Digital ASIC Design, ECE521, Lab 4

Arijit Sengupta, 001441748

Verilog codes:

G:\adder.v - Notepad++

```verilog
`timescale 1ns / 1ps

module HalfAdder(sum, Cout, a, b);
    input     a,b;
    output    Cout, sum;
    assign    sum  = a ^ b;      // Bitwise XOR            xor(sum, a, b);
    assign    Cout = a & b;      // Bitwise AND            and(Cout, a, b);
endmodule

module FullAdder(sum, Cout, a, b, Cin);
    input         a, b, Cin;
    output        sum, Cout;
    wire          w1, w2, w3;
    HalfAdder    HA1 (w1, w2, a, b);
    HalfAdder    HA2 (sum, w3, Cin, w1);
    assign       Cout = w2 | w3;           // Bitwise OR
endmodule
```

Adder.v

C:\Users\Arijit Sengupta\Desktop\Spring 2020\ECE521 - Digital ASIC Design\Labs\systolic.v - Notepad++

```verilog
`timescale 1ns/10ps

module systolicH(SUMout, Cout, SUMin, a, b);
    input a, b;
    output SUMout, Cout, SUMin;
    wire w1;
    assign w1 = a & b;
    HalfAdder  HA1(SUMout, Cout, SUMin, w1);
endmodule

module systolicF(SUMout, Cout, SUMin, a, b, Cin);
    input a, b, Cin;
    output SUMout, Cout, SUMin;
    wire w1;
    assign w1 = a & b;
    FullAdder  FA1(SUMout, Cout, SUMin, w1, Cin);
endmodule

module systolicMult4x4(p, a, b);
    input  [3:0] a, b;
    output [7:0] p;
    wire  [11:0] c;
    wire  [11:0] s;
    wire         a0b0, a1b0, a2b0, a3b0;

    assign a0b0 = a[0] & b[0];
    assign a1b0 = a[1] & b[0];
    assign a2b0 = a[2] & b[0];
    assign a3b0 = a[3] & b[0];
    assign p[0] = a0b0;

    systolicH S00(s[0], c[0], a1b0, a[0], b[1]);
    systolicF S01(s[1], c[1], a2b0, a[1], b[1], c[0]);
    systolicF S02(s[2], c[2], a3b0, a[2], b[1], c[1]);
    systolicH S03(s[3], c[3], c[2], a[3], b[1]);
    systolicH S04(s[4], c[4], s[1], a[0], b[2]);
    systolicF S05(s[5], c[5], s[2], a[1], b[2], c[4]);
    systolicF S06(s[6], c[6], s[3], a[2], b[2], c[5]);
```

Verilog file | length : 1,479 lines : 54 | Ln : 30 Col : 24 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

```verilog
    systolicF S07(s[7], c[7], c[3], a[3], b[2], c[6]);
    systolicH S08(s[8], c[8], s[5], a[0], b[3]);
    systolicF S09(s[9], c[9], s[6], a[1], b[3], c[8]);
    systolicF S10(s[10], c[10], s[7], a[2], b[3], c[9]);
    systolicF S11(s[11], c[11], c[7], a[3], b[3], c[10]);

    assign p[1] = s[0];
    assign p[2] = s[4];
    assign p[3] = s[8];
    assign p[4] = s[9];
    assign p[5] = s[10];
    assign p[6] = s[11];
    assign p[7] = c[11];
endmodule
```

Verilog file | length : 1,479 lines : 54 | Ln : 30 Col : 24 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

Systolic.v

C:\Users\Arijit Sengupta\Desktop\Spring 2020\ECE521 - Digital ASIC Design\Labs\testbench.v - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

systolic.v | testbench.v | adder.v

```
`timescale 1ns/10ps

module testbench();
    parameter HC =3;
    reg       clk;
    reg [7:0] count;
    wire[3:0] a, b;
    wire[7:0] p;

    assign  b = count[7:4];
    assign  a = count[3:0];

    initial
    begin
        clk <= 1;
        count <= 0;
    end

    always #HC clk <= ~clk;

    always @(posedge clk)
        begin
            count <= count+1;
        end

    systolicMult4x4 sys_mult(.p(p),
                             .a(a),
                             .b(b));
endmodule
```

