Counters

© Carmi Merimovich

January 29, 2025

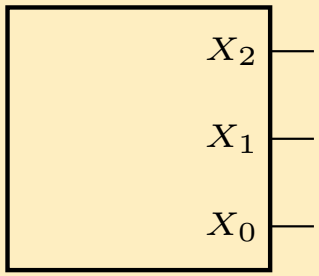
(Carmi) BALAGAN Lecture 17 begins here

Binary Counters

Binary Counter

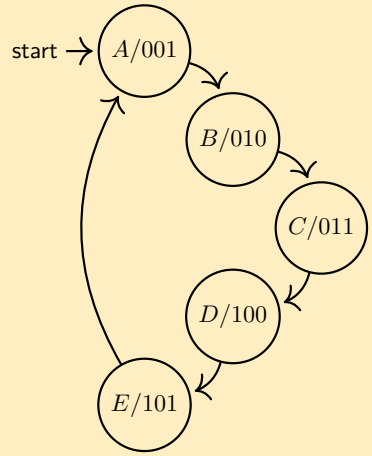
1. Naive with D-FF
2. Naive with SR-FF
3. Naive 1-hot D-FF
4. Better states code DFF
5. No nonsense D-FF

Binay Counter



Binary Counter

State Diagram



State Table

state	next state
A	B
B	C
C	D
D	E
E	A

Binary Counter v1, State Codes

State	Code		
	Q_2	Q_1	Q_0
A	0	0	0
B	0	0	1
C	0	1	0
D	0	1	1
E	1	0	0

Binary Counter v1

State Table

state			next state			Output		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	X_2	X_1	X_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	0	0	0	1	0	1

Binary Counter v1 (Equations)

D-FF input equations

$$D_2 = Q_1 Q_0$$
$$D_1 = Q_1 \oplus Q_0$$
$$D_0 = \bar{Q}_2 \bar{Q}_0$$

Output equations

$$X_2 = Q_2 + Q_1 Q_0$$
$$X_1 = Q_1 \oplus Q_0$$
$$X_0 = \bar{Q}_0$$

Binary Counter v1, Verilog I

Listing 1: demo/01_counter/counter.v

```
`timescale 1ns/1ns

module counter(
  output x2,x1,x0,
  input init,
  input clk
);
  wire q2,q2n,q1,q1n,q0,q0n;
  wire d2,d1,d0;

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

Binary Counter v1, Verilog II

```
  andGate cd2(d2,q1 ,q0 );
  xorGate cd1(d1,q1 ,q0 );
  andGate cd0(d0,q2n,q0n);

  orGate cx2(x2,q2,d2);
  assign x1=d1;
  assign x0=q0n;

endmodule
```

Binary Counter v1 I

Listing 2: demo/01_counter/counter_tb.v

```
`timescale 1ns/1ns

module counter_tb;

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

  logic init;
  wire x2, x1, x0;
  counter c5(x2, x1, x0, init, clk);

  initial begin
```

Binary Counter v1 II

```
    $dumpfile("counter.vcd");
    $dumpvars;

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

    #2000ns;
    $finish;
  end

endmodule
```

The Same Binary Counter Implemented with SR-FF

Binary Counter v2, SR-FF

State Table

state			next state			SR-FF					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	0	0	X	1	0	0	1
0	1	0	0	1	1	0	X	X	0	1	0
0	1	1	1	0	0	1	0	0	1	0	1
1	0	0	0	0	0	0	1	0	X	0	X

Binary Counter v2

SR-FF input equations

$$S_2 = Q_1 Q_0$$
$$R_2 = Q_2$$
$$S_1 = Q_1 \oplus Q_0$$
$$R_1 = Q_1 Q_0$$
$$S_0 = \bar{Q}_2 \bar{Q}_0$$
$$R_0 = \bar{Q}_2 Q_0$$

Output equations (Same as with D-FF)

$$X_2 = Q_2 + Q_1 Q_0$$
$$X_1 = Q_1 \oplus Q_0$$
$$X_0 = \bar{Q}_0$$

Binary Counter v2, Verilog I

Listing 3: demo/02_counter/counter.v

```
`timescale 1ns/1ns

module counter(
  output x2,x1,x0,
  input init,
  input clk
);
  wire q2,q2n,q1,q1n,q0,q0n;
  wire s2,r2,s1,r1,s0,r0;

  srffi srff0(q2,q2n,s2,r2,1'b0,init,clk);
  srffi srff1(q1,q1n,s1,r1,1'b0,init,clk);
  srffi srff2(q0,q0n,s0,r0,1'b0,init,clk);
```

Binary Counter v2, Verilog II

```
  andGate cs2(s2,q1 ,q0 );
  assign r2=q2;
  xorGate cs1(s1,q1 ,q0 );
  andGate cr1(r1,q1 ,q0 );
  norGate cs0(s0,q2 ,q0 );
  andGate cr0(r0,q2n,q0);


  orGate cx2(x2,q2,s2);
  assign x1=s1;
  assign x0=q0n;

endmodule
```

Binary Counter v2 I

Listing 4: demo/02_counter/counter_tb.v

```
`timescale 1ns/1ns

module counter_tb;

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

  logic init;
  wire x2, x1, x0;
  counter c5(x2, x1, x0, init, clk);

  initial begin
```

Binary Counter v2 II

```
    $dumpfile("counter.vcd");
    $dumpvars;

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

    #2000ns;
    $finish;
  end

endmodule
```

Same Counter, 1-Hot with D-FF

Binary Counter v3, State Codes

State	Code				
	Q_4	Q_3	Q_2	Q_1	Q_0
A	0	0	0	0	1
B	0	0	0	1	0
C	0	0	1	0	0
D	0	1	0	0	0
E	1	0	0	0	0

Binary Counter v3

State Table

state					next state				
Q_4	Q_3	Q_2	Q_1	Q_0	Q_4	Q_3	Q_2	Q_1	Q_0
0	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0	0	0
1	0	0	0	1	0	0	0	0	1

Binary Counter v3

Output functions

state					Output		
Q_4	Q_3	Q_2	Q_1	Q_0	X_2	X_1	X_0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	0
1	0	0	0	0	1	0	1

Binary Counter v3

D-FF input equations

$$D_4 = Q_3$$
$$D_3 = Q_2$$
$$D_2 = Q_1$$
$$D_1 = Q_0$$
$$D_0 = Q_4$$

Output equations

$$X_2 = Q_3 + Q_4$$
$$X_1 = Q_1 + Q_2$$
$$X_0 = Q_0 + Q_2 + Q_4$$

Binary Counter v3, Verilog I

Listing 5: demo/03_counter/counter.v

```
`timescale 1ns/1ns

module counter(
  output x2,x1,x0,
  input init,
  input clk
);
  wire q4,q4n,q3,q3n,q2,q2n,q1,q1n,q0,q0n;
  wire d4,d3,d2,d1,d0;

  dffi dff4(q4,q4n,d4,1'b0,init,clk);
  dffi dff3(q3,q3n,d3,1'b0,init,clk);
  dffi dff2(q2,q2n,d2,1'b0,init,clk);
```

Binary Counter v3, Verilog II

```
  dffi dff1(q1,q1n,d1,1'b0,init,clk);
  dffi dff0(q0,q0n,d0,init,1'b0,clk);

  assign d4=q3;
  assign d3=q2;
  assign d2=q1;
  assign d1=q0;
  assign d0=q4;

  orGate  cx2(x2,q3,q4);
  orGate  cx1(x1,q1,q2);
  or3Gate cx0(x0,q0,q2,q4);

endmodule
```

Binary Counter v3 I

Listing 6: demo/03_counter/counter_tb.v

```
`timescale 1ns/1ns

module counter_tb;

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

  logic init;
  wire x2, x1, x0;
  counter c5(x2, x1, x0, init, clk);

  initial begin
```

Binary Counter v3 II

```
    $dumpfile("counter.vcd");
    $dumpvars;

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

    #2000ns;
    $finish;
  end

endmodule
```

Same Counter with D-FF, Better Code

Binary Counter v4, States Codes

State	Code		
	Q_2	Q_1	Q_0
A	0	0	1
B	0	1	0
C	0	1	1
D	1	0	0
E	1	0	1

Binary Counter v4

State Table

state			next state			Output		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	X_2	X_1	X_0
0	0	1	0	1	0	0	0	1
0	1	0	0	1	1	0	1	0
0	1	1	1	0	0	0	1	1
1	0	0	1	0	1	1	0	0
1	0	1	0	0	1	1	0	1

Next State and Output are Equal

Binary Counter v4

D-FF input equations

$$D_2 = \overline{Q_1 \oplus Q_0}$$
$$D_1 = Q_2 \downarrow Q_1 + Q_2 \downarrow Q_0$$
$$D_0 = Q_2 + Q_2 \downarrow Q_0$$

Output equations

$$X_2 = Q_2$$
$$X_1 = Q_1$$
$$X_0 = Q_0$$

Binary Counter v4, Verilog I

Listing 7: demo/04_counter/counter.v

```
`timescale 1ns/1ns

module counter(
  output x2,x1,x0,
  input init,
  input clk
);
  wire q2,q2n,q1,q1n,q0,q0n;
  wire d2,d1,d0;

  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);
```

Binary Counter v4, Verilog II

```
  xnorGate cd2(d2,q1 ,q0 );

  wire q2norq1,q2norq0;
  norGate cq2q1(q2norq1,q2,q1);
  norGate cq2q0(q2norq0,q2,q0);
  orGate cd1(d1,q2norq1,q2norq0);

  orGate cd0(d0,q2,q2norq0);

  assign x2=q2;
  assign x1=q1;
  assign x0=q0;

endmodule
```

Binary Counter v4, Verilog III

Binary Counter v4 I

Listing 8: demo/04_counter/counter_tb.v

```
`timescale 1ns/1ns

module counter_tb;

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

  logic init;
  wire x2, x1, x0;
  counter c5(x2, x1, x0, init, clk);

  initial begin
```

Binary Counter v4 II

```
    $dumpfile("counter.vcd");
    $dumpvars;

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

    #2000ns;
    $finish;
  end

endmodule
```

Same Counter, binaryAdder and Comparator

Binary Counter v5, Verilog I

Listing 9: demo/05_counter/counter.v

```
`timescale 1ns/1ns

module counter(
  output x2,x1,x0,
  input init,
  input clk
);
  wire [2:0] q;
  wire [2:0] d;

  dffi dff2(q[2],,d[2],1'b0,init,clk);
  dffi dff1(q[1],,d[1],1'b0,init,clk);
  dffi dff0(q[0],,d[0],init,1'b0,clk);
```

Binary Counter v5, Verilog II

```
wire [2:0] e[2];
assign e[1] = 3'b001;
binaryAdder #(3) ba(,,e[0],q,3'b001,1'b0);

wire eq;
comparator #(3) com(,eq,,q,3'b101);

muxMulti #(1,3) mm(d,e,eq);

assign x2=q[2];
assign x1=q[1];
assign x0=q[0];
```

Binary Counter v5, Verilog III

```
endmodule
```

Binary Counter v5 I

Listing 10: demo/05_counter/counter_tb.v

```
`timescale 1ns/1ns

module counter_tb;

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

  logic init;
  wire x2, x1, x0;
  counter c5(x2, x1, x0, init, clk);

  initial begin
```

Binary Counter v5 II

```
    $dumpfile("counter.vcd");
    $dumpvars;

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

    #2000ns;
    $finish;
  end

endmodule
```

(Carmi) BALAGAN Lecture 17 reached here

Gray Counters

Gray Code

1-Bit difference between successive lines

1-bit	2-bits	3-bits
0		
1		

Gray Code

1-Bit difference between successive lines

1-bit	2-bits	3-bits
0	0	
1	1	

Gray Code

1-Bit difference between successive lines

1-bit	2-bits	3-bits
0	0	
1	1	
	1	
	0	

Gray Code

1-Bit difference between successive lines

1-bit	2-bits	3-bits
0	00	
1	01	
	1	
	0	

Gray Code

1-Bit difference between successive lines

1-bit	2-bits	3-bits
0	00	
1	01	
	11	
	10	

Gray Code

1-Bit difference between successive lines

1-bit	2-bits	3-bits
0	00	00
1	01	01
	11	11
	10	10

Gray Code

1-Bit difference between successive lines

1-bit	2-bits	3-bits
0	00	00
1	01	01
	11	11
	10	10
		10
		11
		01
		00

Gray Code

1-Bit difference between successive lines

1-bit	2-bits	3-bits
0	00	000
1	01	001
	11	011
	10	010
		10
		11
		01
		00

Gray Code

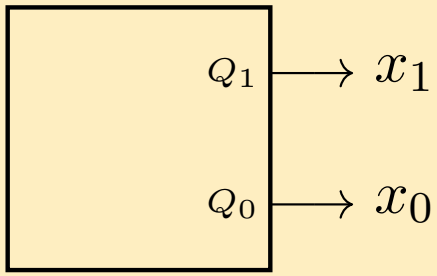
1-Bit difference between successive lines

1-bit	2-bits	3-bits
0	00	000
1	01	001
	11	011
	10	010
		110
		111
		101
		100

2-Bit Gray Counter

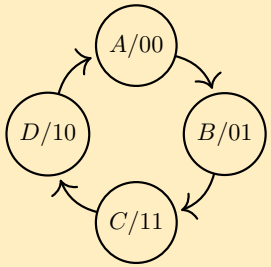
2-Bit Gray Counter

- No input
- 2-Bits output (the state code)



2-Bit Gray Counter

State Diagram



State Table

state	next state
A	B
B	C
C	D
D	A

In this case, we use the output for state code

State	Code
A	00
B	01
C	11
D	10

State Table

State		Next State	
Q_1	Q_0	Q_1	Q_0
0	0	0	1
0	1	1	1
1	1	1	0
1	0	0	0

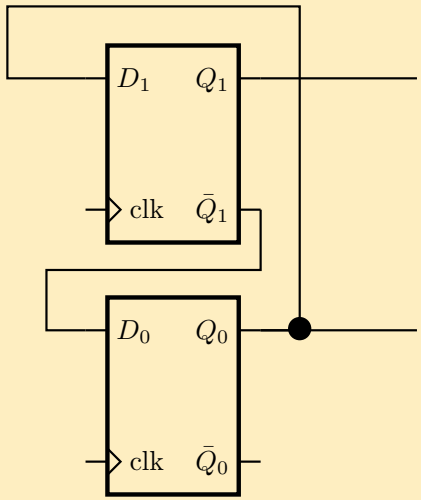
FF input equations

$$D_1 = Q_0$$

$$D_0 = \bar{Q}_1$$

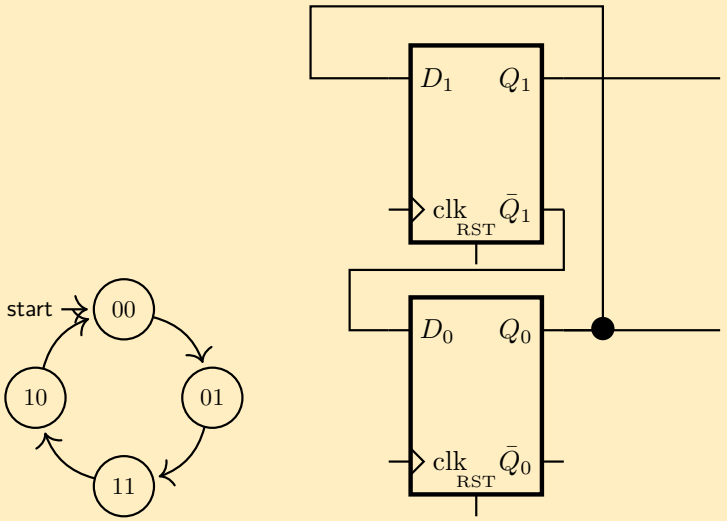
2-Bit Gray Counter

Circuit



Initial state?

2-Bit Gray Counter with Initialization



If not told otherwise

- The FFs can be with or withour sets or resets
- Both pins can be either active low or active high

2-Bits Gray Counter I

Listing 11: demo/065_gray2/gray2.v

```
‘timescale 1ns/1ns

module gray2(
  output x1,x0,
  input init,
  input clk
);

  wire q0,q0n,q1,q1n;

  dffi d1(q1,q1n,q0 ,1’b0,init,clk);
  dffi d0(q0,q0n,q1n,1’b0,init,clk);
```

2-Bits Gray Counter II

```
  assign x1 = q1;
  assign x0 = q0;

endmodule
```

2-Bits Gray Counter I

Listing 12: demo/06_gray2/gray2_tb.v

```
`timescale 1ns/1ns

module gray2_tb;
  logic d;
  wire q, qn;

  wire clk;

  clock #(100) c(clk);

  logic init;
  wire x1, x0;
  gray2 g(x1, x0, init, clk);
```

2-Bits Gray Counter II

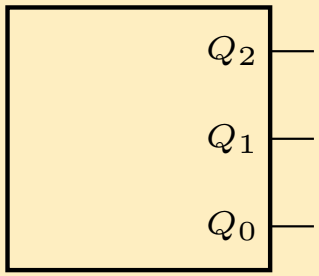
```
  initial begin

    $dumpfile("gray2.vcd");
    $dumpvars;

