Serial Input

January 29, 2025

Modulo 3 of Serial Input

Modulo 3 of Serial Input Example

Input	Number	Output
		00
01	01	01
10	0110	00
11	011011	00
⋮	⋮	⋮

It **looks** as if infinite states are needed

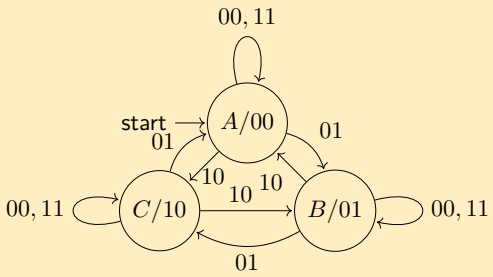
Modulo 3 Remainder Analysis

$$3n \times 4 = 12n$$
$$3n \times 4 + 1 = 12n + 1$$
$$3n \times 4 + 2 = 12n + 2$$
$$3n \times 4 + 3 = 12n + 3$$

$$(3n + 1) \times 4 = 12n + 3 + 1$$
$$(3n + 1) \times 4 + 1 = 12n + 3 + 2$$
$$(3n + 1) \times 4 + 2 = 12n + 6$$
$$(3n + 1) \times 4 + 3 = 12n + 6 + 1$$

$$(3n + 2) \times 4 = 12n + 6 + 2$$
$$(3n + 2) \times 4 + 1 = 12n + 9$$
$$(3n + 2) \times 4 + 2 = 12n + 9 + 1$$
$$(3n + 2) \times 4 + 3 = 12n + 9 + 2$$

Modulo 3 State Diagram



The output is the state code

Modulo 3 State Table

Q_1	Q_0	X_1	X_0	$Q_1(t+1)$	$Q_0(t+1)$	D_1	D_0
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	1
0	0	1	0	1	0	1	0
0	0	1	1	0	0	0	0
0	1	0	0	0	1	0	1
0	1	0	1	1	0	1	0
0	1	1	0	0	0	0	0
0	1	1	1	0	1	0	1
‘1	0	0	0	1	0	1	0
1	0	0	1	0	0	0	0
1	0	1	0	0	1	0	1
1	0	1	1	1	0	1	0

Modulo 3 State Equations

$$D_1 = \bar{Q}_1\bar{Q}_0X_1\bar{X}_0 + \bar{Q}_1Q_0\bar{X}_1X_0 + Q_1\bar{Q}_0\bar{X}_1\bar{X}_0 + Q_1\bar{Q}_0X_1X_0$$
$$D_0 = \bar{Q}_1\bar{Q}_0\bar{X}_1X_0 + \bar{Q}_1Q_0\bar{X}_1\bar{X}_0 + \bar{Q}_1Q_0X_1X_0 + Q_1\bar{Q}_0X_1\bar{X}_0$$

Modulo 3 Module I

Listing 1: demo/10_mod3/mod3.v

```
‘timescale 1ns/1ns

module mod3(
  output m1,m0,
  input  x1,x0,
  input  init,
  input  clk
);
  wire x1n, x0n;
  notGate n1(x1n,x1);
  notGate n0(x0n,x0);

  wire q1, q1n, q0, q0n;
```

Modulo 3 Module II


```
  wire a1, a2, a4, a5, a7, a8, aA, aB;

  and4Gate ag1(a1,q1n,q0n,x1n,x0 );
  and4Gate ag2(a2,q1n,q0n,x1 ,x0n);
  and4Gate ag4(a4,q1n,q0 ,x1n,x0n);
  and4Gate ag5(a5,q1n,q0 ,x1n,x0 );
  and4Gate ag7(a7,q1n,q0 ,x1 ,x0 );
  and4Gate ag8(a8,q1 ,q0n,x1n,x0n);
  and4Gate agA(aA,q1 ,q0n,x1 ,x0n);
  and4Gate agB(aB,q1 ,q0n,x1 ,x0 );

  wire d1,d0;
  or4Gate o1(d1,a2,a5,a8,aB);
  or4Gate o2(d0,a1,a4,a7,aA);
```


Modulo 3 Module III


```
  dffi dff1(q1,q1n,d1,1'0b,init,clk);
  dffi dff0(q0,q0n,d0,1'0b,init,clk);

  assign m1 = q1;
  assign m0 = q0;

endmodule
```


Modulo 3 Testbench I

Listing 2: demo/10_mod3/mod3_tb.v

```
`timescale 1ns/1ns

module mod3_tb;

  wire clk;
  clock #(100) c(clk);

  logic init;
  logic x1, x0;
  wire m1, m0;

  mod3 g(m1, m0, x1, x0, init, clk);
```