Verilog file | length : 417 lines : 29 | Ln : 1 Col : 20 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

Testbench.v

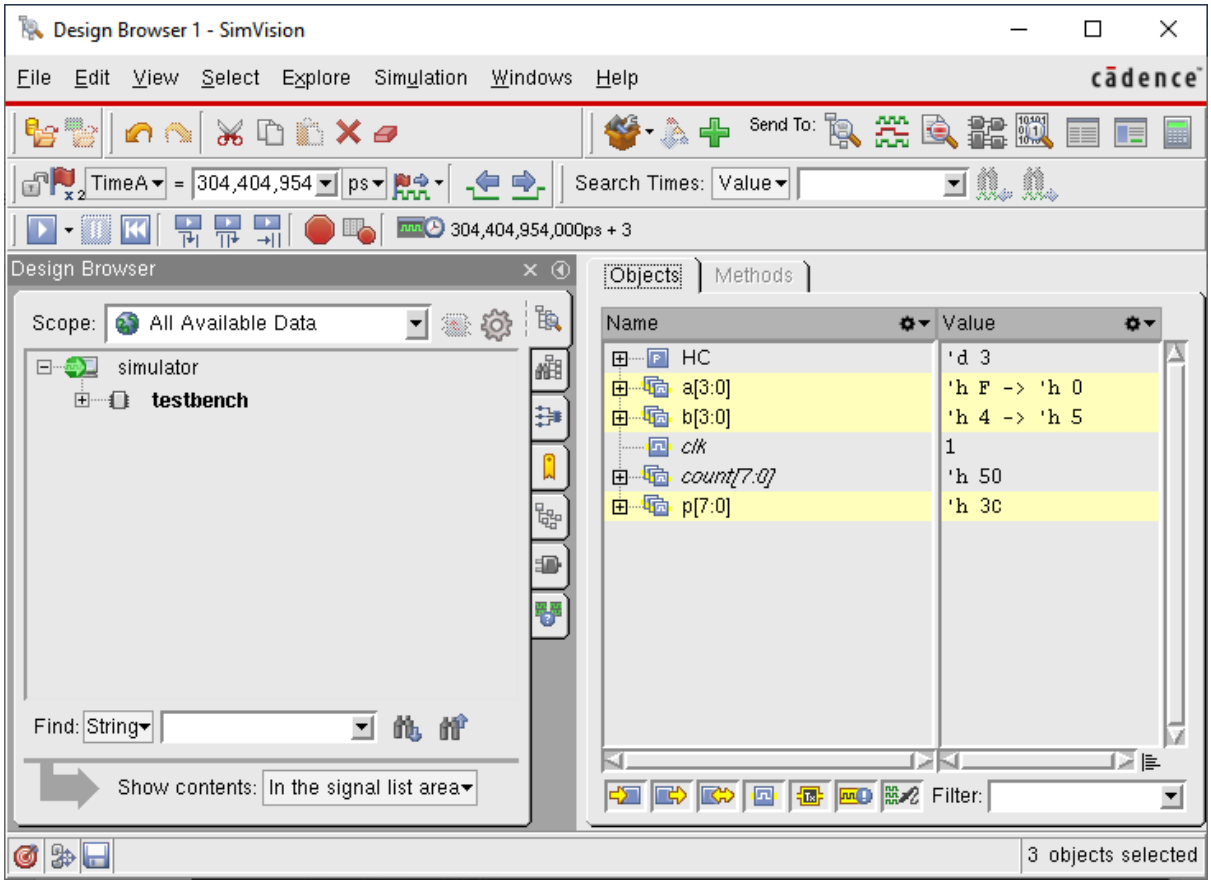
Compilation and Results:

ceashpc-12.rit.albany.edu - PuTTY

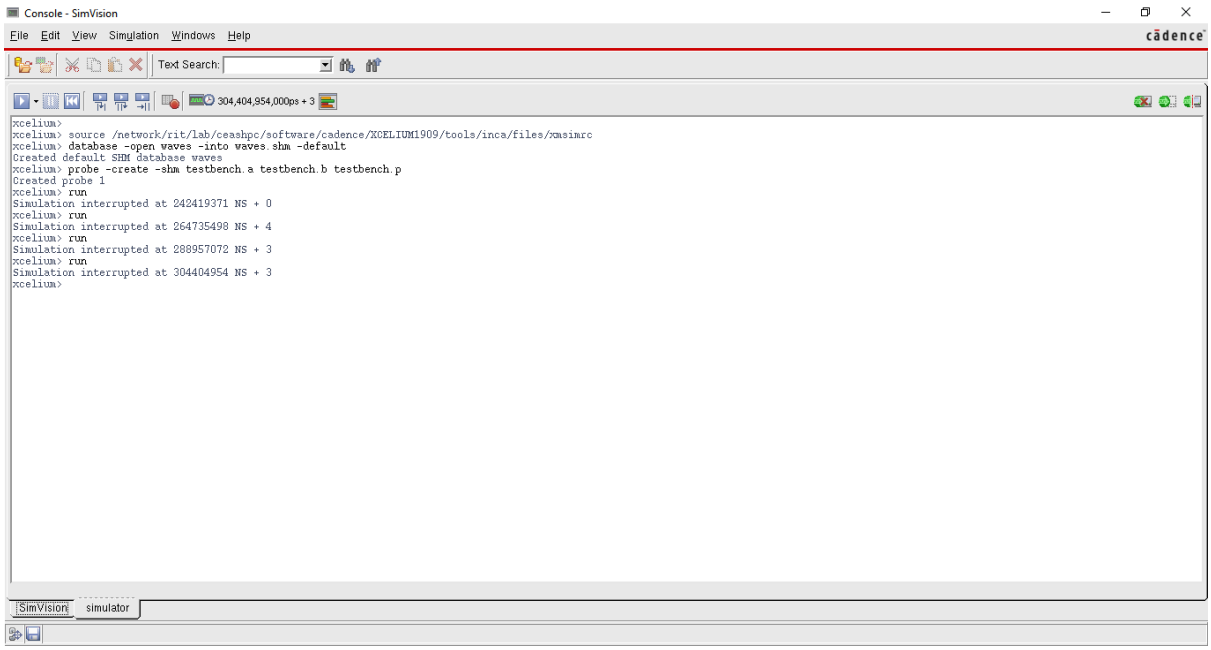
```
-bash-4.2$ xrun -access +r -gui testbench.v adder.v systolic.v &
[1] 43810
-bash-4.2$ TOOL:        xrun(64)        19.09-s001: Started on Mar 09, 2020 at 23:56:55 EDT
xrun(64): 19.09-s001: (c) Copyright 1995-2019 Cadence Design Systems, Inc.
Recompiling... reason: file './systolic.v' is newer than expected.
        expected: Mon Mar  9 23:52:36 2020
        actual:   Mon Mar  9 23:56:49 2020
file: systolic.v
        module worklib.systolicMult4x4:v
                errors: 0, warnings: 0
                Caching library 'worklib' ....... Done
        Elaborating the design hierarchy:
        Top level design units:
                testbench
        Building instance overlay tables: .................... Done
        Generating native compiled code:
                worklib.FullAdder:v <0x2d6aa252>
                        streams:   0, words:     0
                worklib.systolicF:v <0x52ec8aa3>
                        streams:   0, words:     0
                worklib.HalfAdder:v <0x73f264bc>
                        streams:   0, words:     0
                worklib.systolicH:v <0x78340b43>
                        streams:   0, words:     0
                worklib.systolicMult4x4:v <0x22bc75bf>
                        streams:   1, words:   304
                worklib.testbench:v <0x6a7d0124>
                        streams:   6, words:  1581
        Building instance specific data structures.
        Loading native compiled code:     .................... Done
        Design hierarchy summary:
                                Instances  Unique
                Modules:               42       6
                Registers:              2       2
                Scalar wires:          72       -
                Expanded wires:         8       2
                Always blocks:          2       2
                Initial blocks:         1       1
                Cont. assignments:      2      19
                Simulation timescale:  1ps
        Writing initial simulation snapshot: worklib.testbench:v

-bash-4.2$ waiting for SimVision/Indago to connect...
```