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

    #2000ns;
    $finish;
  end

endmodule
```

2-Bits Gray Counter III

3-Bit Gray Counter DFF

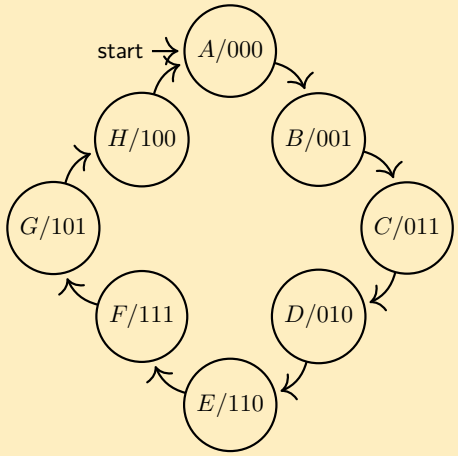
(using D-FFs)

3-Bit Gray Counter



3-Bit Gray Counter

State Diagram



State Table

state	next state
A	B
B	C
C	D
D	E
E	F
F	G
G	H

State Codes

Once again, the output is the state code

State	Code		
	Q_2	Q_1	Q_0
A	0	0	0
B	0	0	1
C	0	1	0
D	0	1	1
E	1	0	0
F	1	0	1
G	1	1	0
H	1	1	1

3-Bit Gray Counter

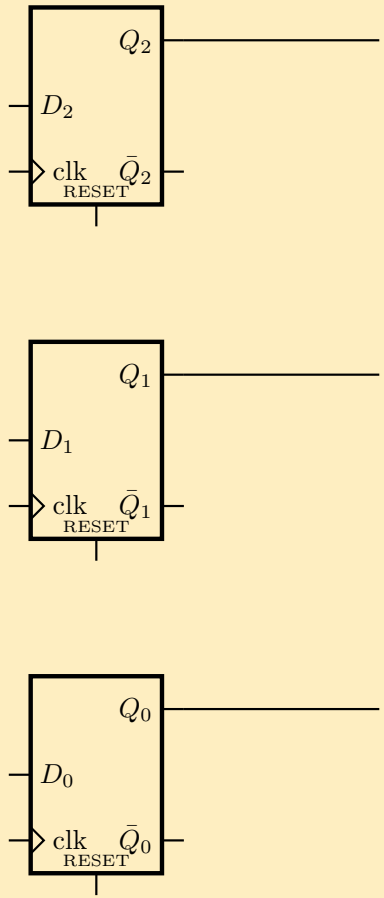
State Table

state			next state		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	1

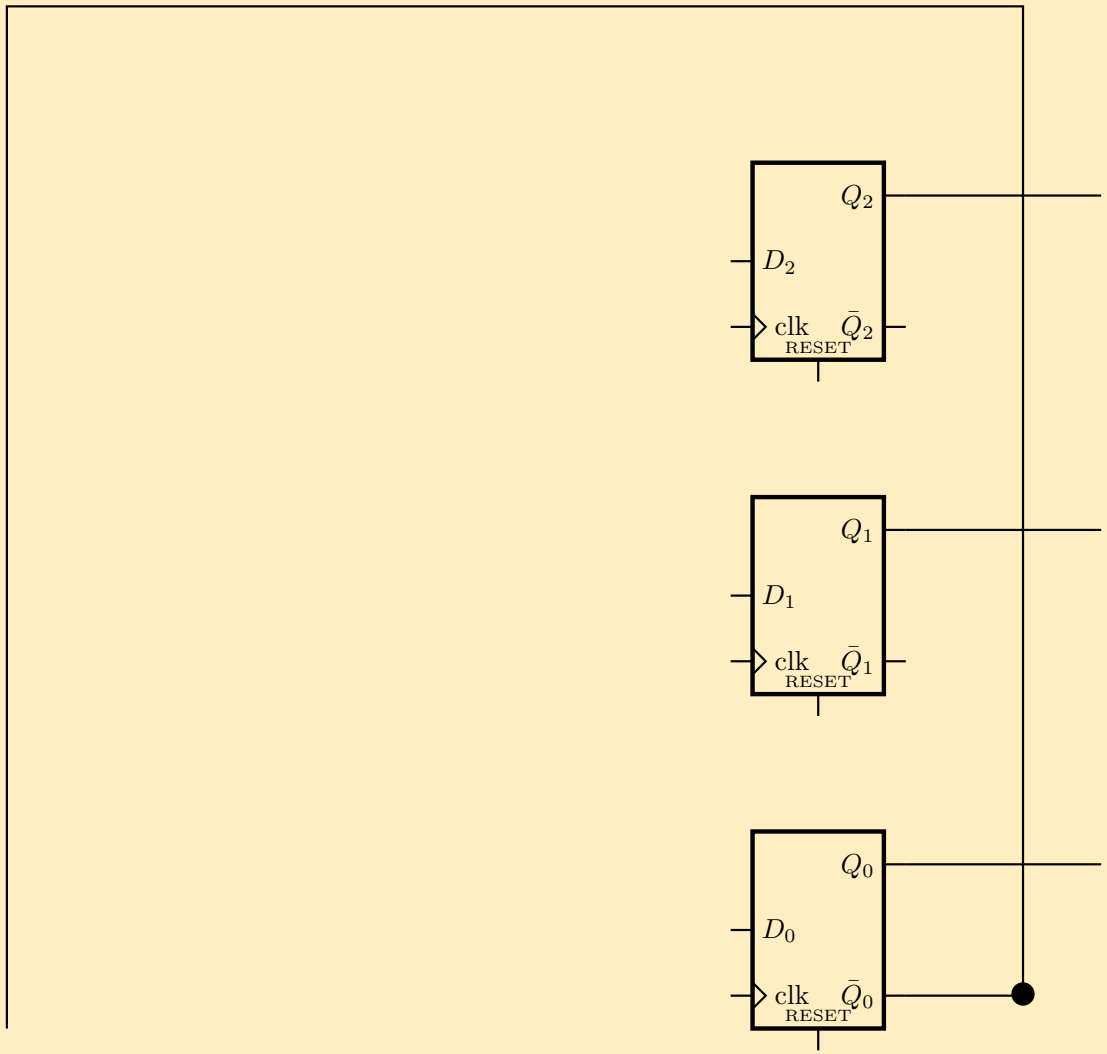
3-Bit Gray Counter (D-FF Input Equations)

$$D_2 = Q_1\bar{Q}_0 + Q_2Q_0 + Q_2Q_1$$
$$D_1 = \bar{Q}_2Q_0 + \bar{Q}_2Q_1 + Q_1\bar{Q}_0$$
$$D_0 = \bar{Q}_2\bar{Q}_1 + Q_2Q_1$$

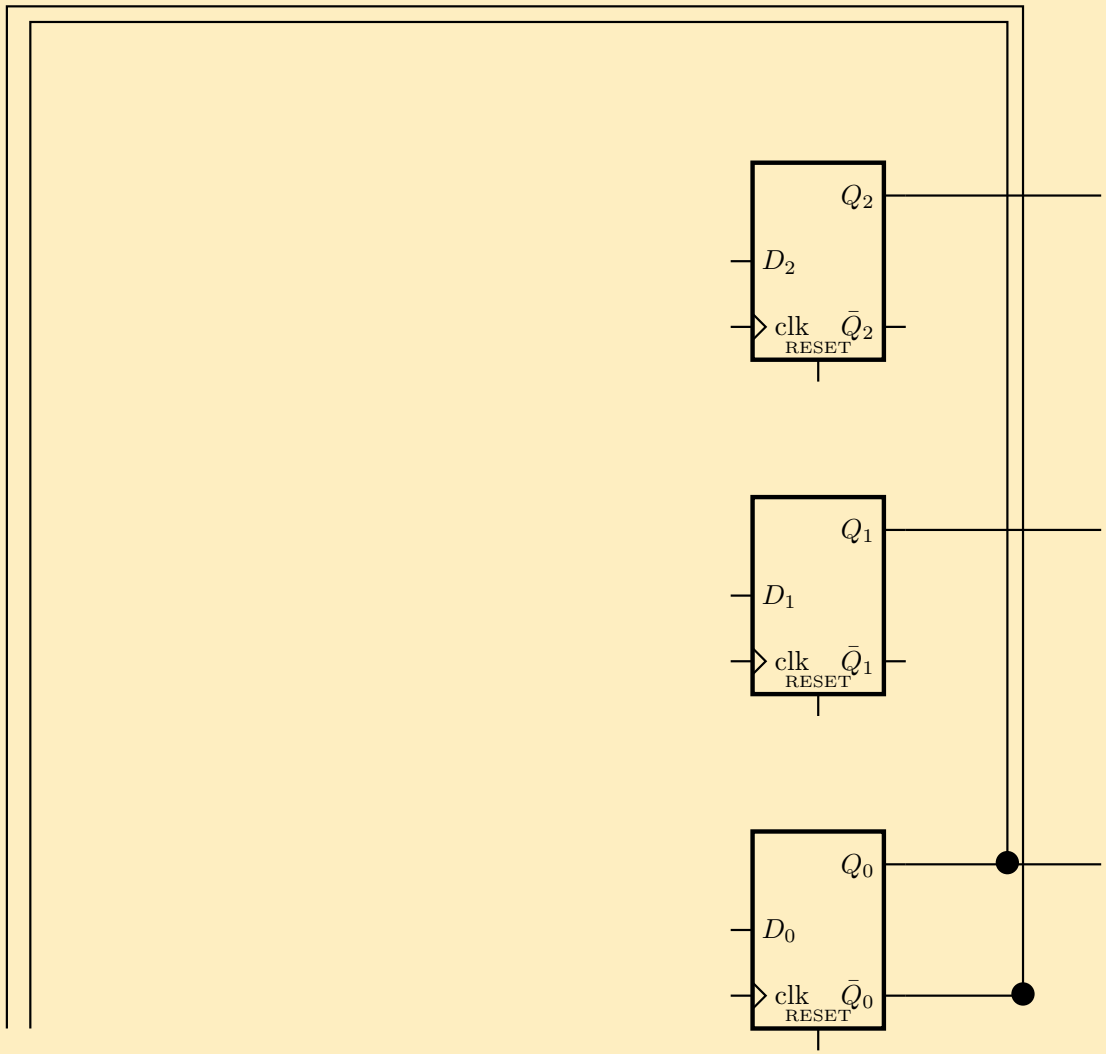
3-Bit Gray Counter (D-FF Circuits)



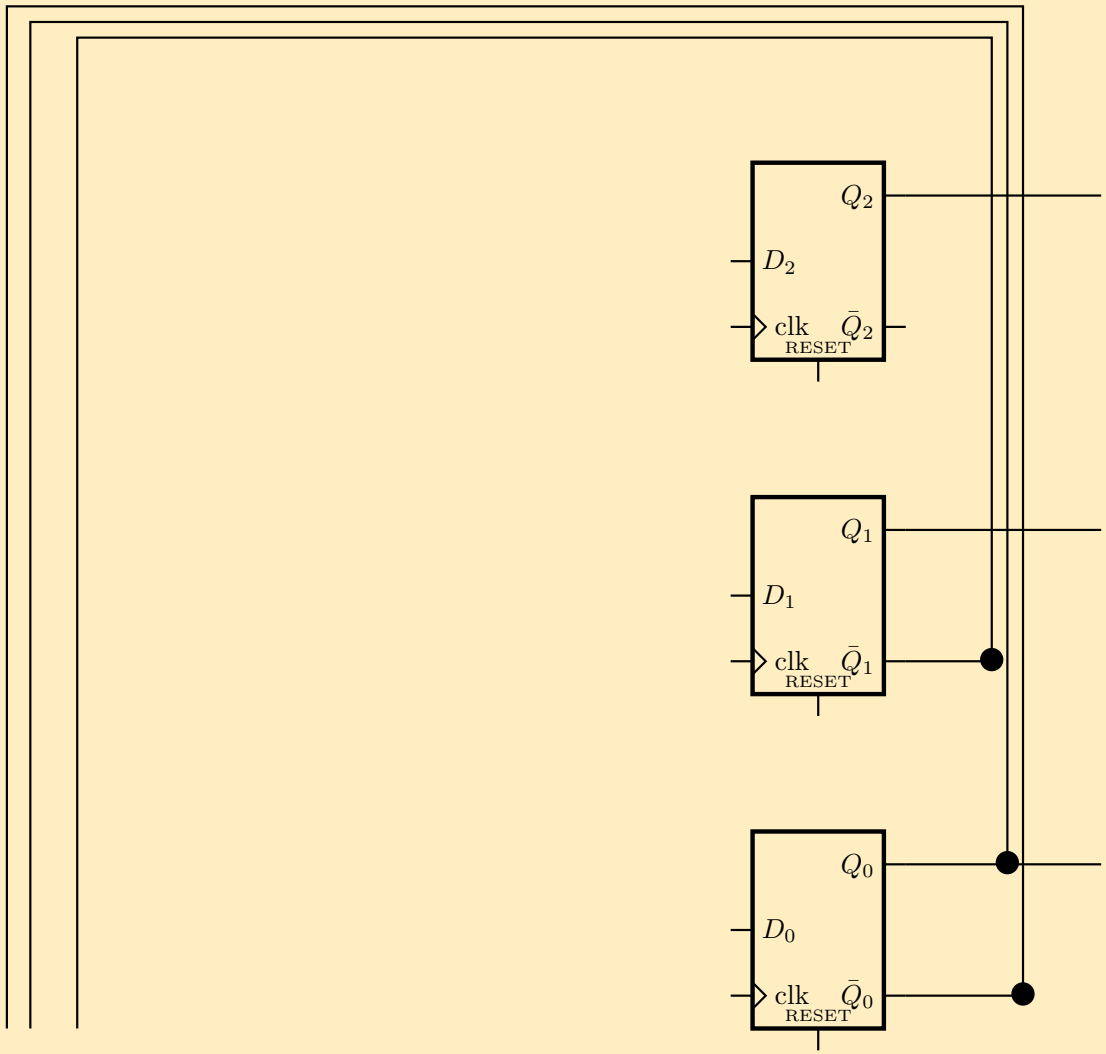
3-Bit Gray Counter (D-FF Circuits)



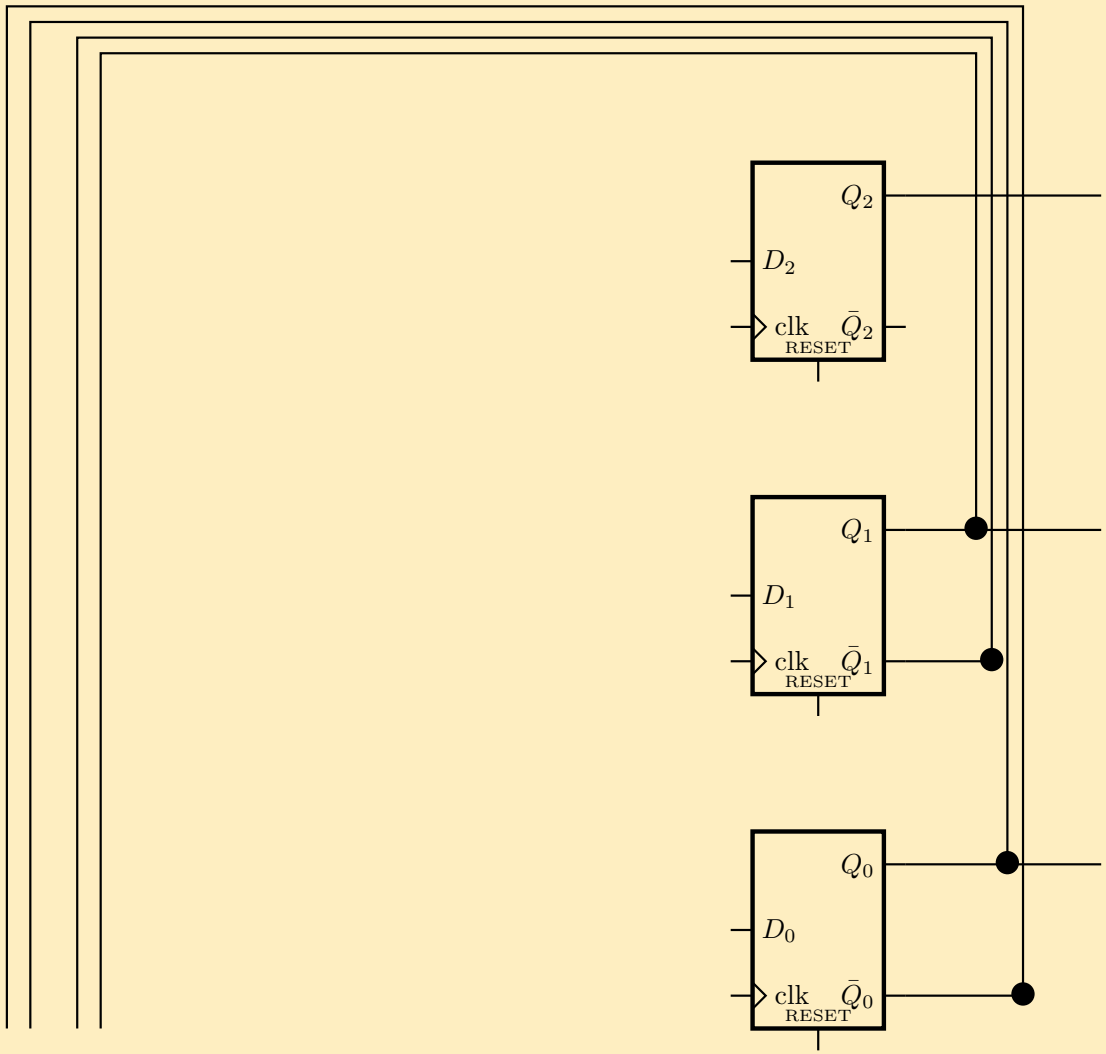
3-Bit Gray Counter (D-FF Circuits)