Modulo 3 Testbench II


```
  initial begin

    $dumpfile("mod3.vcd");
    $dumpvars;

    x1 = 1;
    x0 = 0;

    init = 1;
    #100ns;
    init = 0;

    #2000ns;
    $finish;
  end
```

Modulo 3 Testbench III

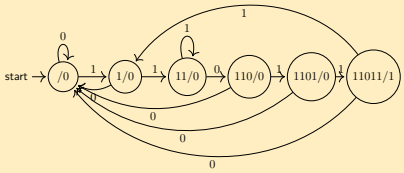
```
always @(posedge clk) begin
    $display(m1,m0);
end
endmodule
```


Identifying disjoint "11011"

disjoint "11011" Serial Input Example

Input	Number	Output
		0
1	1	0
1	11	0
0	110	0
1	1101	0
1	11011	1
0	110110	0
⋮	⋮	⋮

disjoint 11011 State Diagram



State Codes

Name	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
	0	0	0	0	0	1
1	0	0	0	0	1	0
11	0	0	0	1	0	0
110	0	0	1	0	0	0
1101	0	1	0	0	0	0
11011	1	0	0	0	0	0

disjoint "11011" State Table

Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	X	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	1	0	0	0	0	0	0	1
0	0	0	0	0	1	1	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	0	1	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	1	0	0	0	0	1	0

disjoint "11011" FF Equations

$$D_5 = Q_4 X$$
$$D_4 = Q_3 X$$
$$D_3 = Q_2 \bar{X}$$
$$D_2 = Q_1 X + Q_2 X$$
$$D_1 = Q_0 X + Q_5 X$$
$$D_0 = Q_0 \bar{X} + Q_1 \bar{X} + Q_3 \bar{X} + Q_4 \bar{X} + Q_5 \bar{X}$$

Output Equation

$$Z = Q_5$$

disjoint 11011 Module I

Listing 3: demo/20_disjoint/disjoint11011.v

```
‘timescale 1ns/1ns

module disjoint11011(
  output z,
  input  x,
  input  init,
  input  clk
);

  wire q5,q5n,q4,q4n,q3,q3n,q2,q2n,q1,q1n,q0,q0n;
  wire d5,d4,d3,d2,d1,d0;
  dffi dff5(q5,q5n,d5,1'b0,init,clk);
  dffi dff4(q4,q4n,d4,1'b0,init,clk);
```


disjoint 11011 Module II


```
  dffi dff3(q3,q3n,d3,1'b0,init,clk);
  dffi dff2(q2,q2n,d2,1'b0,init,clk);
  dffi dff1(q1,q1n,d1,1'b0,init,clk);
  dffi dff0(q0,q0n,d0,init,1'b0,clk);

  assign z = q5;

  wire xn;
  notGate n(xn,x);

  andGate cd5(d5,q4,x);
  andGate cd4(d4,q3,x);
  andGate cd3(d3,q2,xn);

  wire q1x,q2x;
```

disjoint 11011 Module III

```
  andGate cq1x(q1x,q1,x);
  andGate cq2x(q2x,q2,x);
  orGate cd2(d2,q1x,q2x);

  wire q0x,q5x;
  andGate cq0x(q0x,q0,x);
  andGate cq5x(q5x,q5,x);
  orGate cd1(d1,q0x,q5x);

  andGate cd0(d0,q2n,xn);

endmodule
```


Disjoint 11011 Testbench I

Listing 4: demo/02_disjoint11011/disjoint11011_tb.v

```
`timescale 1ns/1ns

module disjoint11011_tb;

  wire clk;
  clock #(100) c(clk);

  logic init;
  logic x;
  wire z;

  disjoint11011 g(z, x, init, clk);
```


Disjoint 11011 Testbench II

```
  initial begin

    $dumpfile("disjoint11011.vcd");
    $dumpvars;

    init = 1;
    #100ns;
    init = 0;
    x = 1;

    #2000ns;
    $finish;
  end

always @(posedge clk) begin
```


Disjoint 11011 Testbench III

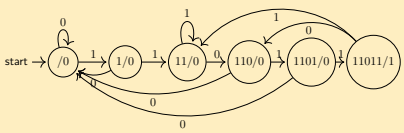
```
  if ($urandom_range(0,5) == 0)
    x = 0;
  else
    x = 1;
end
endmodule
```