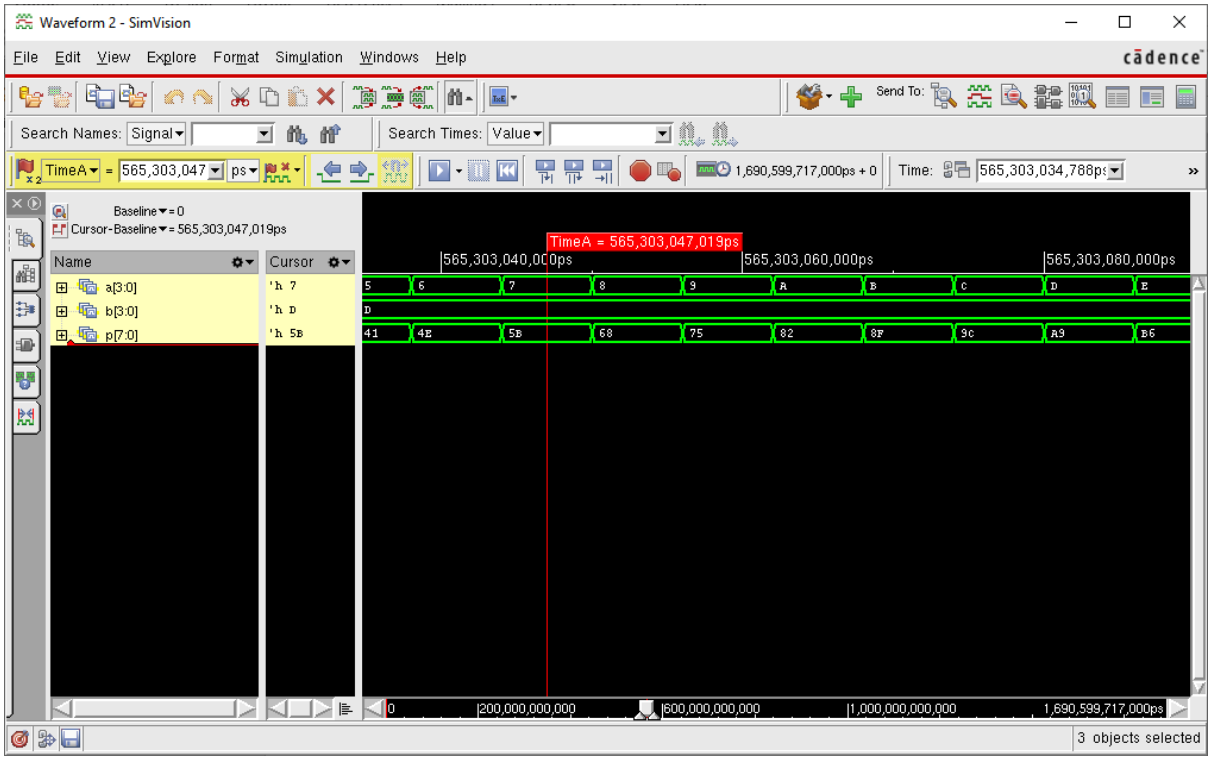
Ssh terminal message



SimVision Design Browser

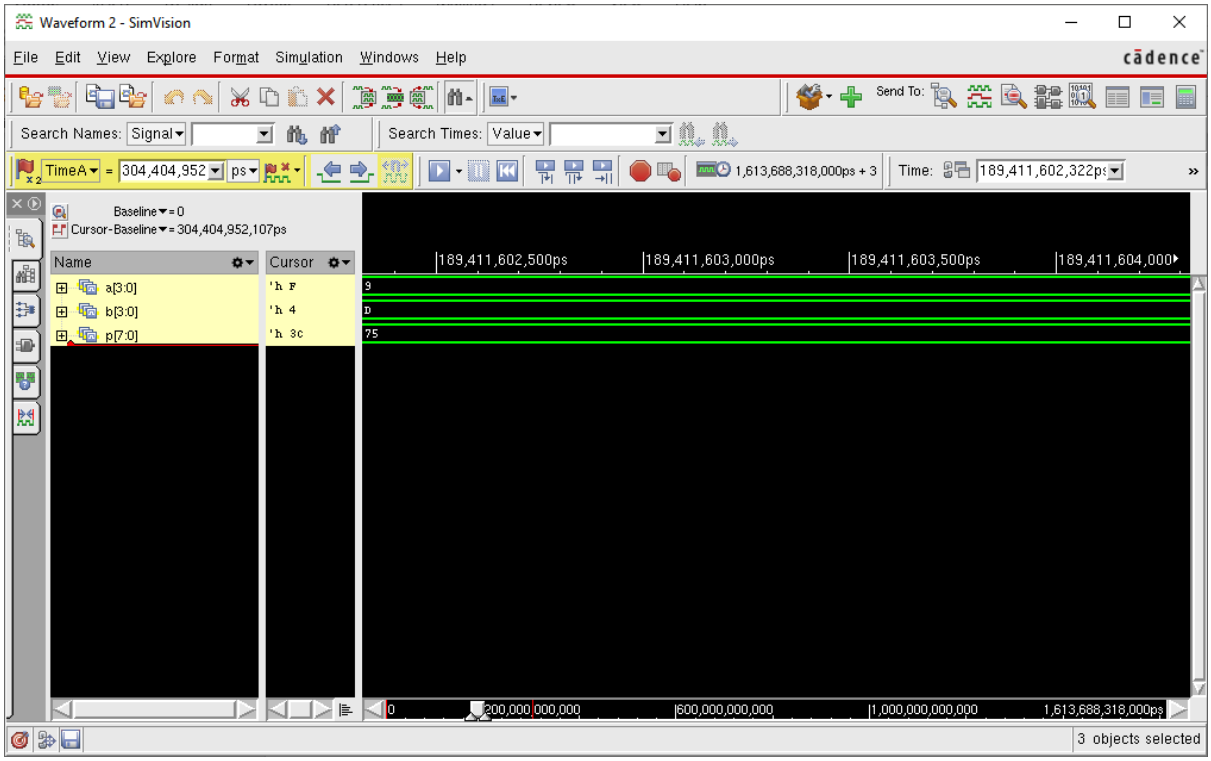


SimVision Console



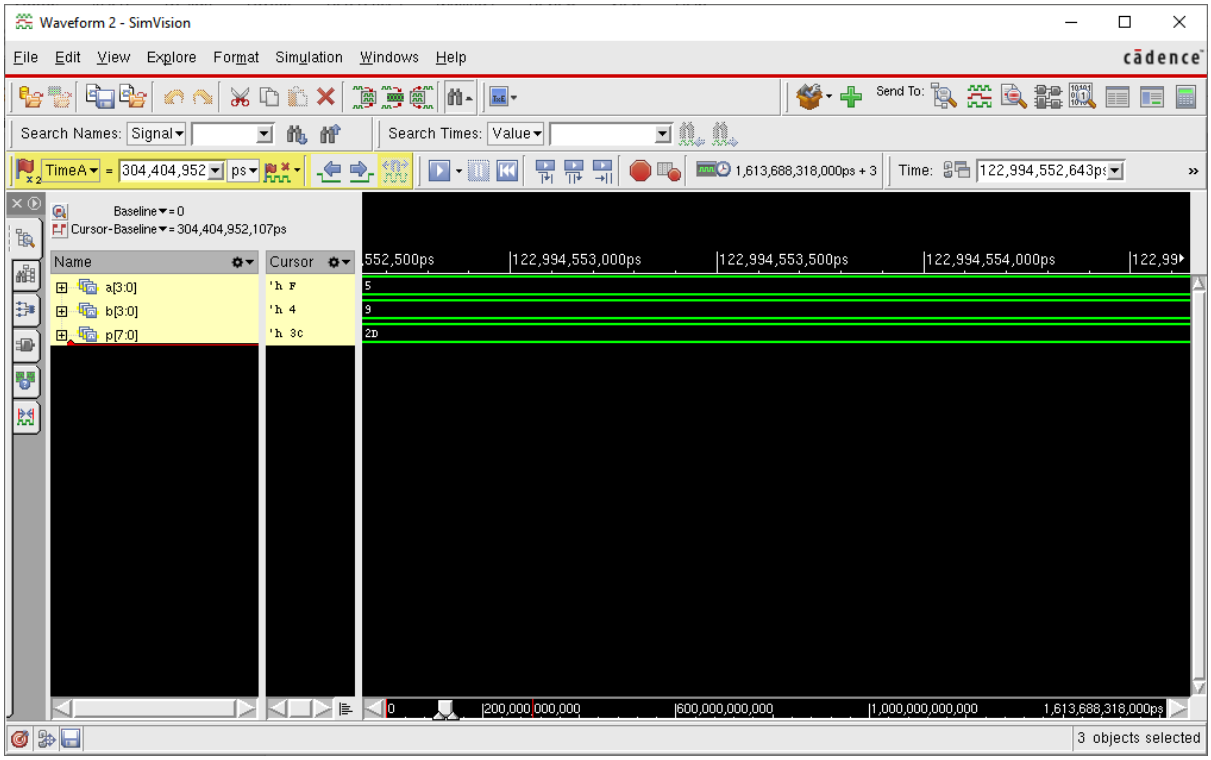
Simulation Waveform