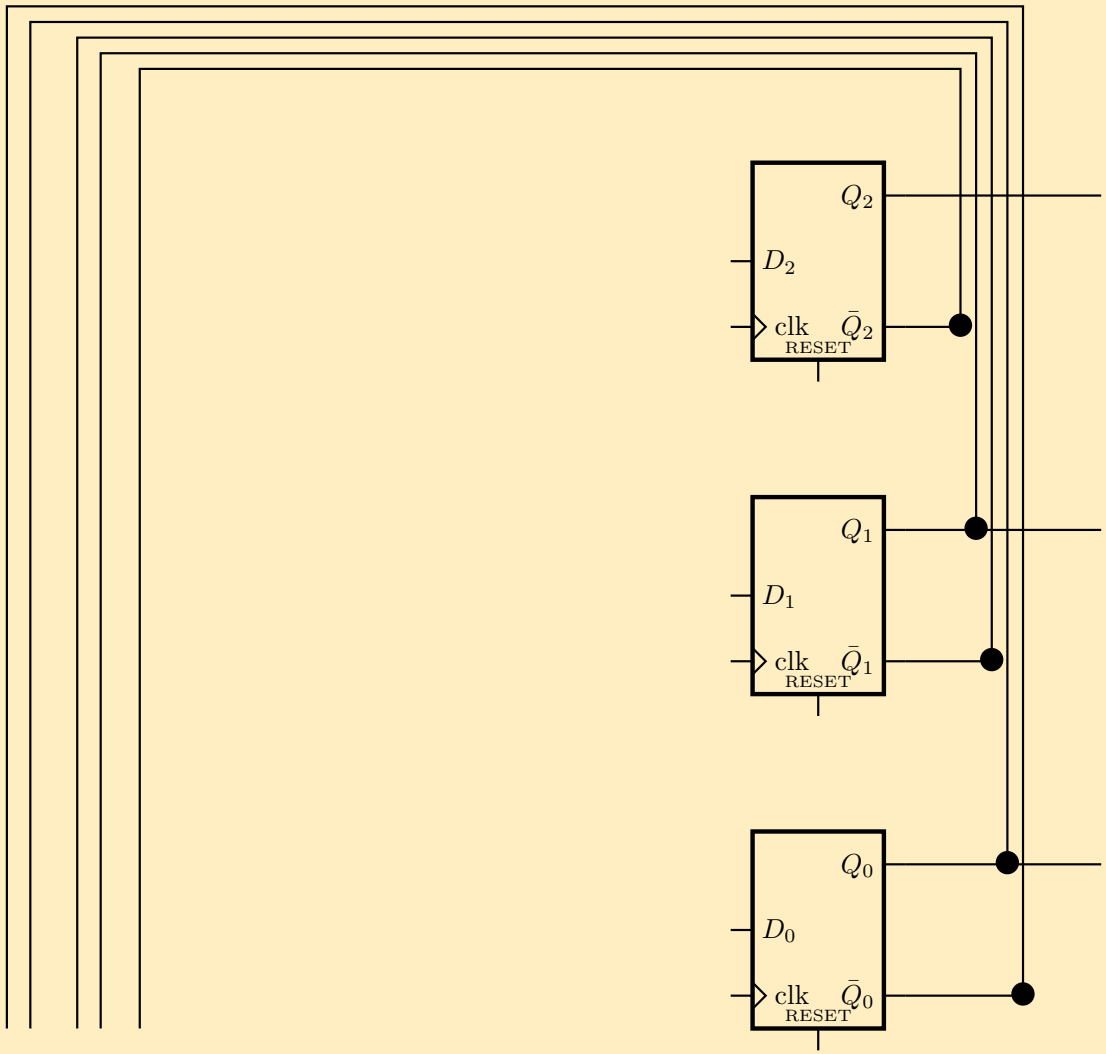
3-Bit Gray Counter (D-FF Circuits)



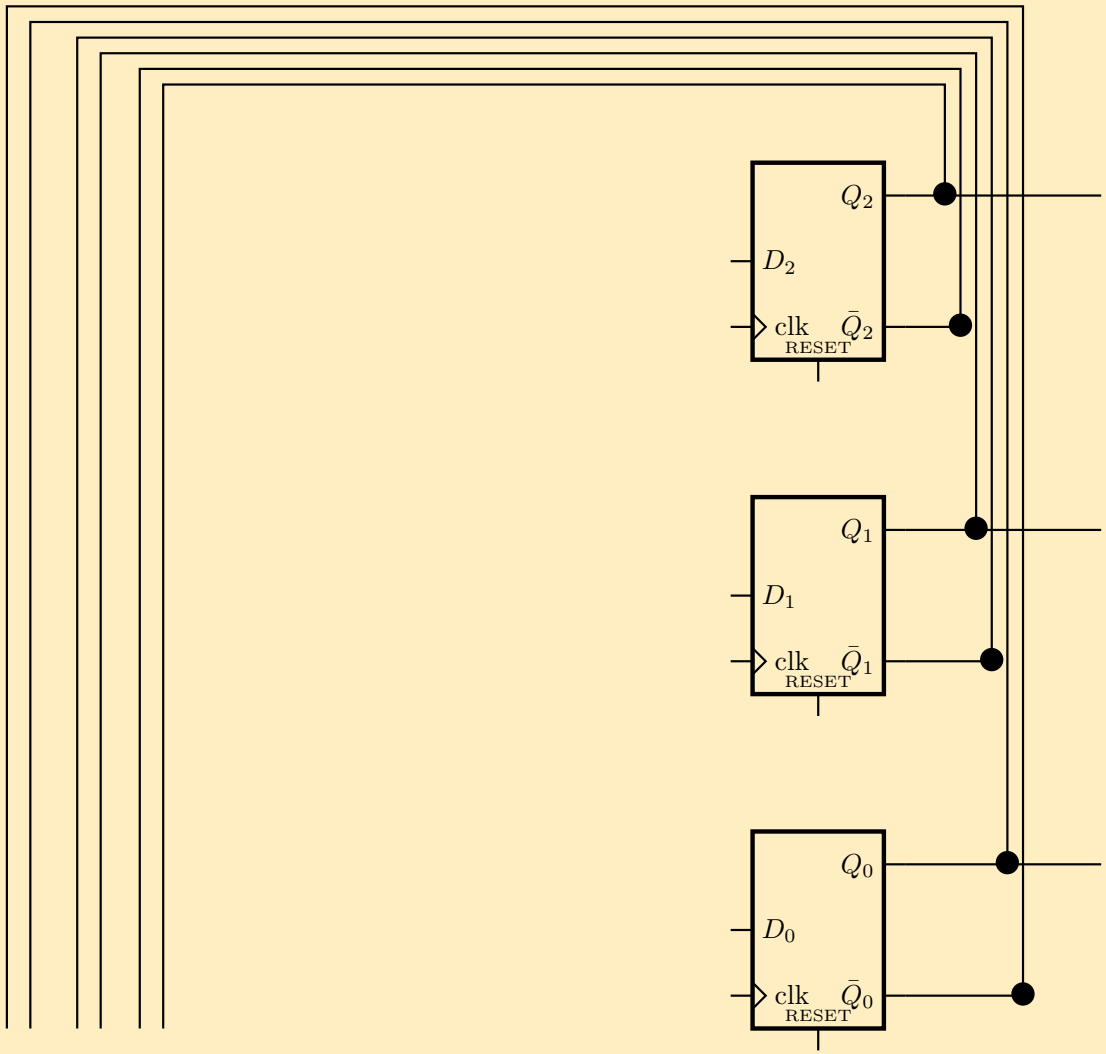
3-Bit Gray Counter (D-FF Circuits)



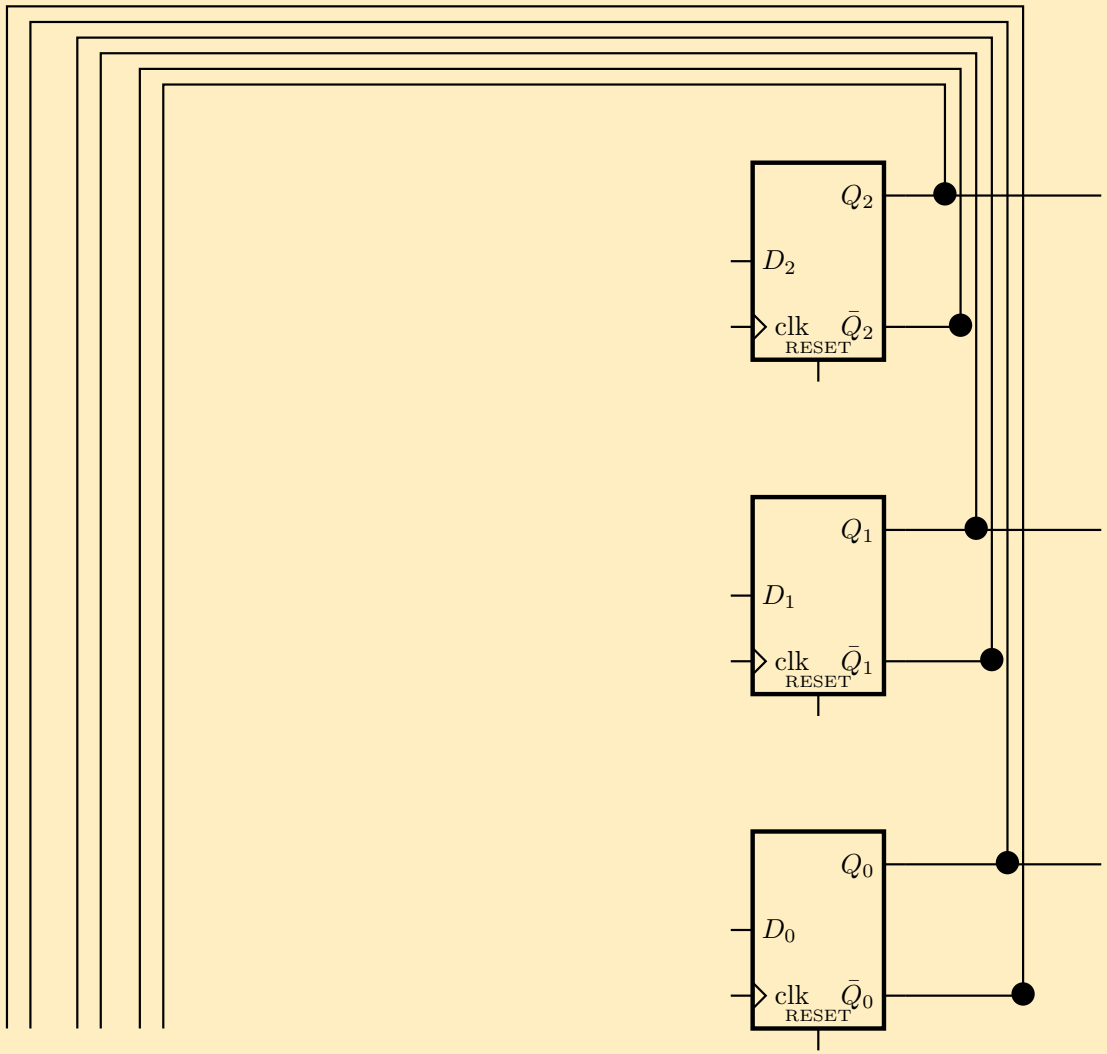
3-Bit Gray Counter (D-FF Circuits)



3-Bit Gray Counter (D-FF Circuits)

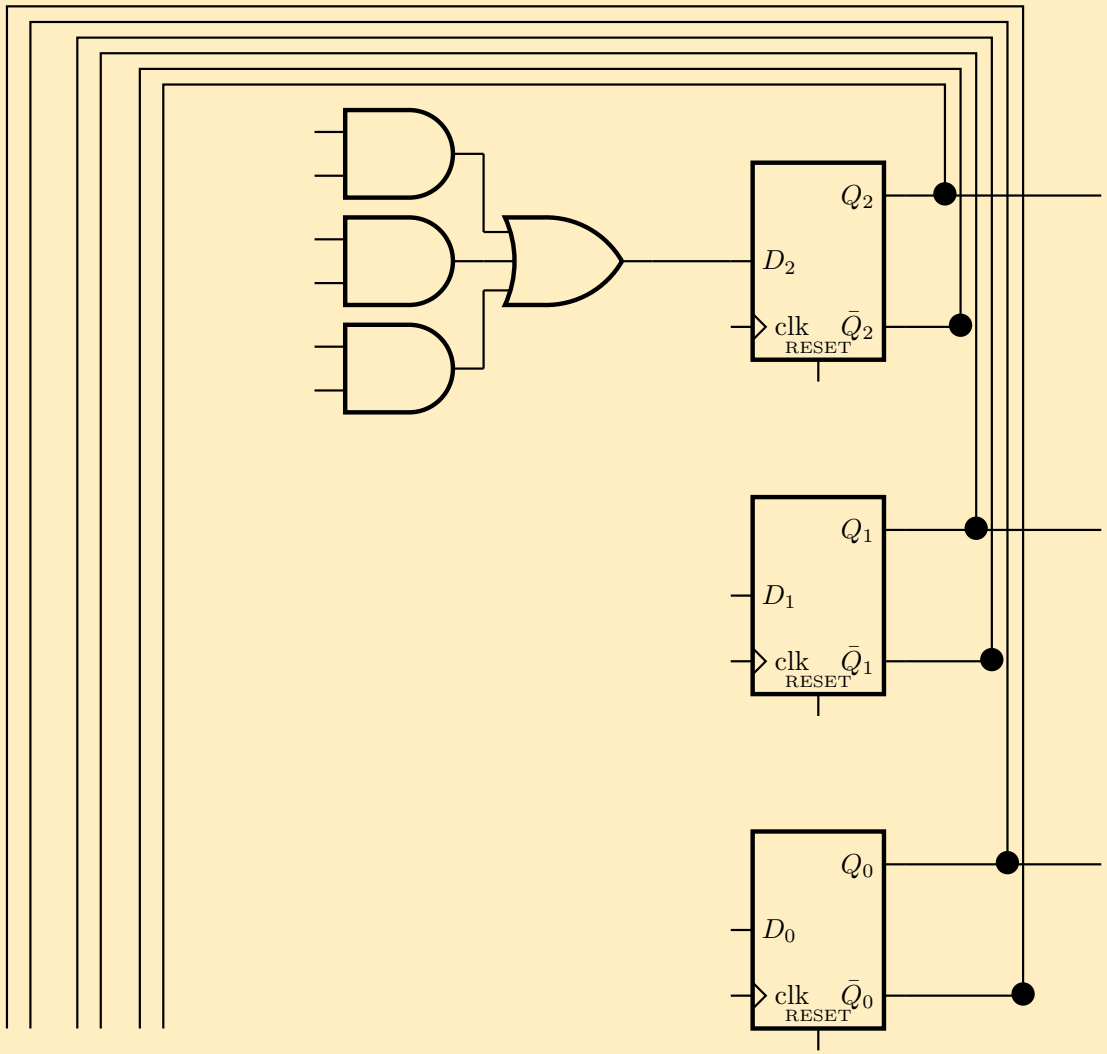


3-Bit Gray Counter (D-FF Circuits)



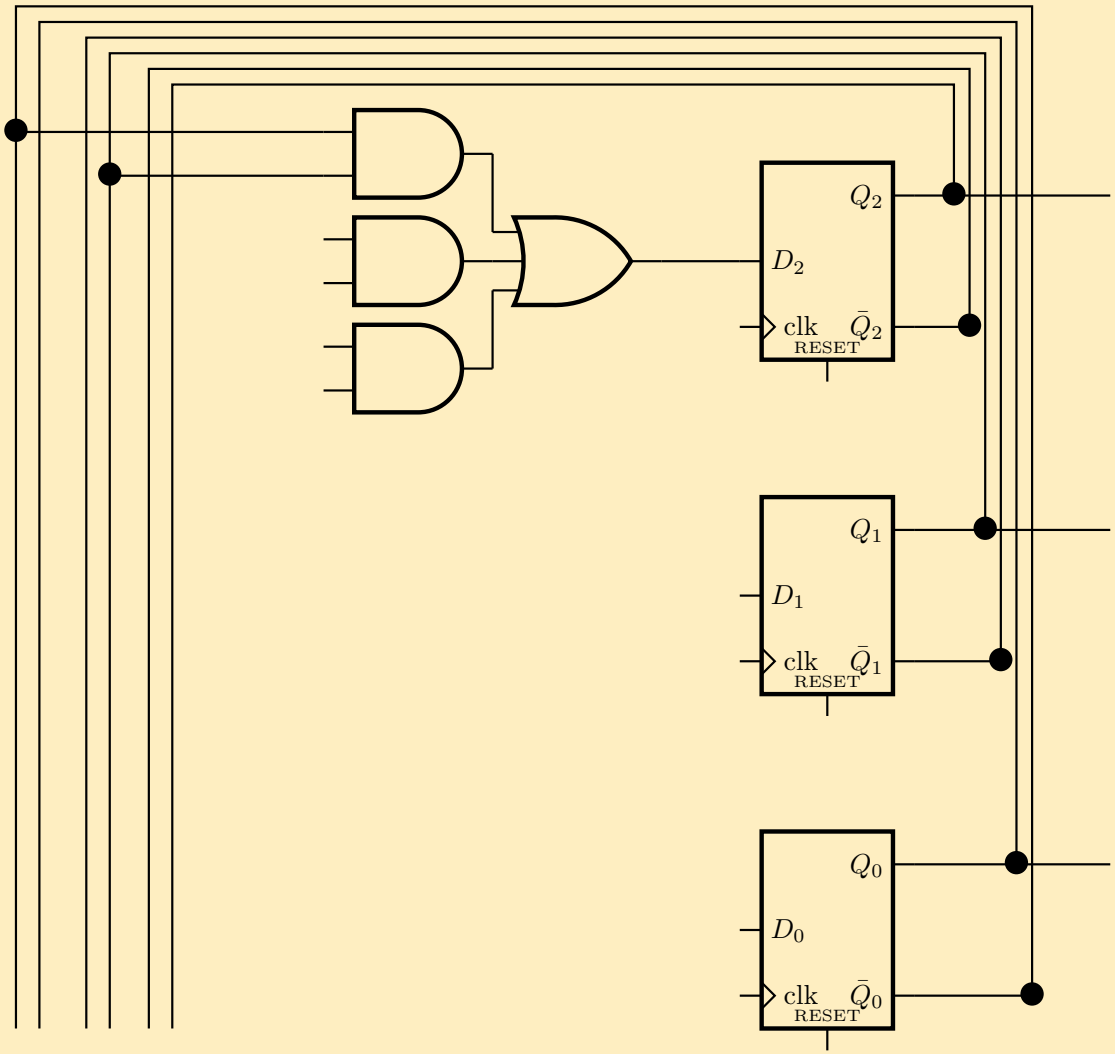
$$D_2 = Q_1\bar{Q}_0 + Q_2Q_0 + Q_2Q_1$$

3-Bit Gray Counter (D-FF Circuits)



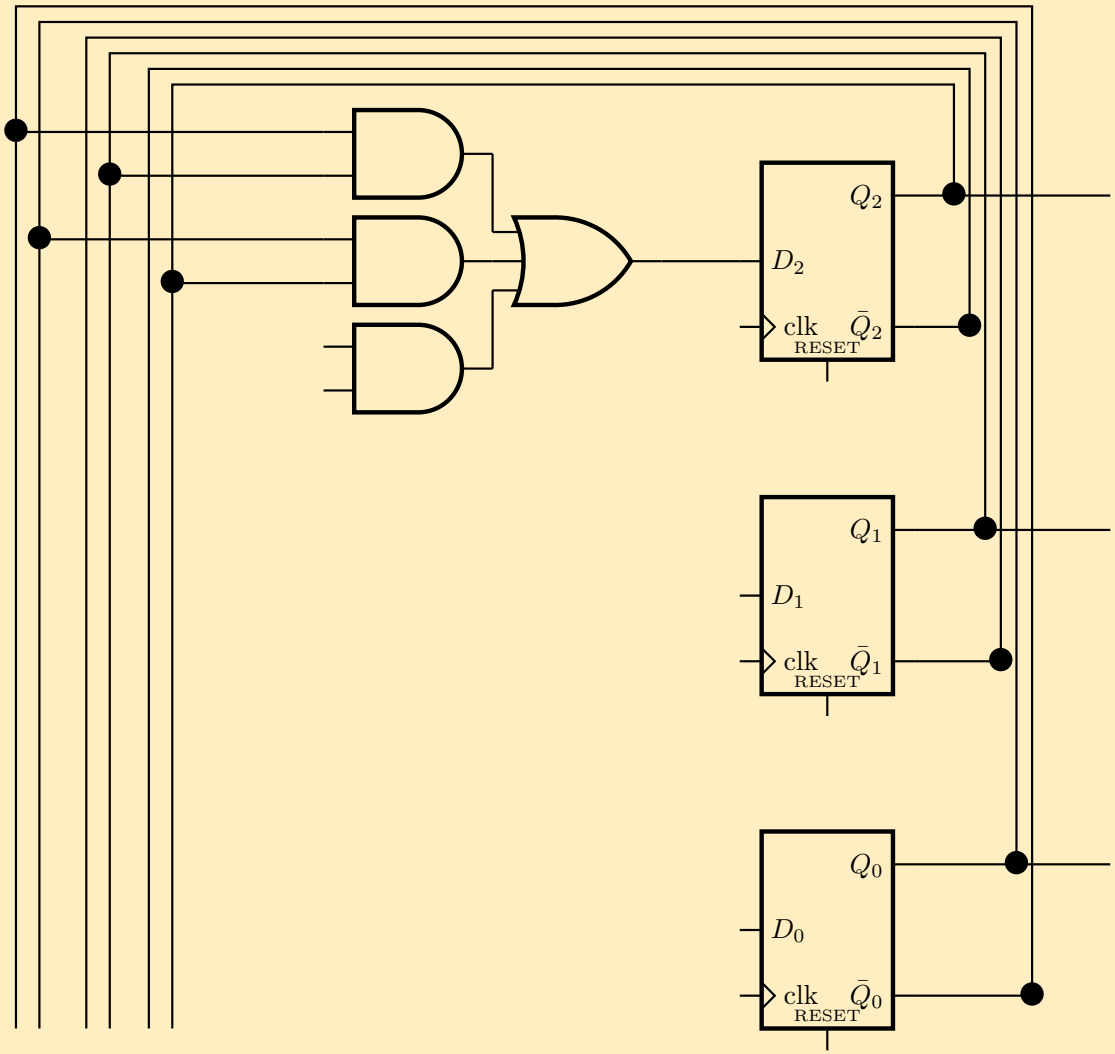
$$D_2 = Q_1\bar{Q}_0 + Q_2Q_0 + Q_2Q_1$$

3-Bit Gray Counter (D-FF Circuits)



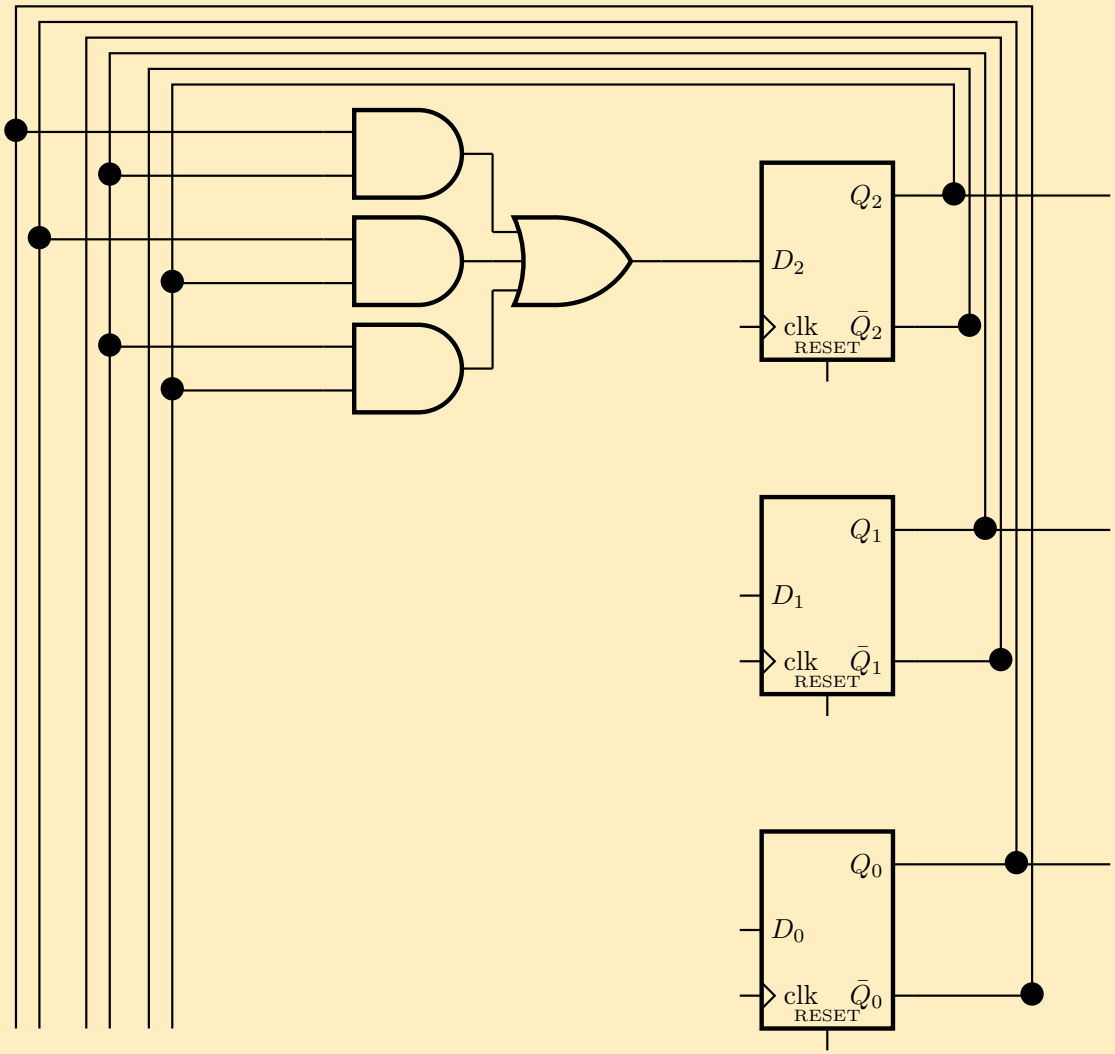
$$D_2 = Q_1\bar{Q}_0 + Q_2Q_0 + Q_2Q_1$$

3-Bit Gray Counter (D-FF Circuits)



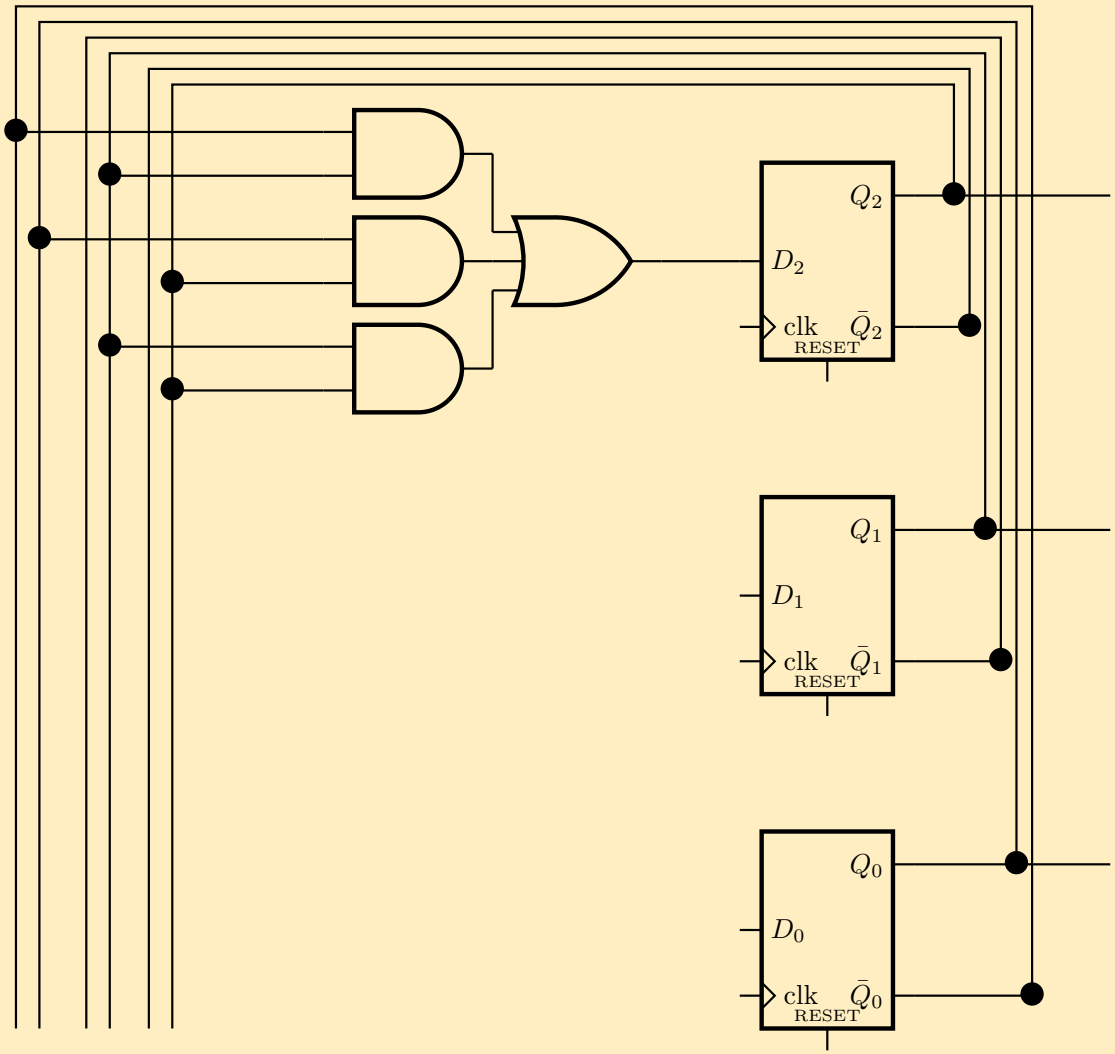
$$D_2 = Q_1\bar{Q}_0 + Q_2Q_0 + Q_2Q_1$$

3-Bit Gray Counter (D-FF Circuits)



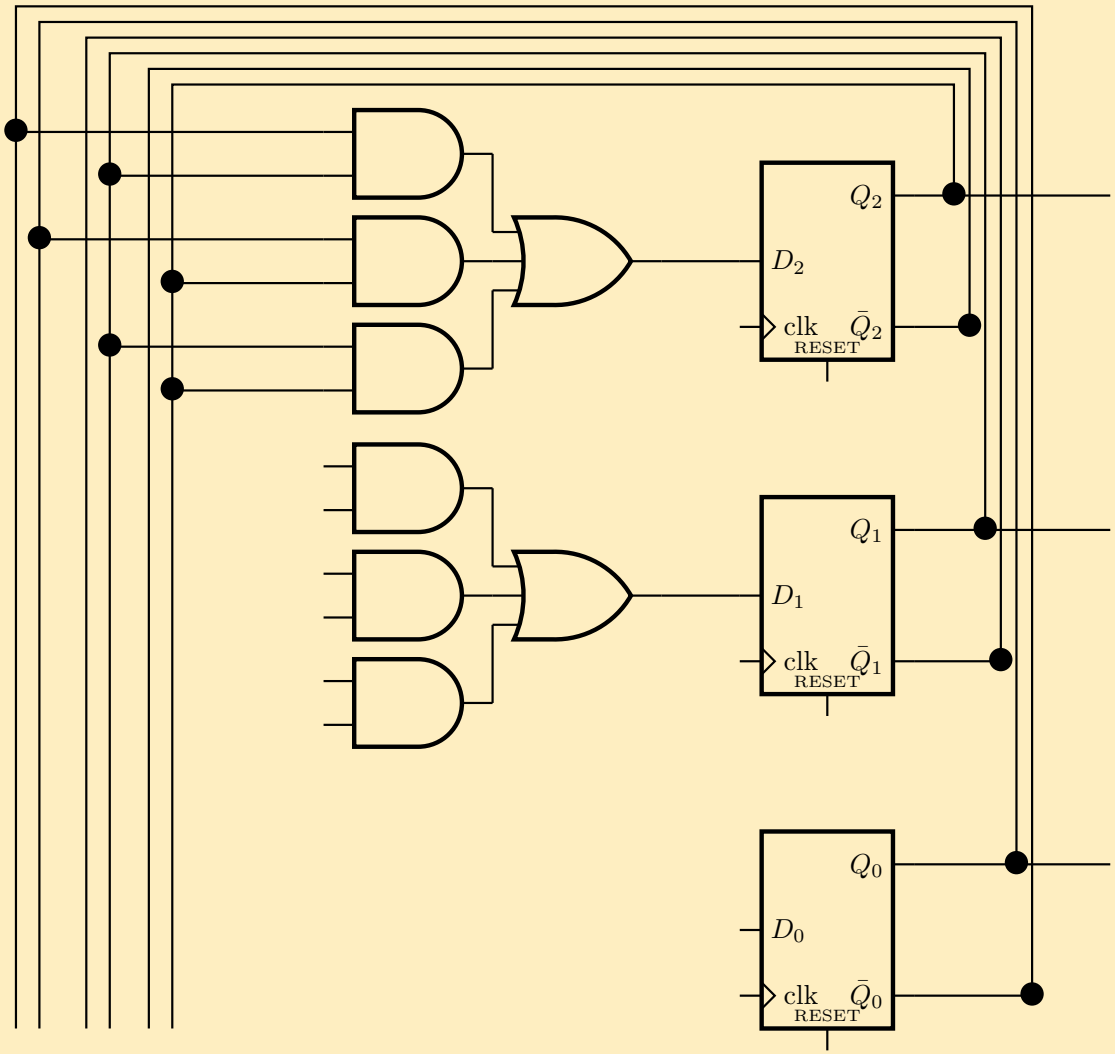
$$D_2 = Q_1\bar{Q}_0 + Q_2Q_0 + Q_2Q_1$$

3-Bit Gray Counter (D-FF Circuits)



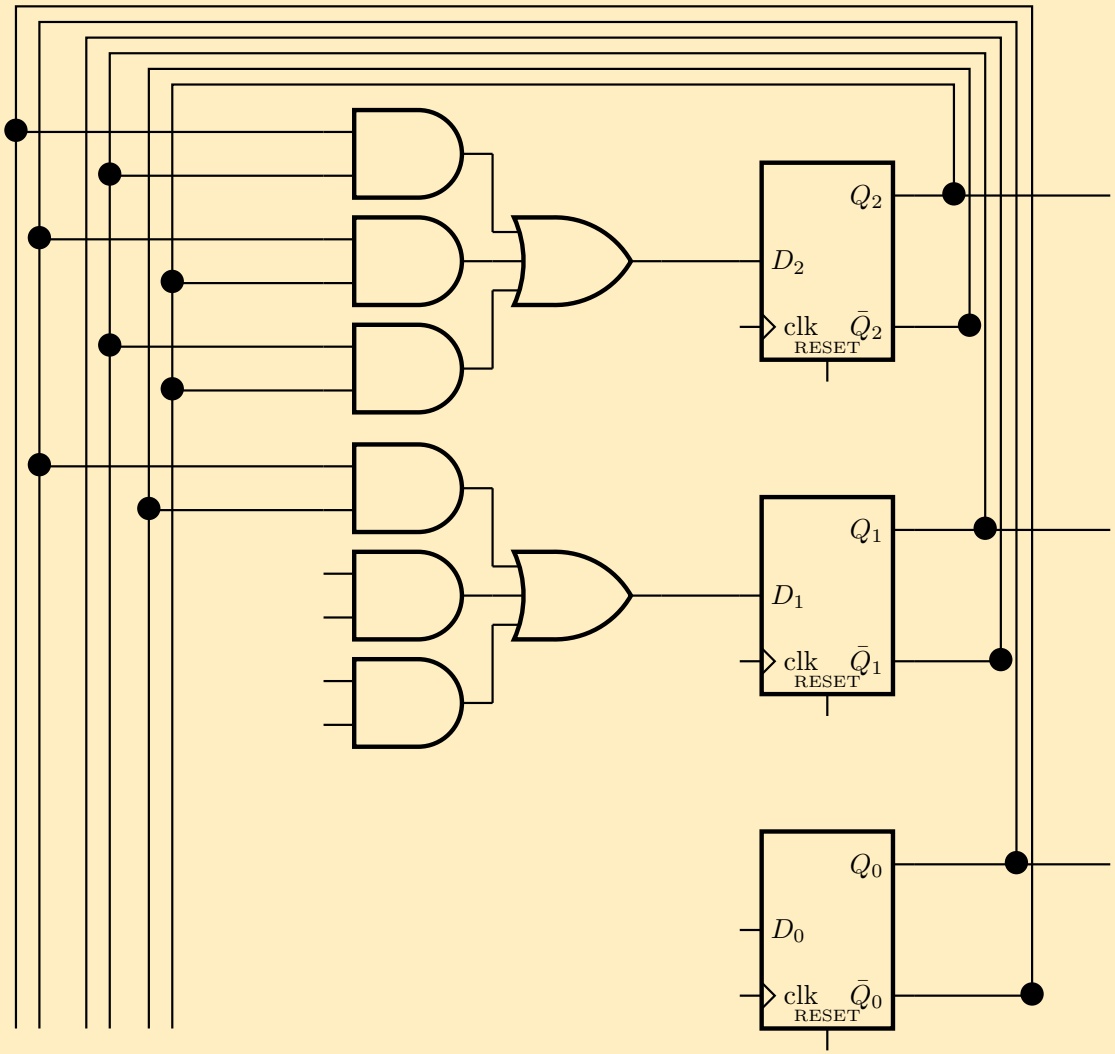
$$D_1 = \bar{Q}_2 Q_0 + \bar{Q}_2 Q_1 + Q_1 \bar{Q}_0$$