Identifying overlap "11011"

overlap "11011" Serial Input Example

Input	Number	Output
		0
1	1	0
1	11	0
0	110	0
1	1101	0
1	11011	1
0	110110	0
1	1101101	0
1	11011011	1
⋮	⋮	⋮

overlap 11011 State Diagram



State Codes

Name	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
	0	0	0	0	0	1
1	0	0	0	0	1	0
11	0	0	0	1	0	0
110	0	0	1	0	0	0
1101	0	1	0	0	0	0
11011	1	0	0	0	0	0

overlap "11011" State Table

Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	X	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	1	0	0	0	0	0	0	1
0	0	0	0	0	1	1	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	0	1	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	1	0	0

overlap "11011" FF Equations

$$D_5 = Q_4 X$$
$$D_4 = Q_3 X$$
$$D_3 = Q_2 \bar{X} + Q_5 \bar{X}$$
$$D_2 = Q_1 X + Q_2 X + Q_5 X$$
$$D_1 = Q_0 X$$
$$D_0 = Q_0 \bar{X} + Q_1 \bar{X} + Q_3 \bar{X} + Q_4 \bar{X}$$

Output Equation

$$Z = Q_5$$

overlap 11011 I

Listing 5: demo/03_overlap11011/overlap11011.v

```
`timescale 1ns/1ns

module overlap11011(
  output z,
  input  x,
  input  init,
  input  clk
);

  wire q5,q5n,q4,q4n,q3,q3n,q2,q2n,q1,q1n,q0,q0n;
  wire d5,d4,d3,d2,d1,d0;
  dffi dff5(q5,q5n,d5,1'b0,init,clk);
  dffi dff4(q4,q4n,d4,1'b0,init,clk);
```


overlap 11011 II

```
  dffi dff3(q3,q3n,d3,1'b0,init,clk);
  dffi dff2(q2,q2n,d2,1'b0,init,clk);
  dffi dff1(q1,q1n,d1,1'b0,init,clk);
  dffi dff0(q0,q0n,d0,init,1'b0,clk);

  assign z = q5;

  wire xn;
  notGate n(xn,x);

  wire q0x,q1x,q2x,q3x,q4x,q5x;
  andGate cq0x(q0x,q0,x);
  andGate cq1x(q1x,q1,x);
  andGate cq2x(q2x,q2,x);
  andGate cq3x(q3x,q3,x);
```


overlap 11011 III

```
  andGate cq4x(q4x,q4,x);
  andGate cq5x(q5x,q5,x);

  wire q0xn,q1xn,q2xn,q3xn,q4xn,q5xn;
  andGate cq0xn(q0xn,q0,xn);
  andGate cq1xn(q1xn,q1,xn);
  andGate cq2xn(q2xn,q2,xn);
  andGate cq3xn(q3xn,q3,xn);
  andGate cq4xn(q4xn,q4,xn);
  andGate cq5xn(q5xn,q5,xn);


  assign d5=q4x;
  assign d4=q3x;
  orGate cd3(d3,q2xn,q5xn);
```

overlap 11011 IV

```
  or3Gate cd2(d2,q1x,q2x,q5x);

  assign d1=q0x;

  or4Gate cd0(d0,q0xn,q1xn,q3xn,q4xn);

endmodule
```


overlap 11011 Testbench I

Listing 6: demo/03_overlap11011/overlap11011_tb.v

```
‘timescale 1ns/1ns

module overlap11011_tb;

  wire clk;
  clock #(100) c(clk);

  logic init;
  logic x;
  wire z;

  overlap11011 g(z, x, init, clk);
```


overlap 11011 Testbench II

```
  initial begin

    $dumpfile("overlap11011.vcd");
    $dumpvars;

    init = 1;
    #100ns;
    init = 0;
    x = 1;

    #2000ns;
    $finish;
  end

always @(posedge clk) begin
```

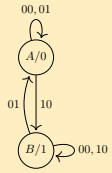

overlap 11011 Testbench III

```
  if ($urandom_range(0,5) == 0)
    x = 0;
  else
    x = 1;
end
endmodule
```

Example: SR-FF from D-FF

Thus: D-FF is equivalent to SR-FF

State Diagram and Table



State	S	R	Next State
A	0	0	A
A	0	1	A
A	1	0	B
A	1	1	X
B	0	0	B
B	0	1	A
B	1	0	B
B	1	1	X

State Code/Table/Equation

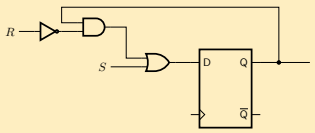
State	Code
A	0
B	1

Q	S	R	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

$$Q(t+1) = S + \bar{R}Q$$

SR-FF from D-FF, FF Input Equation and Circuit

$$D = S + \bar{R}Q$$



Circuit output is the state

(Carmi) Lecture 19(?!) reached here