Now let's verify the multiplier output values for some random sets of inputs a and b.

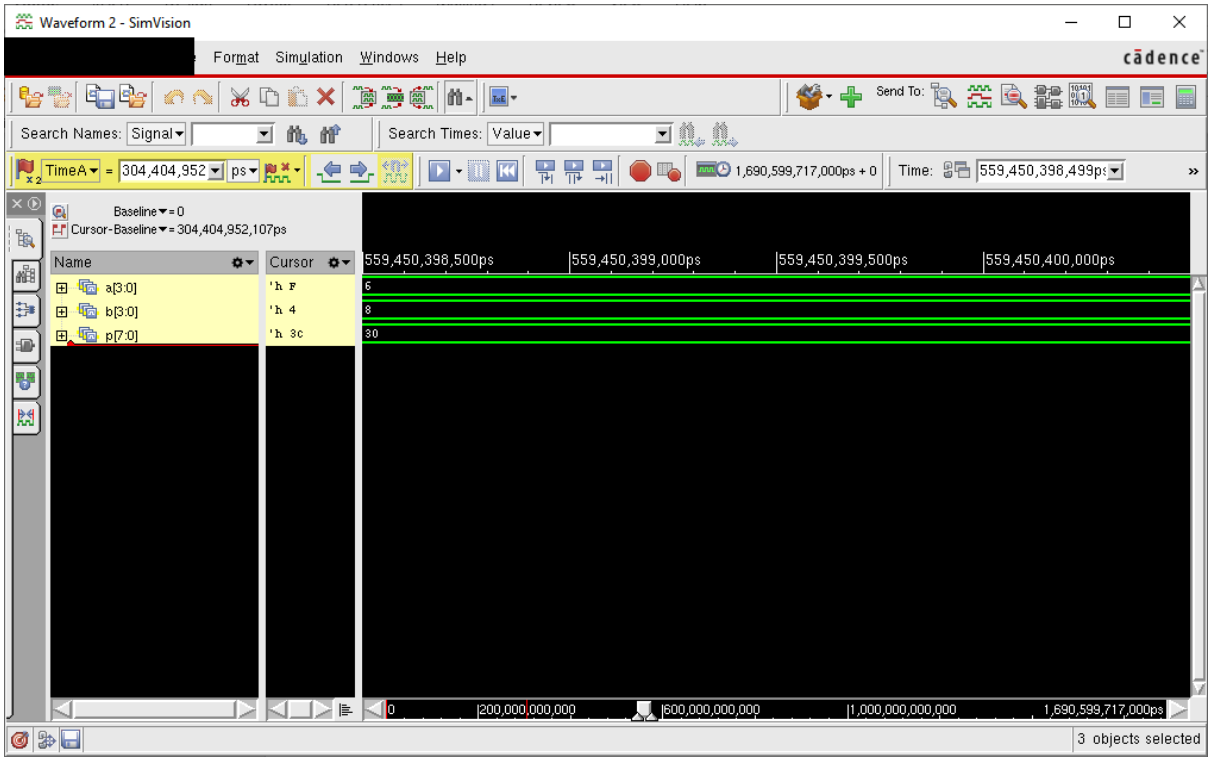


A= 9, B = D (13 in decimal), P= 75 (117 in decimal). Hence the multiplier works perfectly!!

We check the result for few more random values.



A = 5, B = 9, P = 2D (45 in decimal), the value is correct!



A = 6, B = 8, P = 30 (48 in decimal), the value is correct!

Hence the 4x4 systolic multiplier works perfectly as verified from the waveform in Xcelium.