3-Bit Gray Counter (D-FF Circuits)



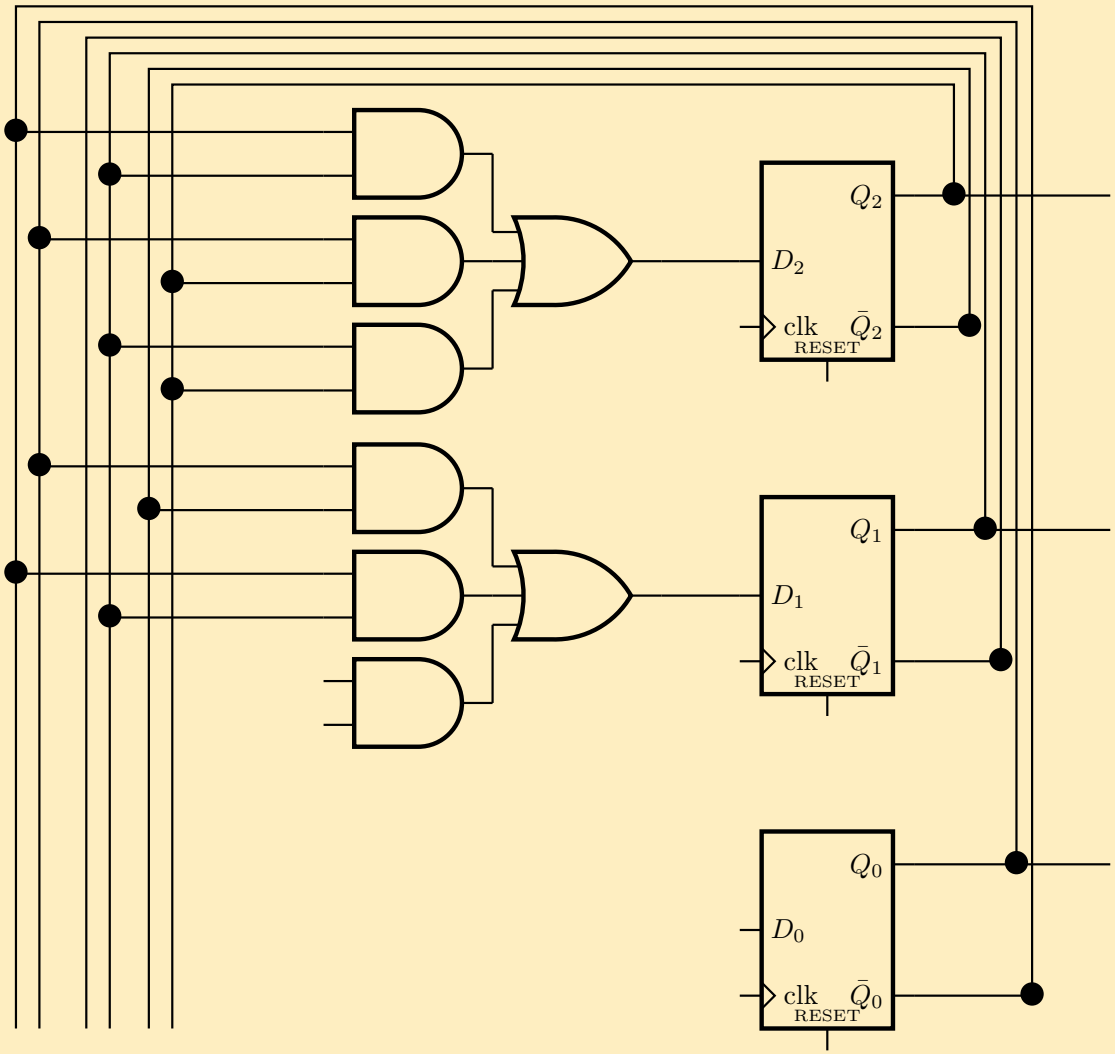
$$D_1 = \bar{Q}_2 Q_0 + \bar{Q}_2 Q_1 + Q_1 \bar{Q}_0$$

3-Bit Gray Counter (D-FF Circuits)



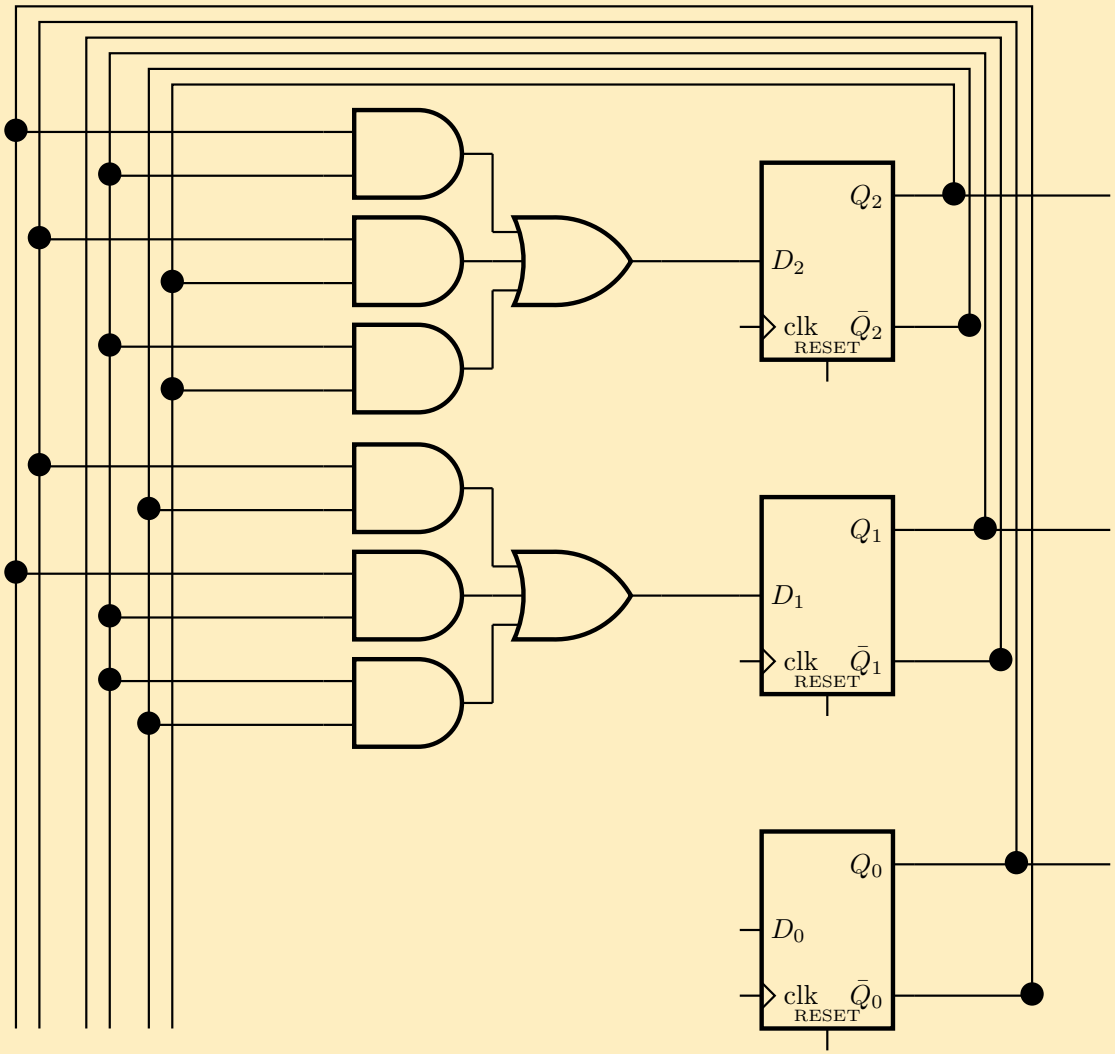
$$D_1 = \bar{Q}_2 Q_0 + \bar{Q}_2 Q_1 + Q_1 \bar{Q}_0$$

3-Bit Gray Counter (D-FF Circuits)



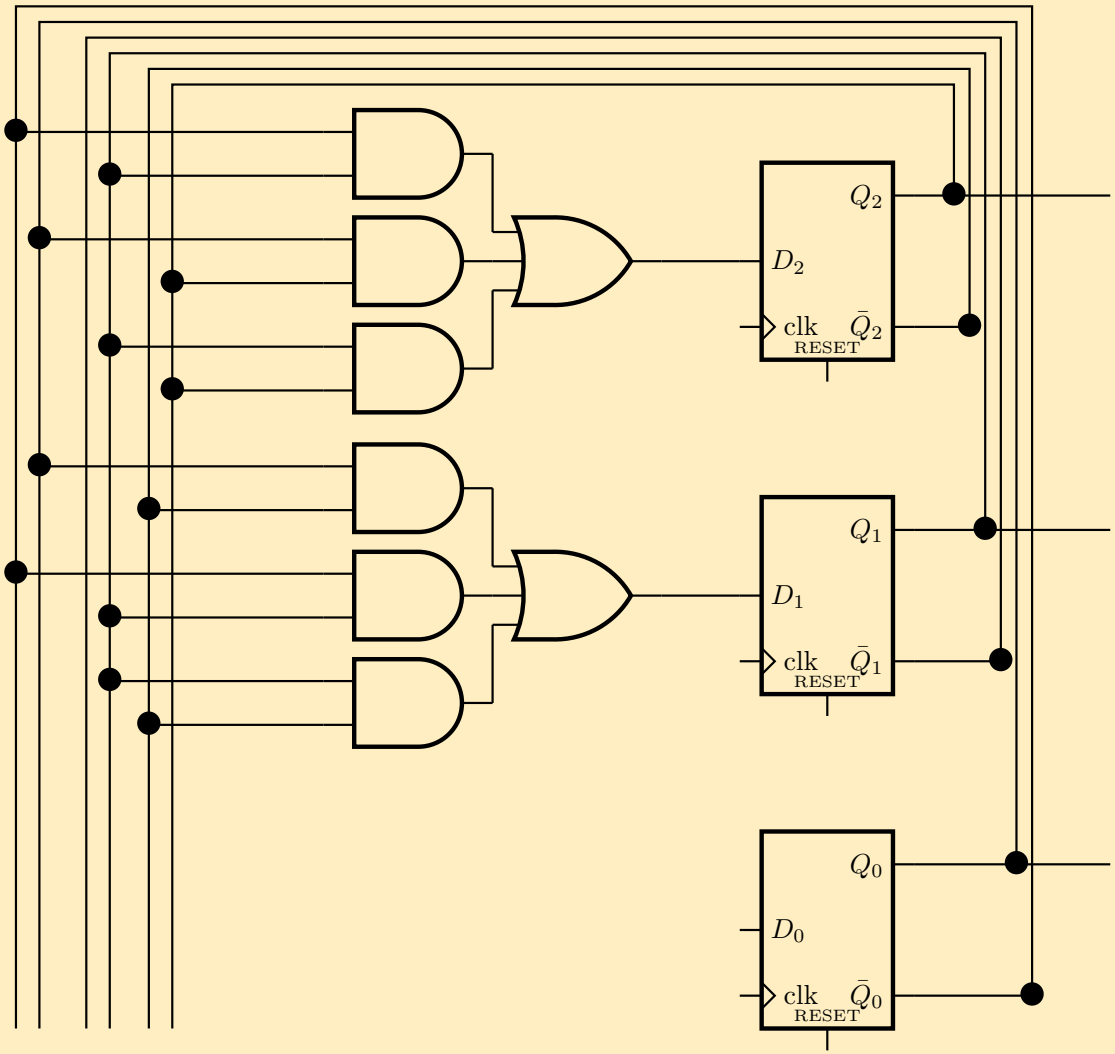
$$D_1 = \bar{Q}_2 Q_0 + \bar{Q}_2 Q_1 + Q_1 \bar{Q}_0$$

3-Bit Gray Counter (D-FF Circuits)



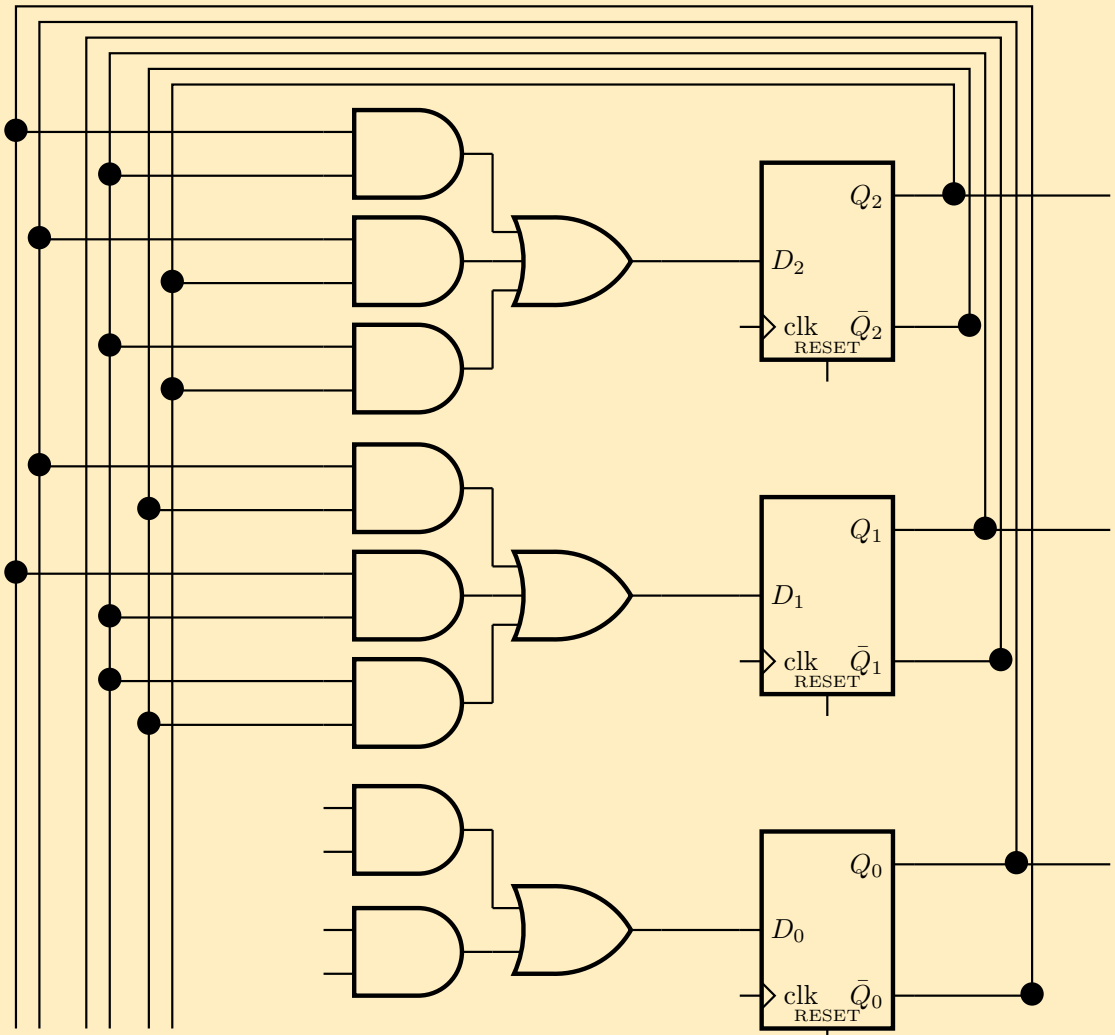
$$D_1 = \bar{Q}_2 Q_0 + \bar{Q}_2 Q_1 + Q_1 \bar{Q}_0$$

3-Bit Gray Counter (D-FF Circuits)



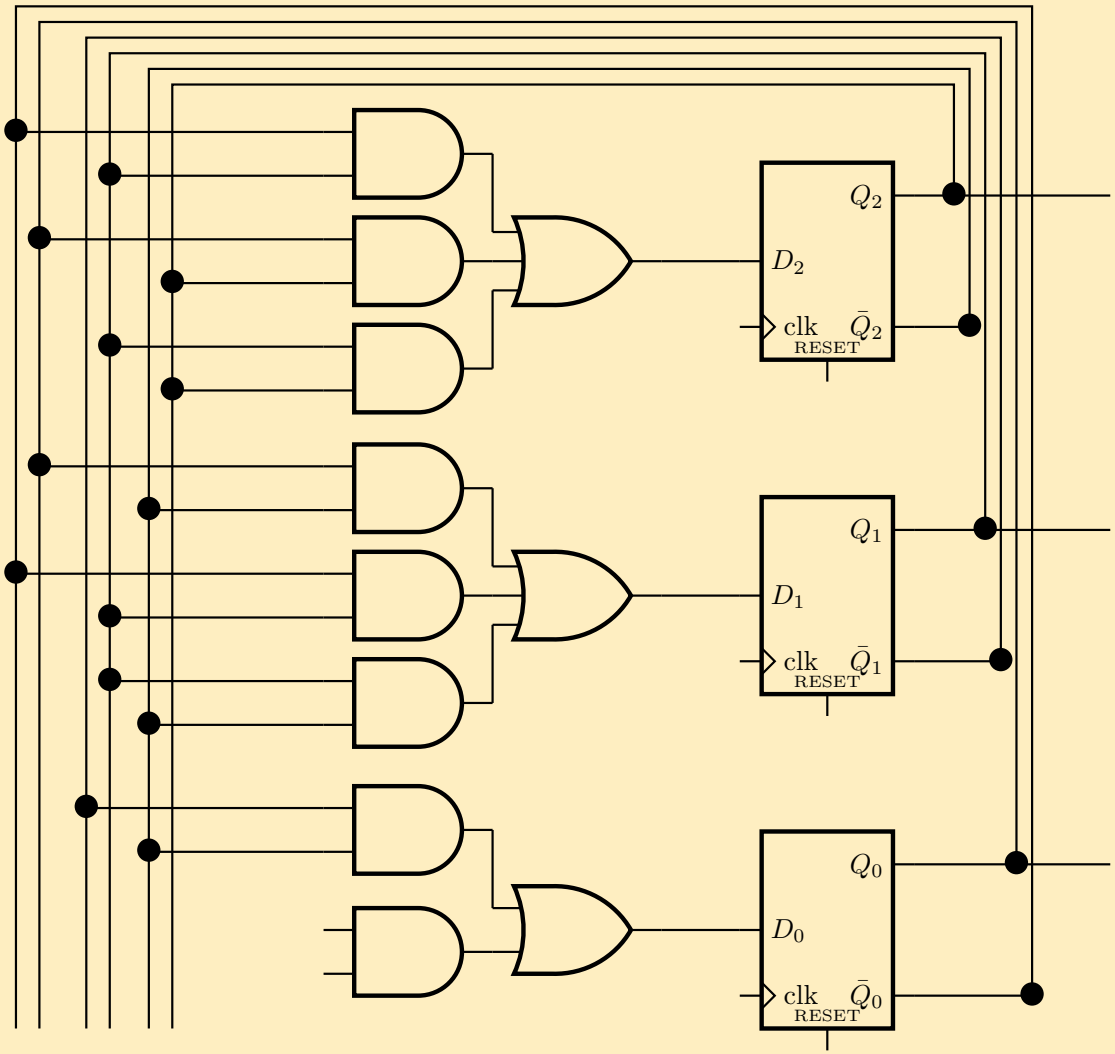
$$D_0 = \bar{Q}_2\bar{Q}_1 + Q_2Q_1$$

3-Bit Gray Counter (D-FF Circuits)



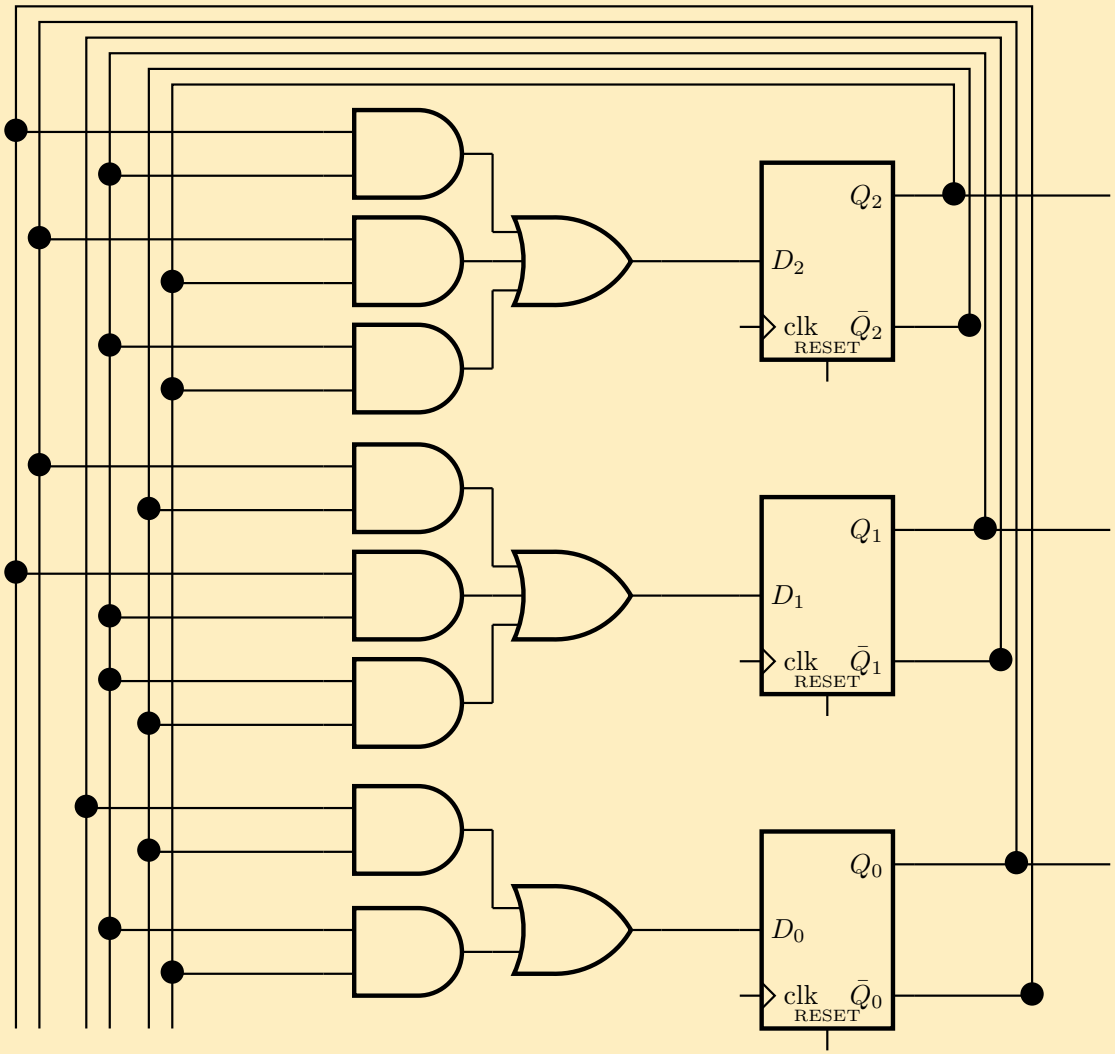
$$D_0 = \bar{Q}_2\bar{Q}_1 + Q_2Q_1$$

3-Bit Gray Counter (D-FF Circuits)



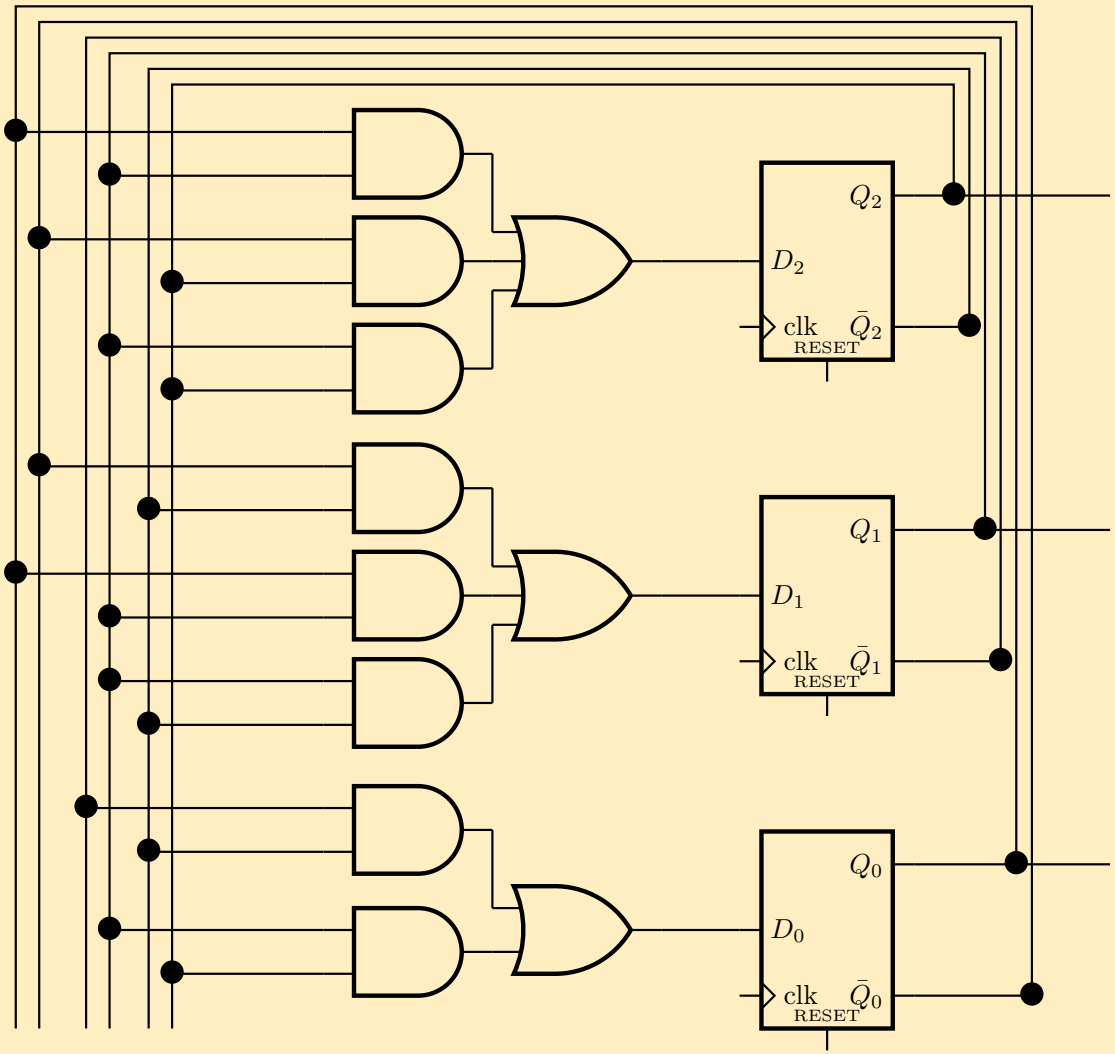
$$D_0 = \bar{Q}_2\bar{Q}_1 + Q_2Q_1$$

3-Bit Gray Counter (D-FF Circuits)



$$D_0 = \bar{Q}_2\bar{Q}_1 + Q_2Q_1$$

3-Bit Gray Counter (D-FF Circuits)



3-Bits Gray Counter I

Listing 13: demo/07_gray3d/gray3.v

