = 1'b0;
    #1

    a = 4'b1100;
    b = 4'b0011;
    bin = 1'b0;
    #1

    a = 4'b1100;
    b = 4'b0011;
    bin = 1'b0;

  end
  initial begin
    $dumpfile("rcs.vcd");
    $dumpvars;
  end
endmodule
```

Ep_wave:



Bugs encontrados:

- Llamar de mala manera la función “fullsubstractor” (puse fullsustractor)
- Puntos y comas (;)
- No saber como declarar las variable de bits de la manera “4’b1100” por ejemplo.
- No declarar como “0” el bin desde el la declaracion de variables.