```
`timescale 1ns/1ns

module gray3(
  output x2,x1,x0,
  input init,
  input clk
);
  wire q2,q2n,q1,q1n,q0,q0n;
  wire d2,d1,d0;

  wire a20,a21,a22;
  andGate and20(a20,q1,q0n);
  andGate and21(a21,q2,q0);
```

3-Bits Gray Counter II

```
andGate and22(a22,q2,q1);
or3Gate o2(d2,a20,a21,a22);

wire a10,a11,a12;
andGate and10(a10,q2n,q0);
andGate and11(a11,q2n,q1);
andGate and12(a12,q1,q0n);
or3Gate o1(d1,a10,a11,a12);

xnorGate o0(d0,q1,q2);

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

3-Bits Gray Counter III

```
  assign x2 = q2;
  assign x1 = q1;
  assign x0 = q0;

endmodule
```

3-Bits Gray Conter I

Listing 14: demo/07_gray3d/gray3_tb.v

```
`timescale 1ns/1ns

module gray3_tb;

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

  logic init;
  wire x2, x1, x0;
  gray3 g(x2, x1, x0, init, clk);

  initial begin
```

3-Bits Gray Conter II

```
    $dumpfile("gray3.vcd");
    $dumpvars;

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

    #2000ns;
    $finish;
  end

endmodule
```

3-Bit Gray Counter

(using SR-FF)

SR-FF Excitation Table

- If we want the Q of D-FF to have some value
  - We set the D to this value
- If we want the Q of SR-FF to have some value
  - We need to use the **excitation table**

SR-FF Tables

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

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

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1						
0	0	1	0	1	1						
0	1	0	1	1	0						
0	1	1	0	1	0						
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X				
0	0	1	0	1	1						
0	1	0	1	1	0						
0	1	1	0	1	0						
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X		
0	0	1	0	1	1						
0	1	0	1	1	0						
0	1	1	0	1	0						
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1						
0	1	0	1	1	0						
0	1	1	0	1	0						
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X				
0	1	0	1	1	0						
0	1	1	0	1	0						
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0		
0	1	0	1	1	0						
0	1	1	0	1	0						
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0						
0	1	1	0	1	0						
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0				
0	1	1	0	1	0						
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0		
0	1	1	0	1	0						
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0						
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X				
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0		
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0						
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1				
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1	0	X		
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1	0	X	0	X
1	0	1	1	0	0						
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1	0	X	0	X
1	0	1	1	0	0	X	0				
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1	0	X	0	X
1	0	1	1	0	0	X	0	0	X		
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1	0	X	0	X
1	0	1	1	0	0	X	0	0	X	0	1
1	1	0	1	1	1						
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1	0	X	0	X
1	0	1	1	0	0	X	0	0	X	0	1
1	1	0	1	1	1	X	0				
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1	0	X	0	X
1	0	1	1	0	0	X	0	0	X	0	1
1	1	0	1	1	1	X	0	X	0		
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1	0	X	0	X
1	0	1	1	0	0	X	0	0	X	0	1
1	1	0	1	1	1	X	0	X	0	1	0
1	1	1	1	0	1						

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1	0	X	0	X
1	0	1	1	0	0	X	0	0	X	0	1
1	1	0	1	1	1	X	0	X	0	1	0
1	1	1	1	0	1	X	0				

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1	0	X	0	X
1	0	1	1	0	0	X	0	0	X	0	1
1	1	0	1	1	1	X	0	X	0	1	0
1	1	1	1	0	1	X	0	0	1		

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1	0	X	0	X
1	0	1	1	0	0	X	0	0	X	0	1
1	1	0	1	1	1	X	0	X	0	1	0
1	1	1	1	0	1	X	0	0	1	X	0

3-Bit Gray Counter (SR-FF)

State and FF inputs

state			next state			FF inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	S_2	R_2	S_1	R_1	S_0	R_0
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	1	0	X	1	0	X	0
0	1	0	1	1	0	1	0	X	0	0	X
0	1	1	0	1	0	0	X	X	0	0	1
1	0	0	0	0	0	0	1	0	X	0	X
1	0	1	1	0	0	X	0	0	X	0	1
1	1	0	1	1	1	X	0	X	0	1	0
1	1	1	1	0	1	X	0	0	1	X	0

3-Bit Gray Counter (SR-FF Input Equations)

state			
Q_2	Q_1	Q_0	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

3-Bit Gray Counter (SR-FF Input Equations)

state			
Q_2	Q_1	Q_0	S_2
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	X
1	1	0	X
1	1	1	X

$$S_2 = Q_1\bar{Q}_0$$

3-Bit Gray Counter (SR-FF Input Equations)

state			
Q_2	Q_1	Q_0	R_2
0	0	0	X
0	0	1	X
0	1	0	0
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

$$S_2 = Q_1\bar{Q}_0$$

$$\boxed{R_2 = \bar{Q}_1\bar{Q}_0}$$

3-Bit Gray Counter (SR-FF Input Equations)

state			
Q_2	Q_1	Q_0	S_1
0	0	0	0
0	0	1	1
0	1	0	X
0	1	1	X
1	0	0	0
1	0	1	0
1	1	0	X
1	1	1	0

$$S_2 = Q_1\bar{Q}_0$$

$$R_2 = \bar{Q}_1\bar{Q}_0$$

$$S_1 = \bar{Q}_2 Q_0$$

3-Bit Gray Counter (SR-FF Input Equations)

state			
Q_2	Q_1	Q_0	R_1
0	0	0	X
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	X
1	0	1	X
1	1	0	0
1	1	1	1

$$S_2 = Q_1\bar{Q}_0$$

$$R_2 = \bar{Q}_1\bar{Q}_0$$

$$S_1 = \bar{Q}_2 Q_0$$

$$\boxed{R_1 = Q_2 Q_0}$$

3-Bit Gray Counter (SR-FF Input Equations)

state			
Q_2	Q_1	Q_0	S_0
0	0	0	1
0	0	1	X
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	X

$$S_2 = Q_1\bar{Q}_0$$

$$R_2 = \bar{Q}_1\bar{Q}_0$$

$$S_1 = \bar{Q}_2 Q_0$$

$$R_1 = Q_2 Q_0$$

$$\boxed{S_0 = Q_2 Q_1 + \bar{Q}_2\bar{Q}_1}$$

3-Bit Gray Counter (SR-FF Input Equations)

state			
Q_2	Q_1	Q_0	R_0
0	0	0	0
0	0	1	0
0	1	0	X
0	1	1	1
1	0	0	X
1	0	1	1
1	1	0	0
1	1	1	0

$$S_2 = Q_1\bar{Q}_0$$

$$R_2 = \bar{Q}_1\bar{Q}_0$$

$$S_1 = \bar{Q}_2 Q_0$$

$$R_1 = Q_2 Q_0$$

$$S_0 = Q_2 Q_1 + \bar{Q}_2\bar{Q}_1$$

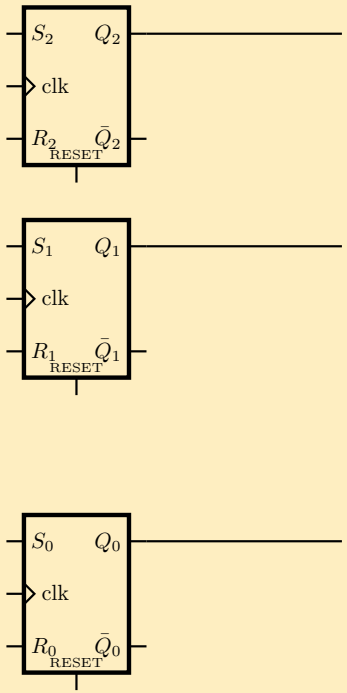
$$\boxed{R_0 = \bar{Q}_2 Q_1 + Q_2\bar{Q}_1}$$

3-Bit Gray Counter (SR-FF Input Equations)

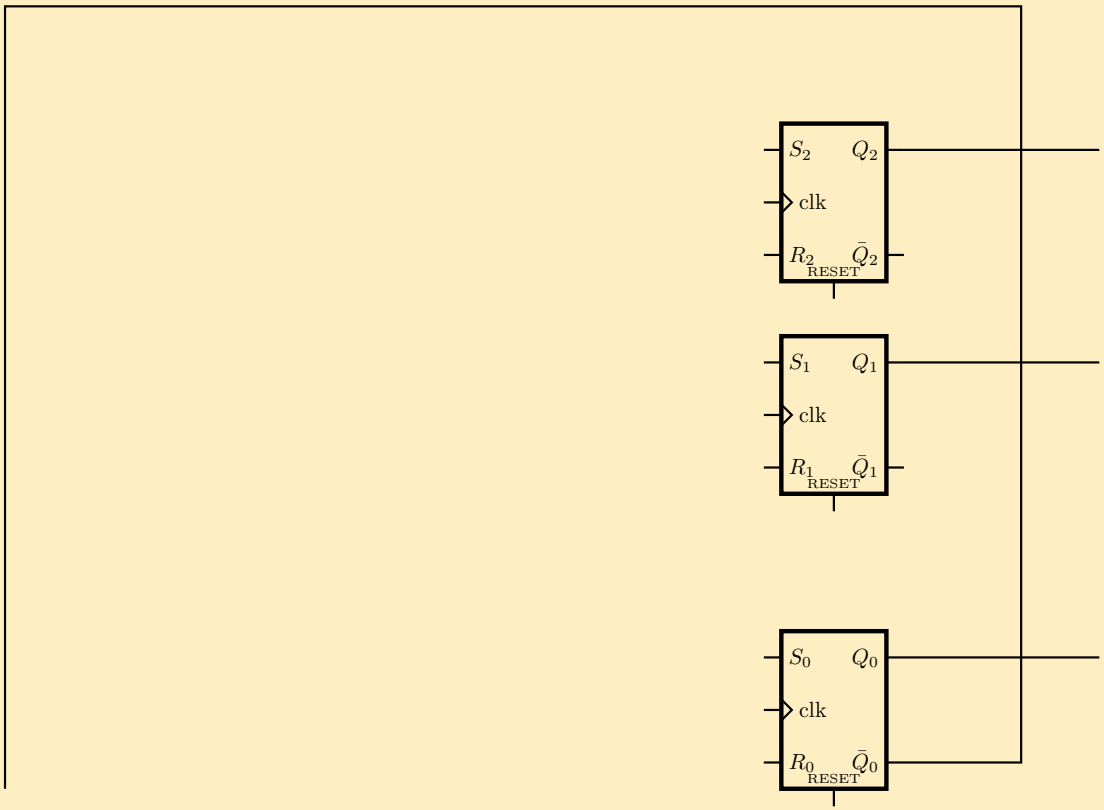
state			
Q_2	Q_1	Q_0	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

$$S_2 = Q_1\bar{Q}_0$$
$$R_2 = \bar{Q}_1\bar{Q}_0$$
$$S_1 = \bar{Q}_2Q_0$$
$$R_1 = Q_2Q_0$$
$$S_0 = Q_2Q_1 + \bar{Q}_2\bar{Q}_1$$
$$R_0 = \bar{Q}_2Q_1 + Q_2\bar{Q}_1$$

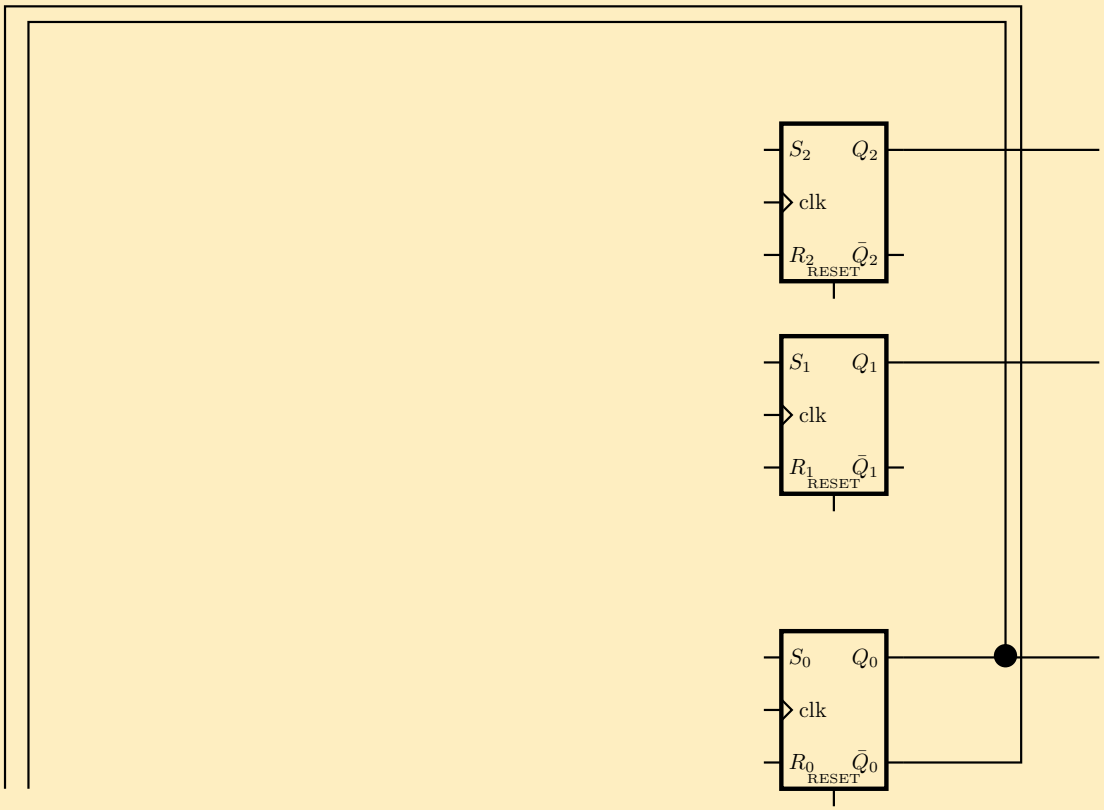
3-Bit Gray Counter (SR-FF Circuits)



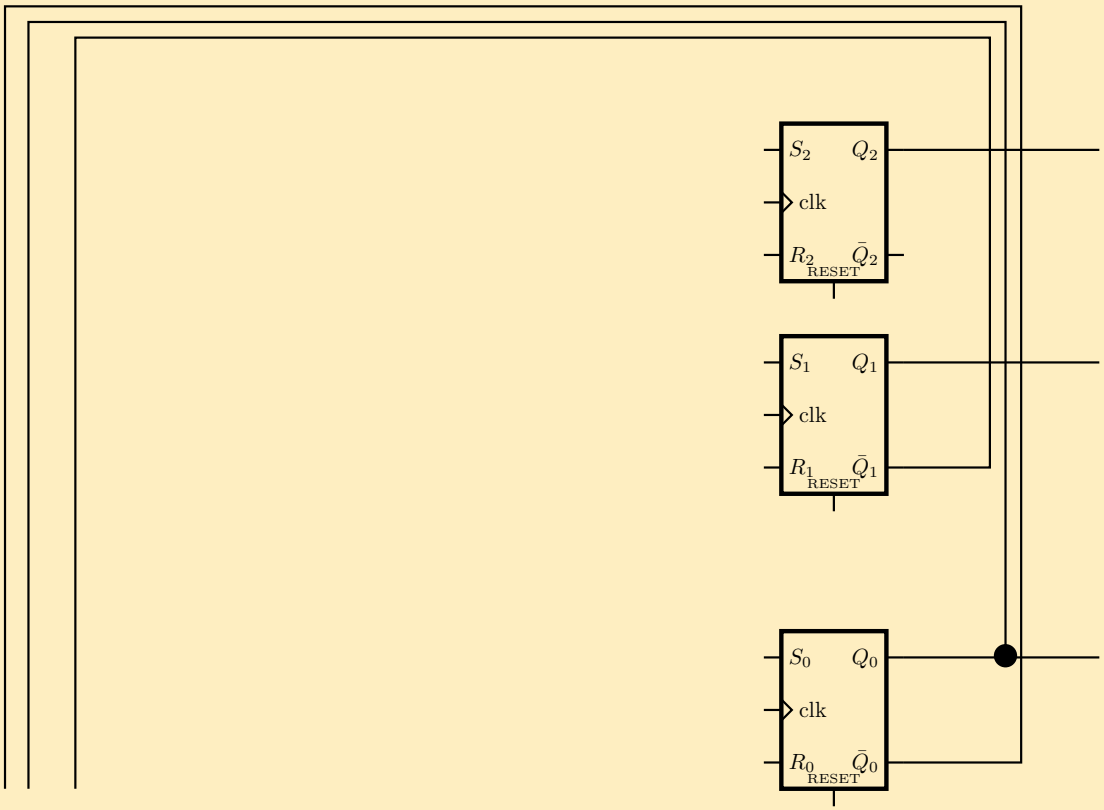
3-Bit Gray Counter (SR-FF Circuits)



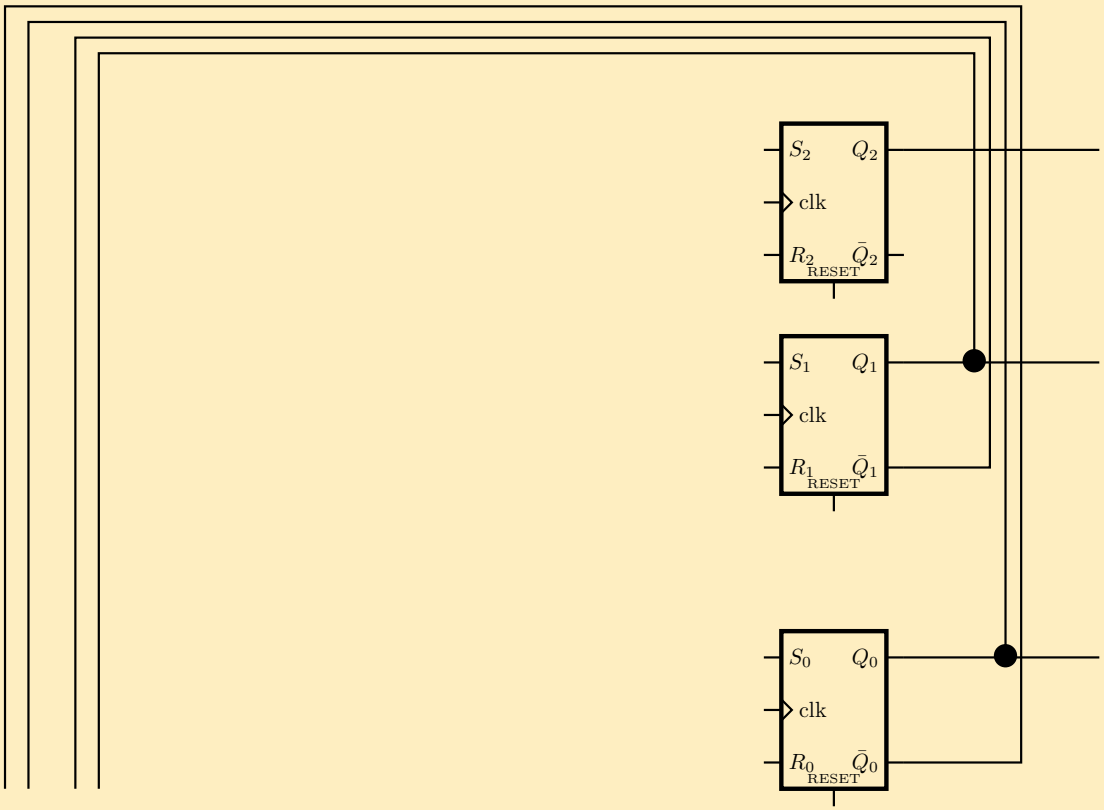
3-Bit Gray Counter (SR-FF Circuits)



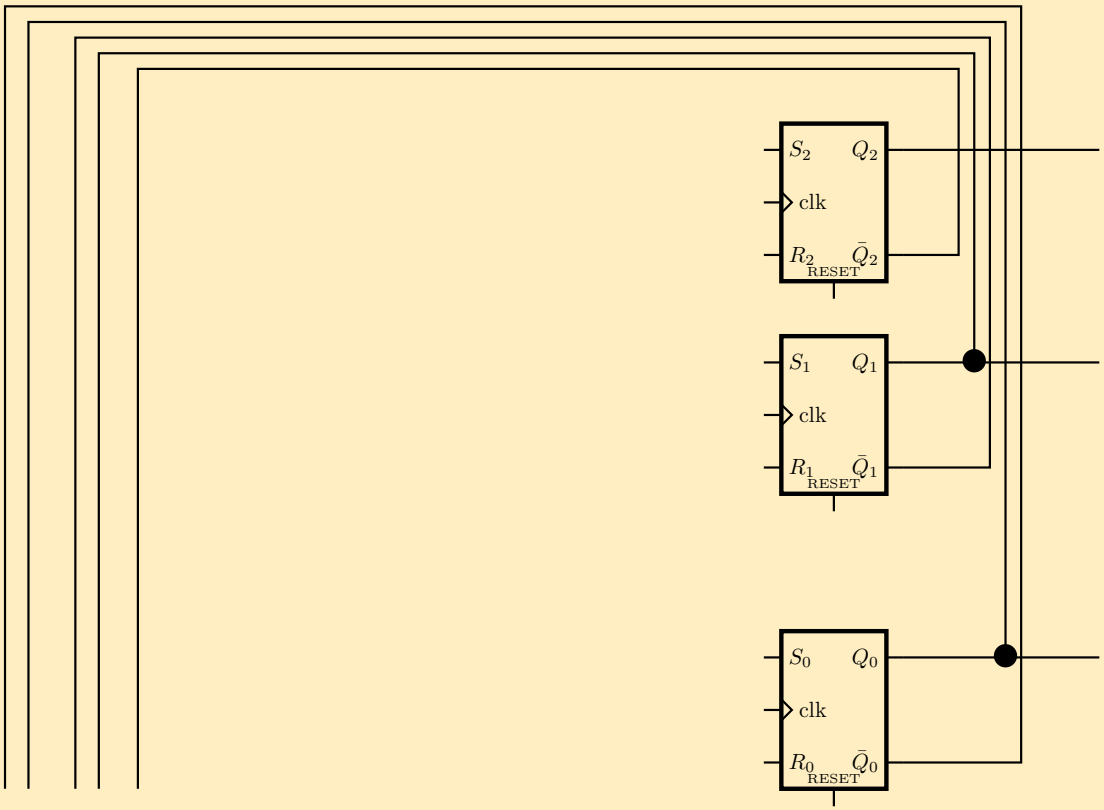
3-Bit Gray Counter (SR-FF Circuits)



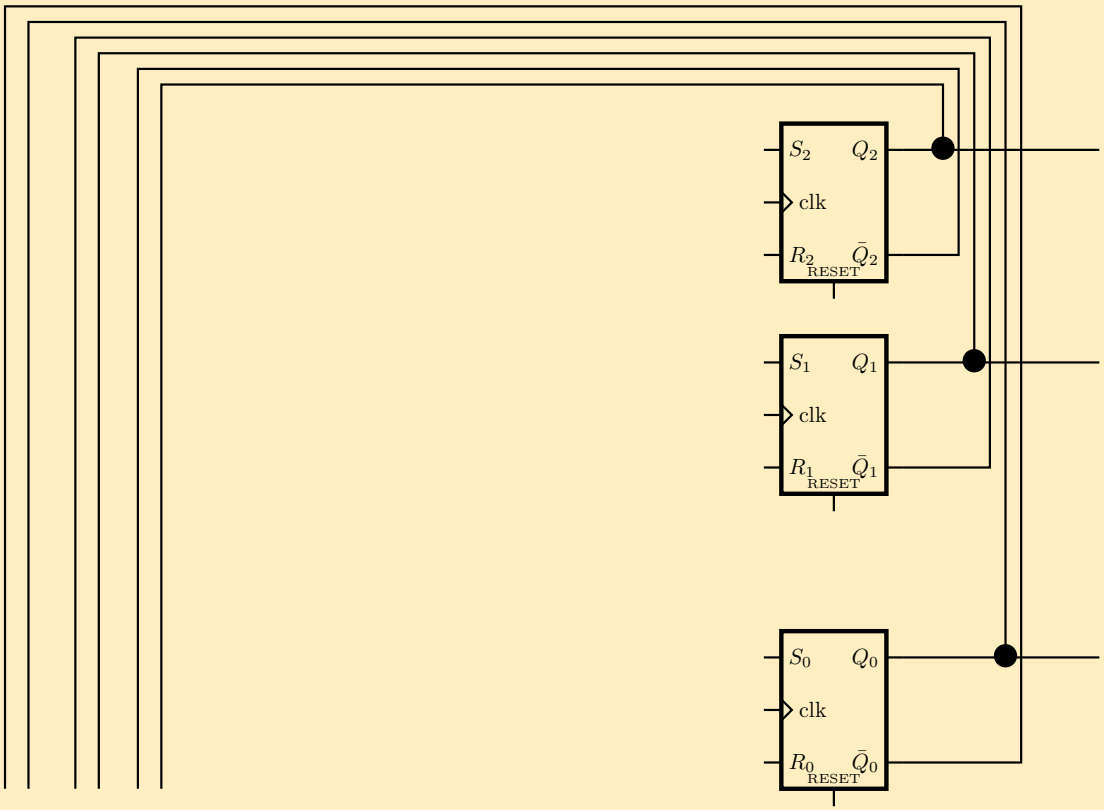
3-Bit Gray Counter (SR-FF Circuits)



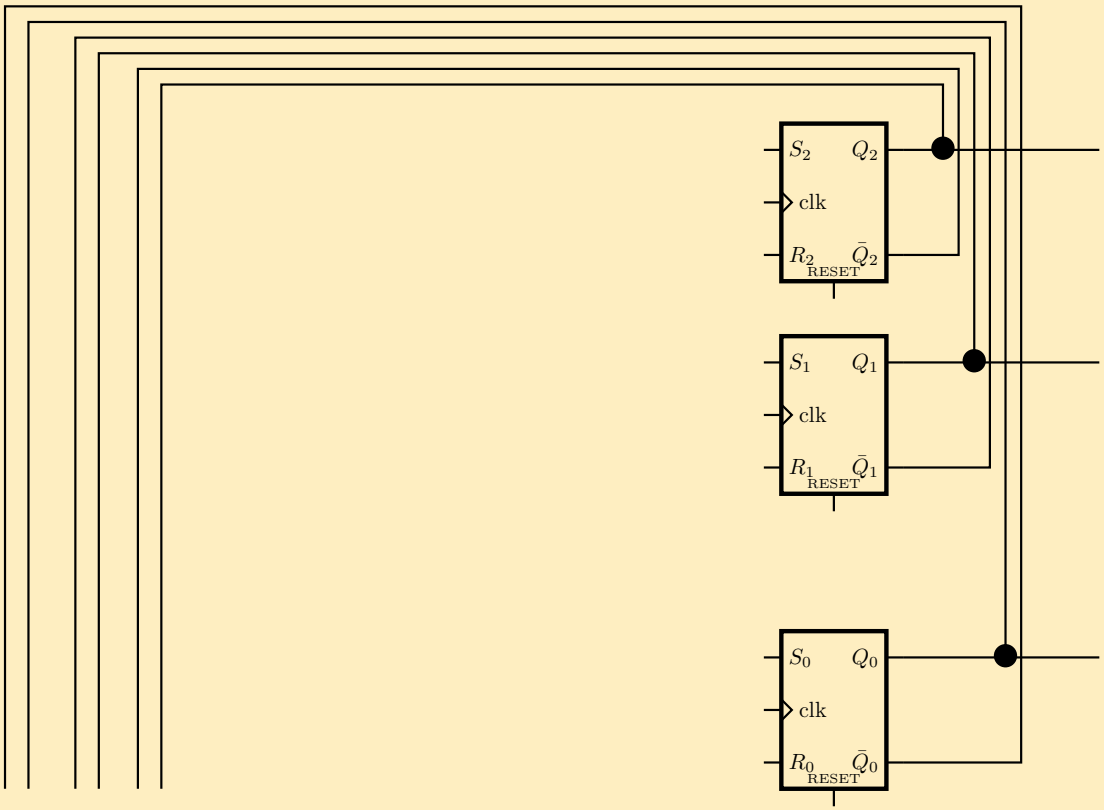
3-Bit Gray Counter (SR-FF Circuits)



3-Bit Gray Counter (SR-FF Circuits)

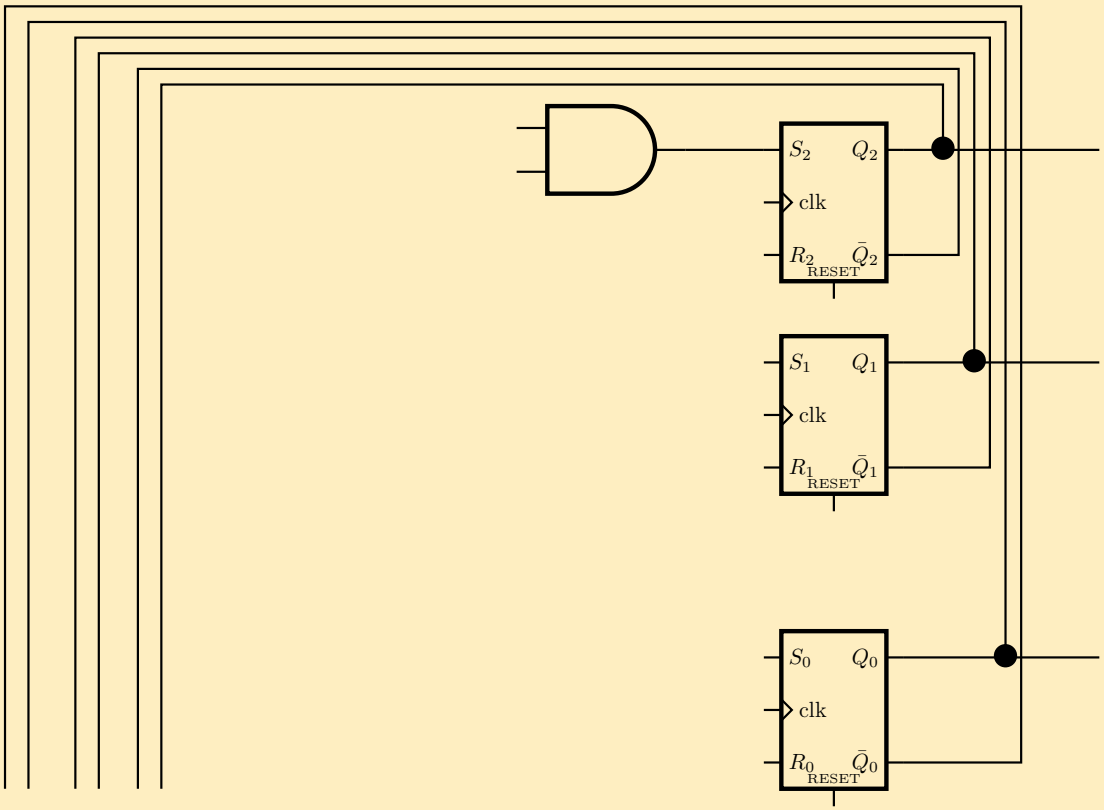


3-Bit Gray Counter (SR-FF Circuits)



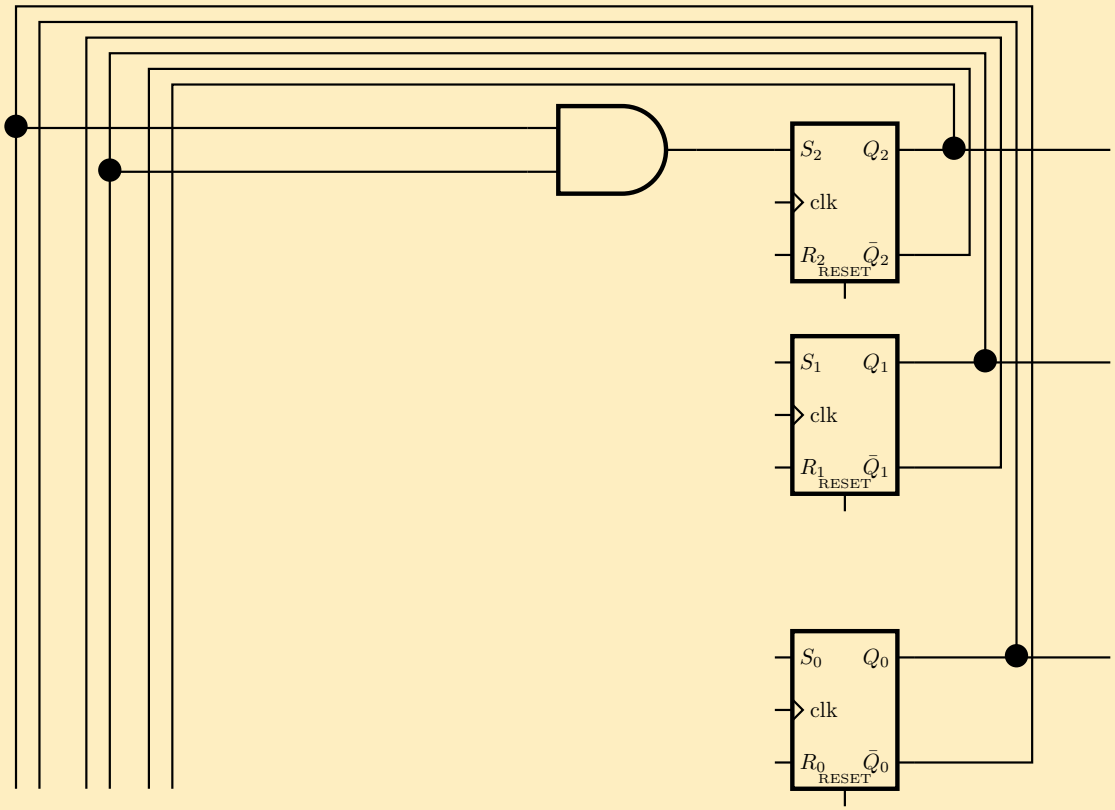
$$S_2 = Q_1\bar{Q}_0$$

3-Bit Gray Counter (SR-FF Circuits)



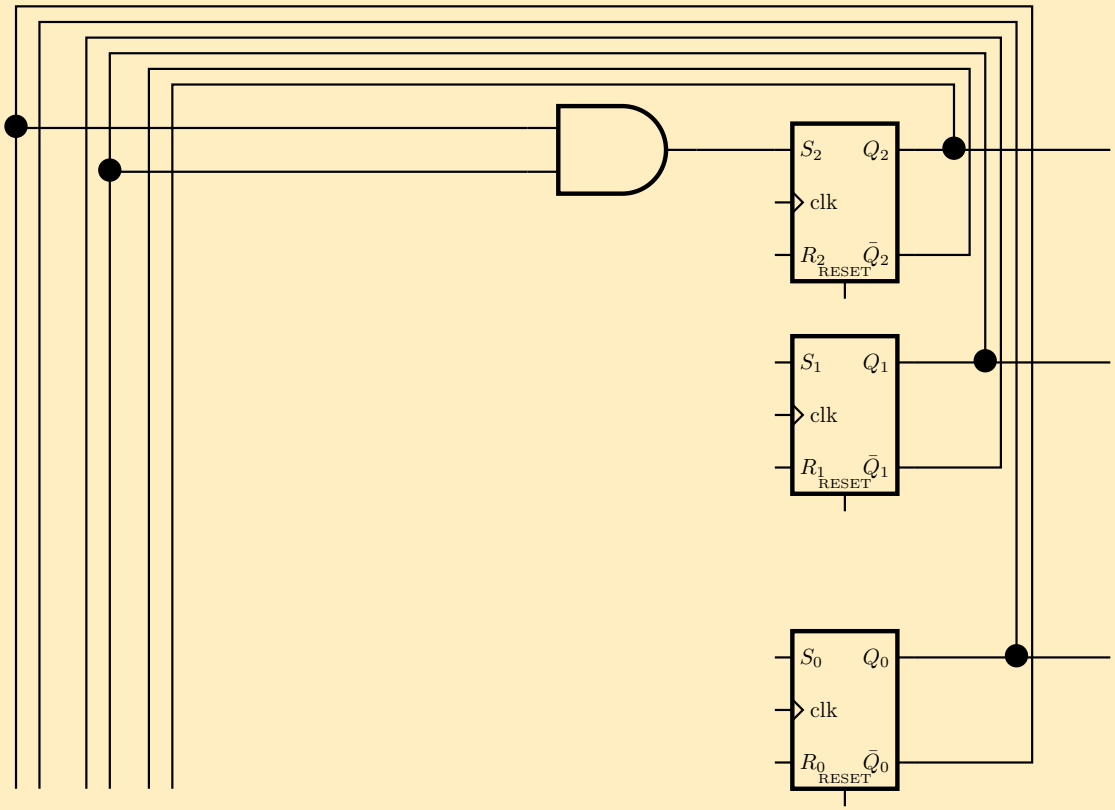
$$S_2 = Q_1\bar{Q}_0$$

3-Bit Gray Counter (SR-FF Circuits)



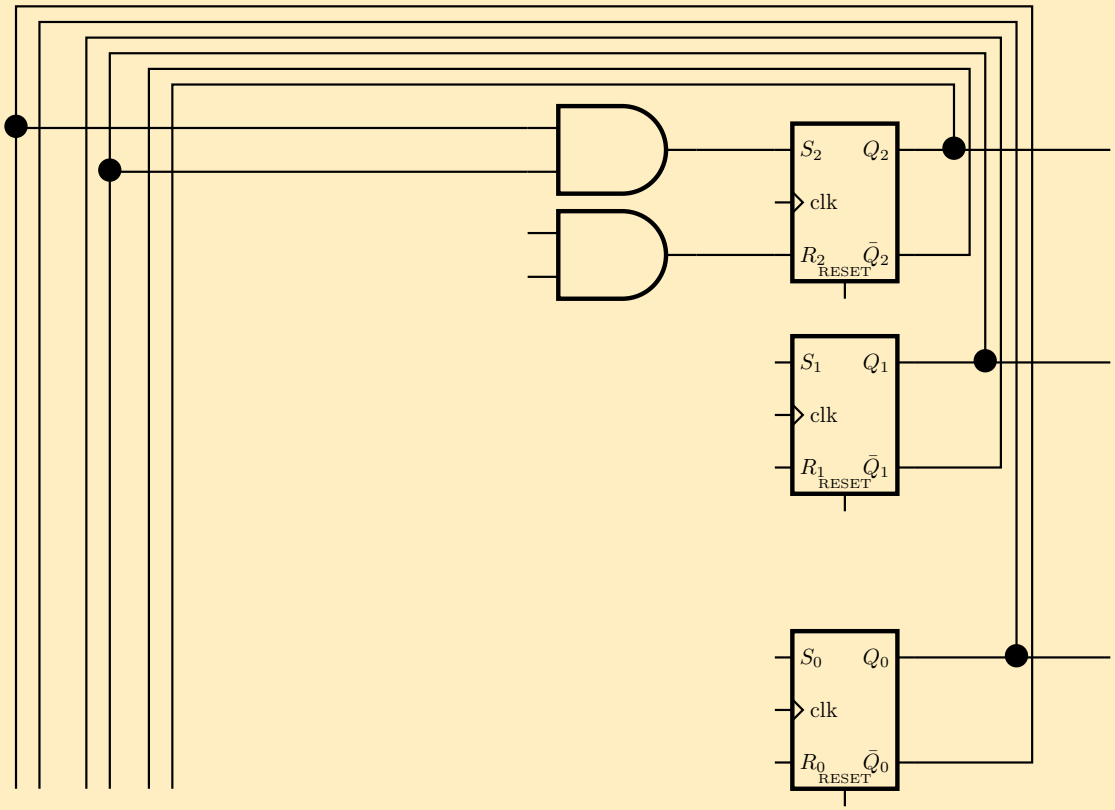
$$S_2 = Q_1\bar{Q}_0$$

3-Bit Gray Counter (SR-FF Circuits)



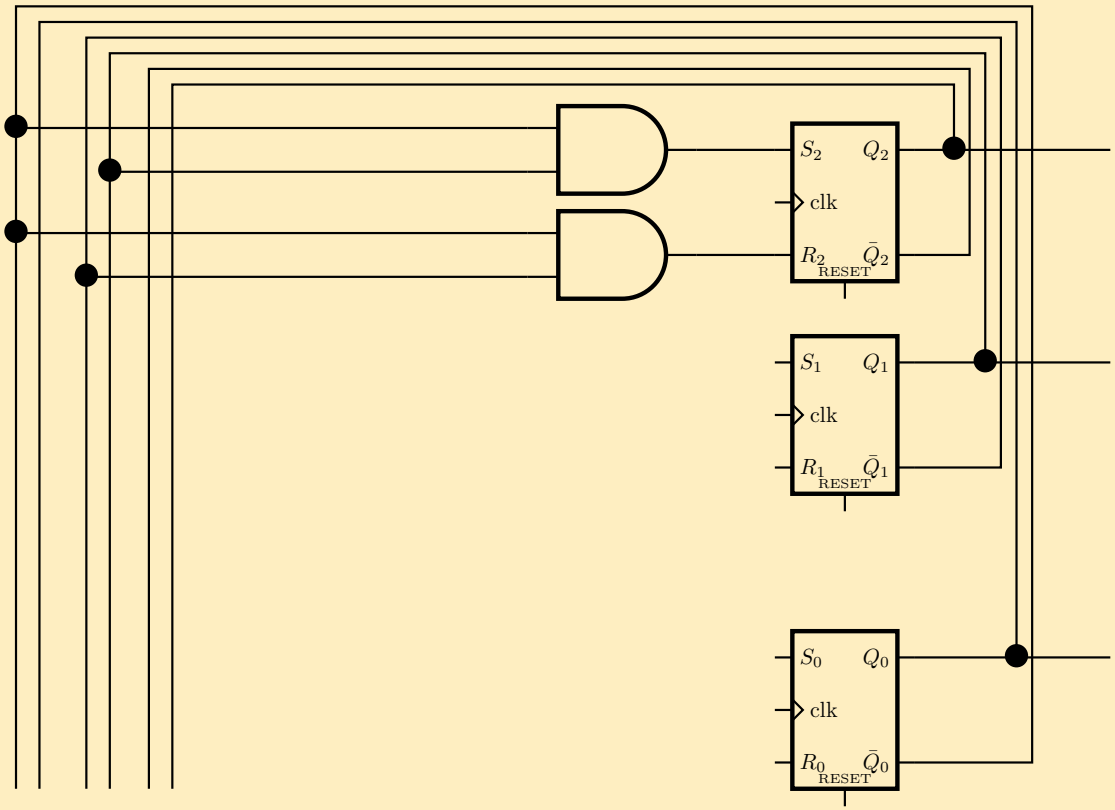
$$R_2 = \bar{Q}_1\bar{Q}_0$$

3-Bit Gray Counter (SR-FF Circuits)



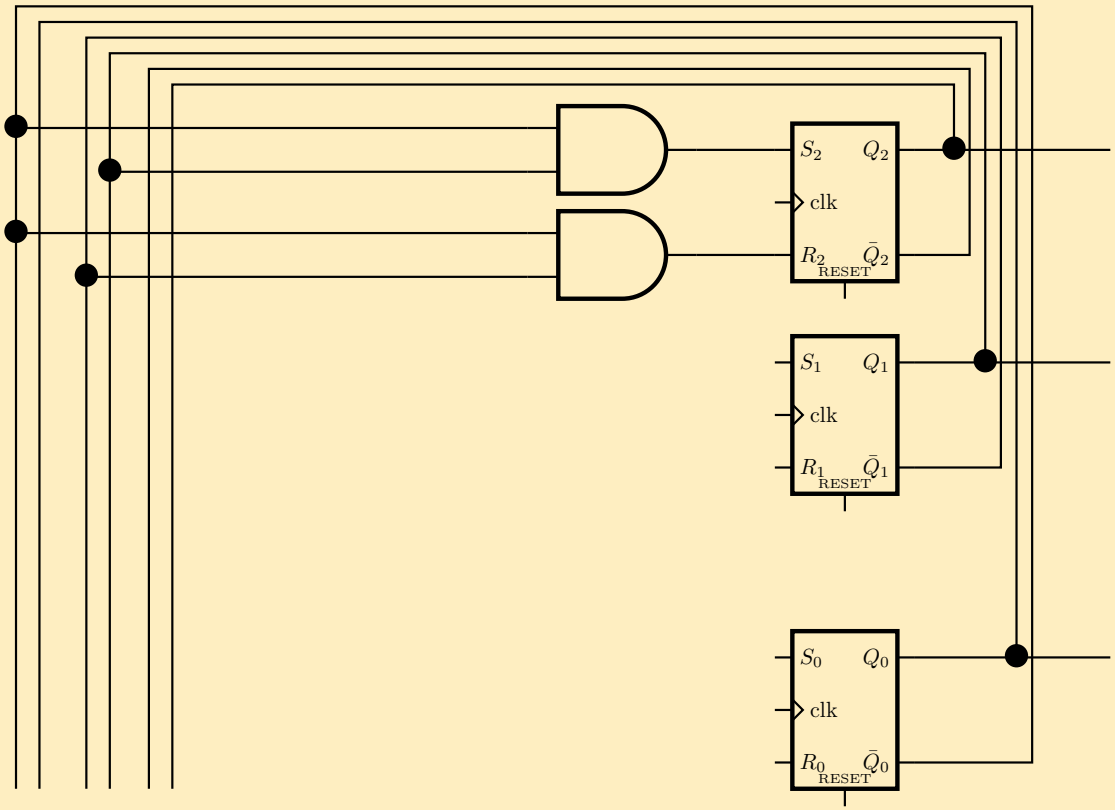
$$R_2 = \bar{Q}_1\bar{Q}_0$$

3-Bit Gray Counter (SR-FF Circuits)



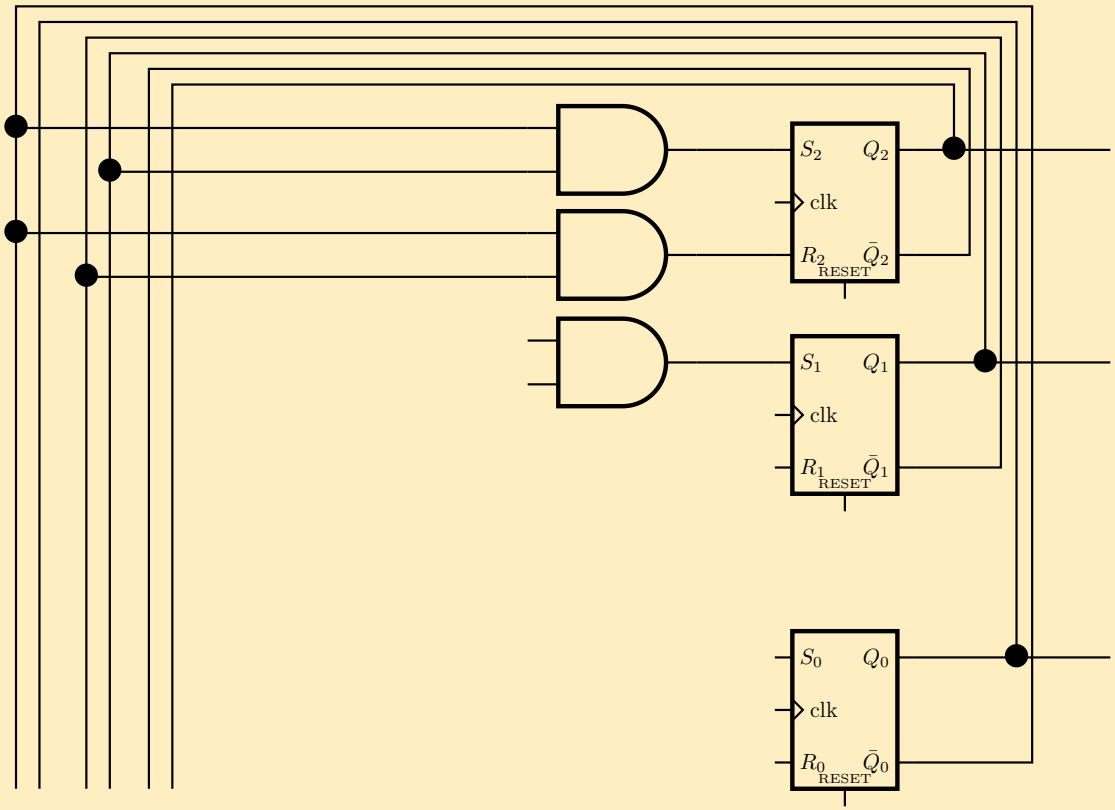
$$R_2 = \bar{Q}_1\bar{Q}_0$$

3-Bit Gray Counter (SR-FF Circuits)



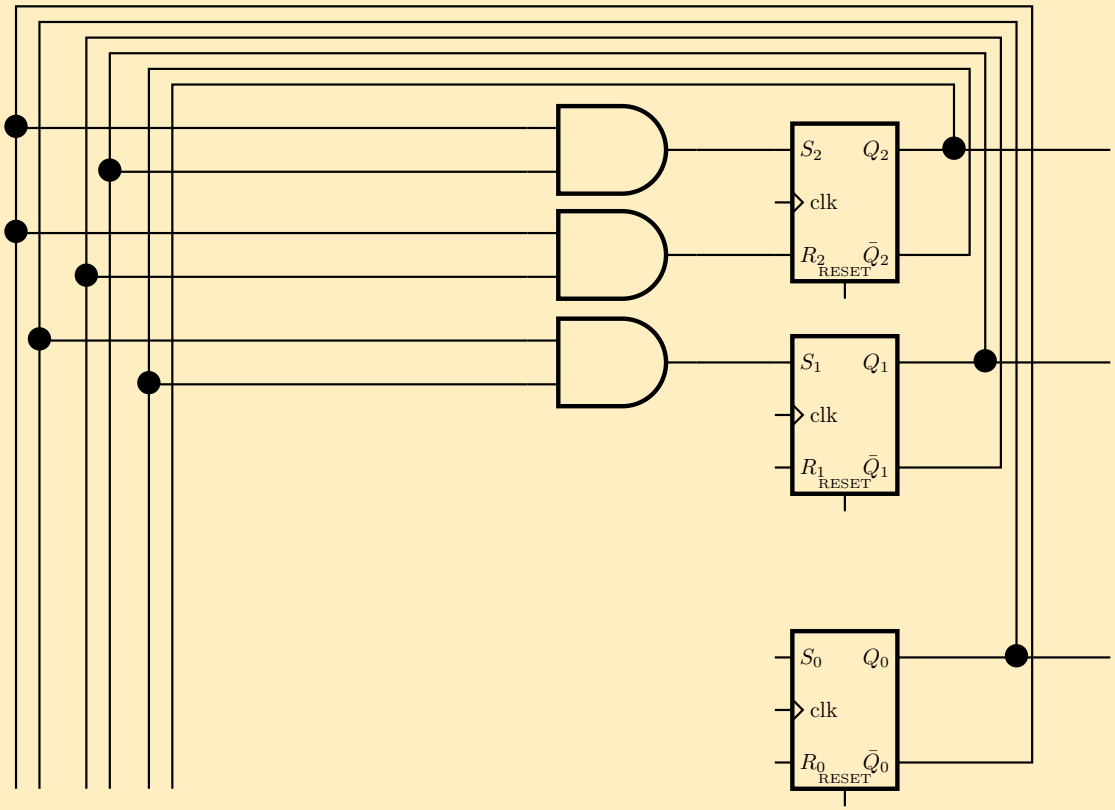
$$S_1 = \bar{Q}_2 Q_0$$

3-Bit Gray Counter (SR-FF Circuits)



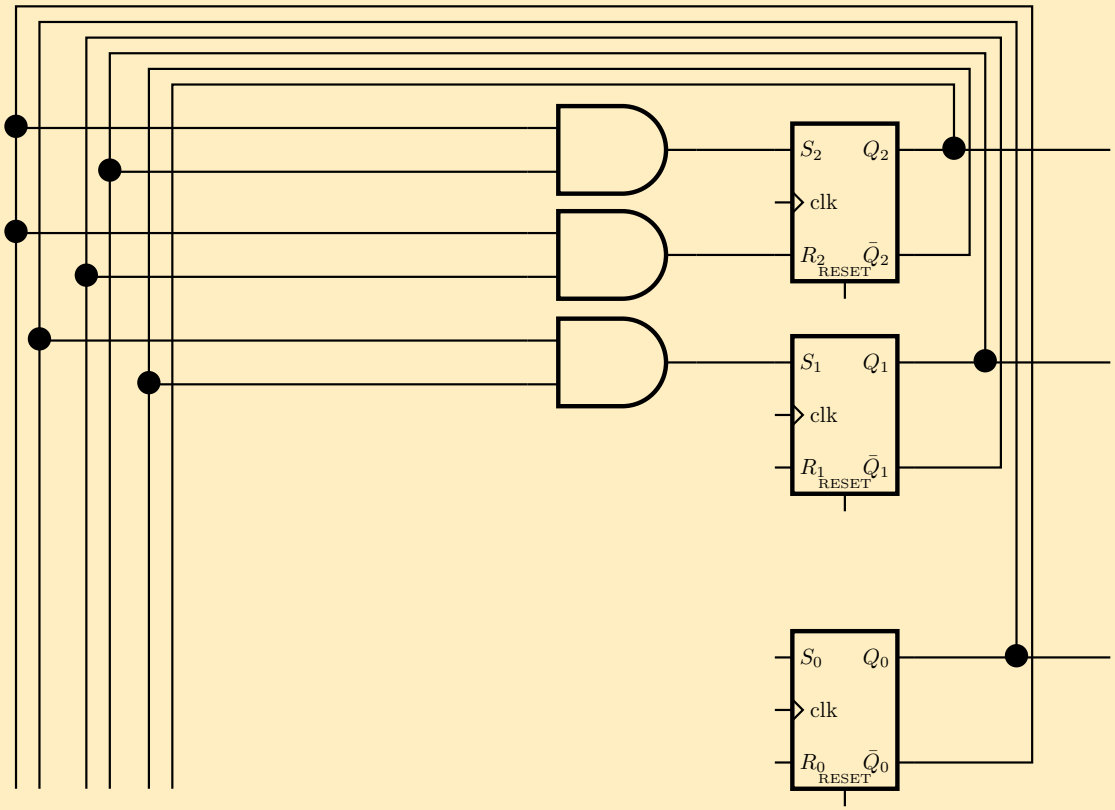
$$S_1 = \bar{Q}_2 Q_0$$

3-Bit Gray Counter (SR-FF Circuits)



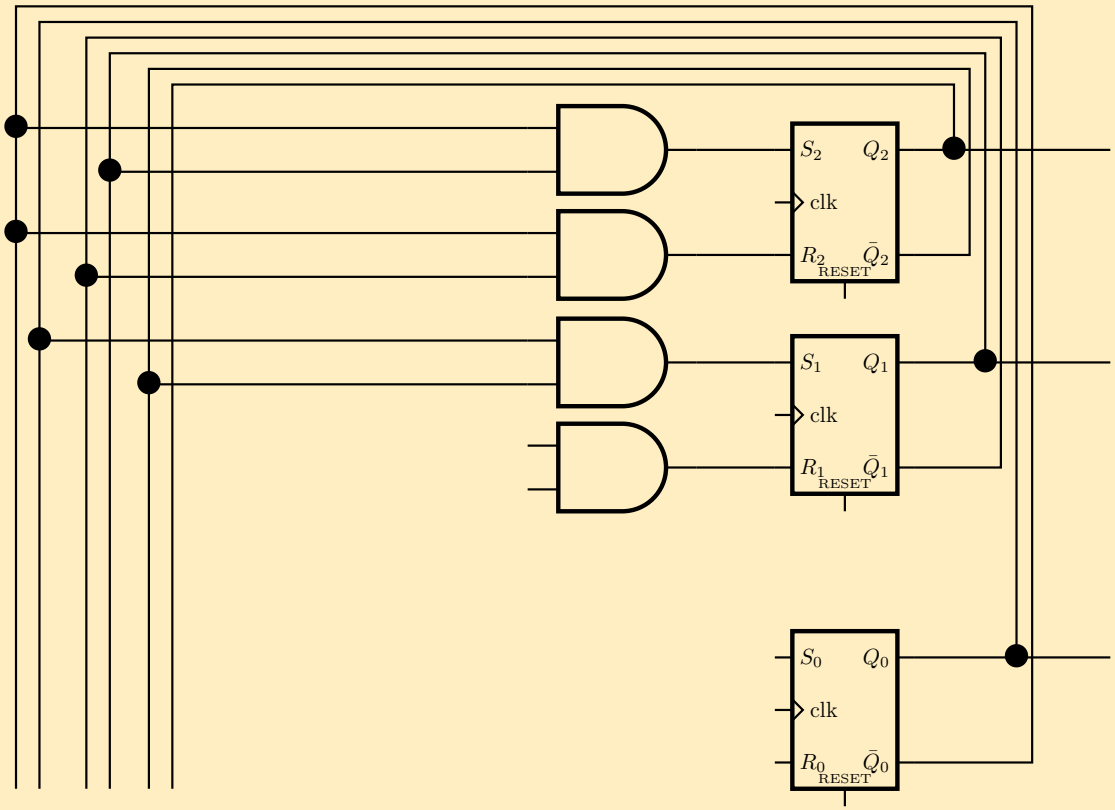
$$S_1 = \bar{Q}_2 Q_0$$

3-Bit Gray Counter (SR-FF Circuits)



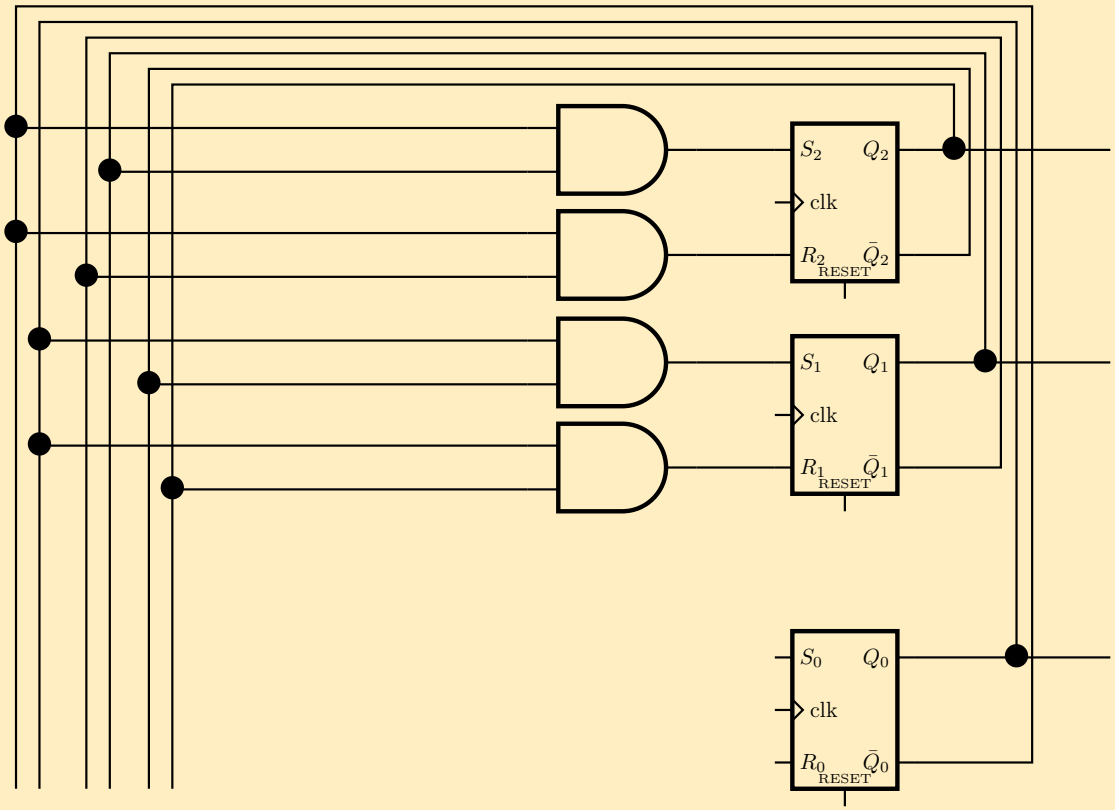
$$R_1 = Q_2 Q_0$$

3-Bit Gray Counter (SR-FF Circuits)



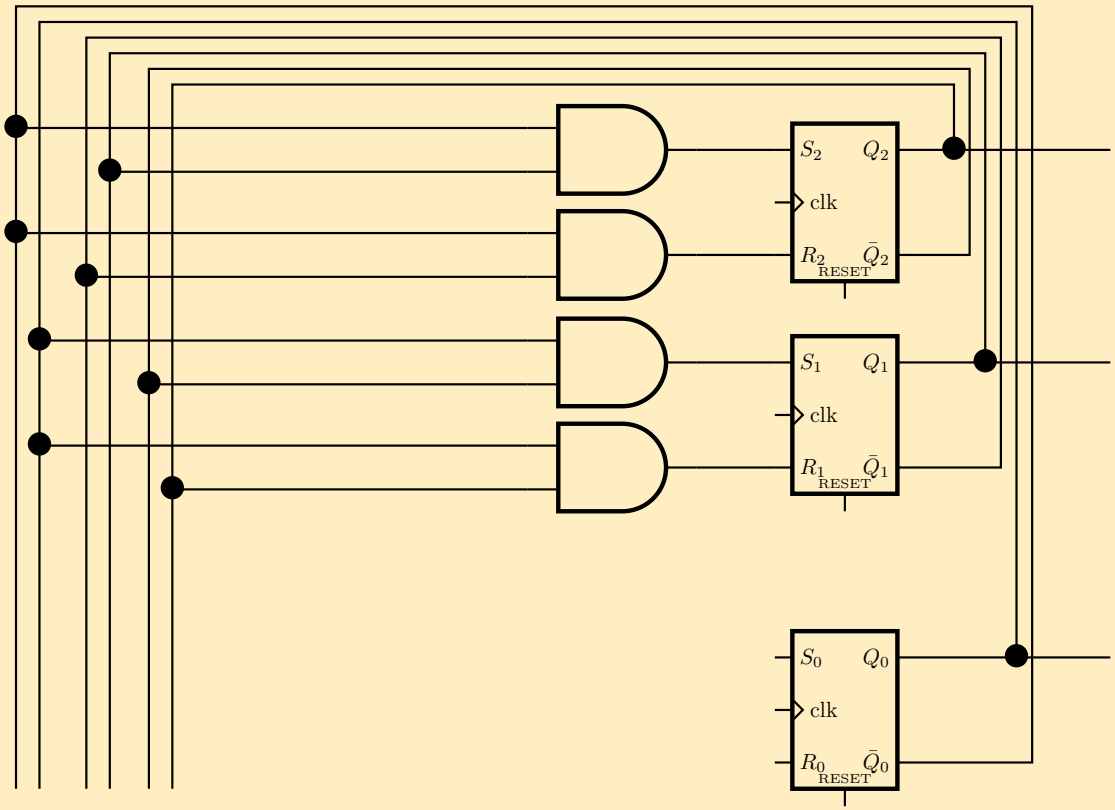
$$R_1 = Q_2 Q_0$$

3-Bit Gray Counter (SR-FF Circuits)



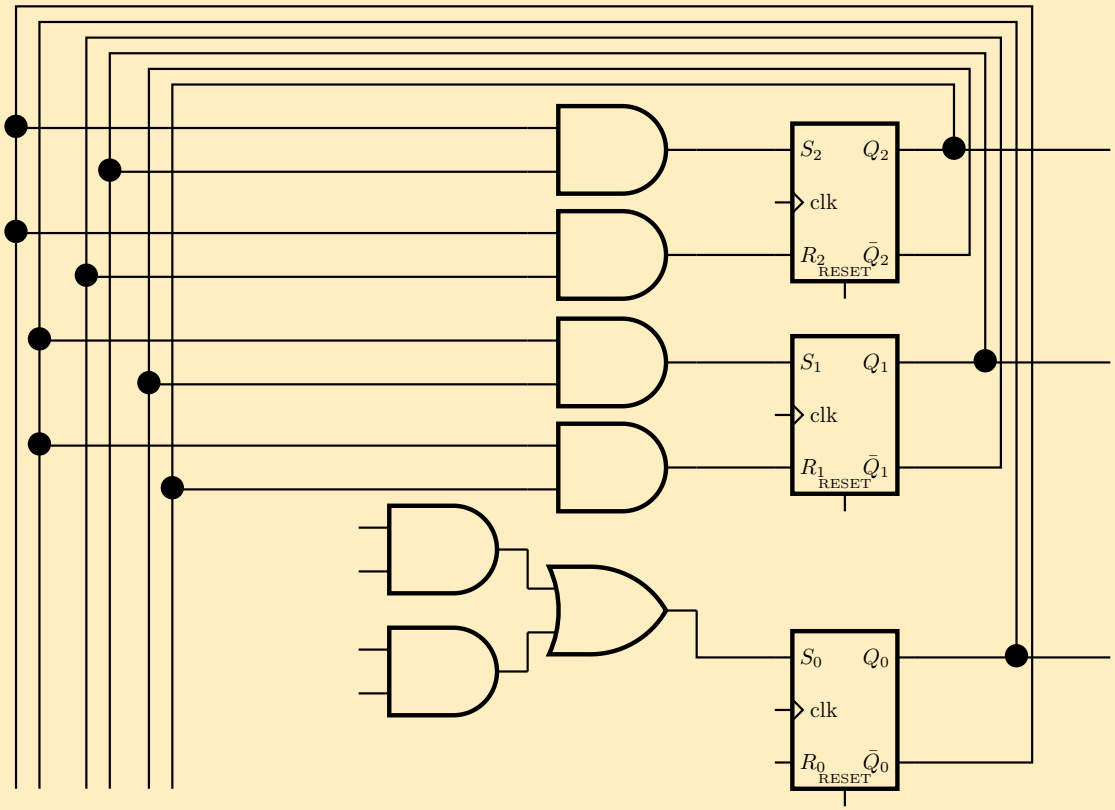
$$R_1 = Q_2 Q_0$$

3-Bit Gray Counter (SR-FF Circuits)



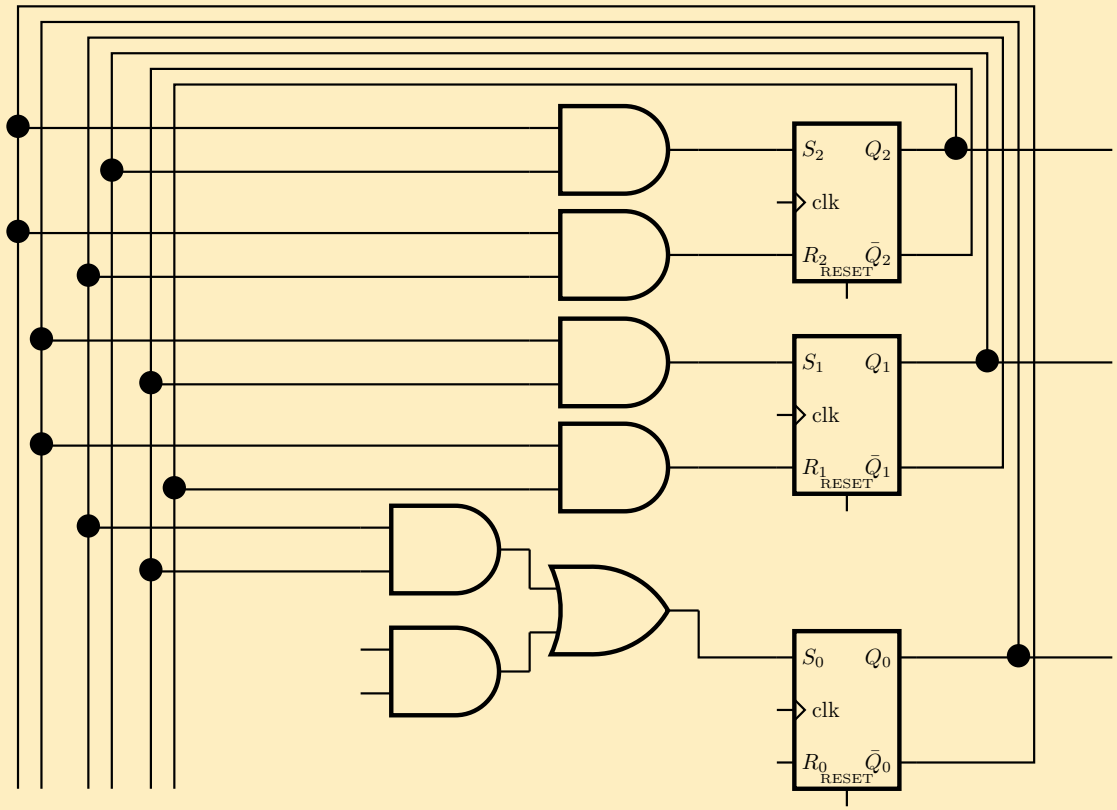
$$S_0 = \bar{Q}_2\bar{Q}_1 + Q_2Q_1$$

3-Bit Gray Counter (SR-FF Circuits)



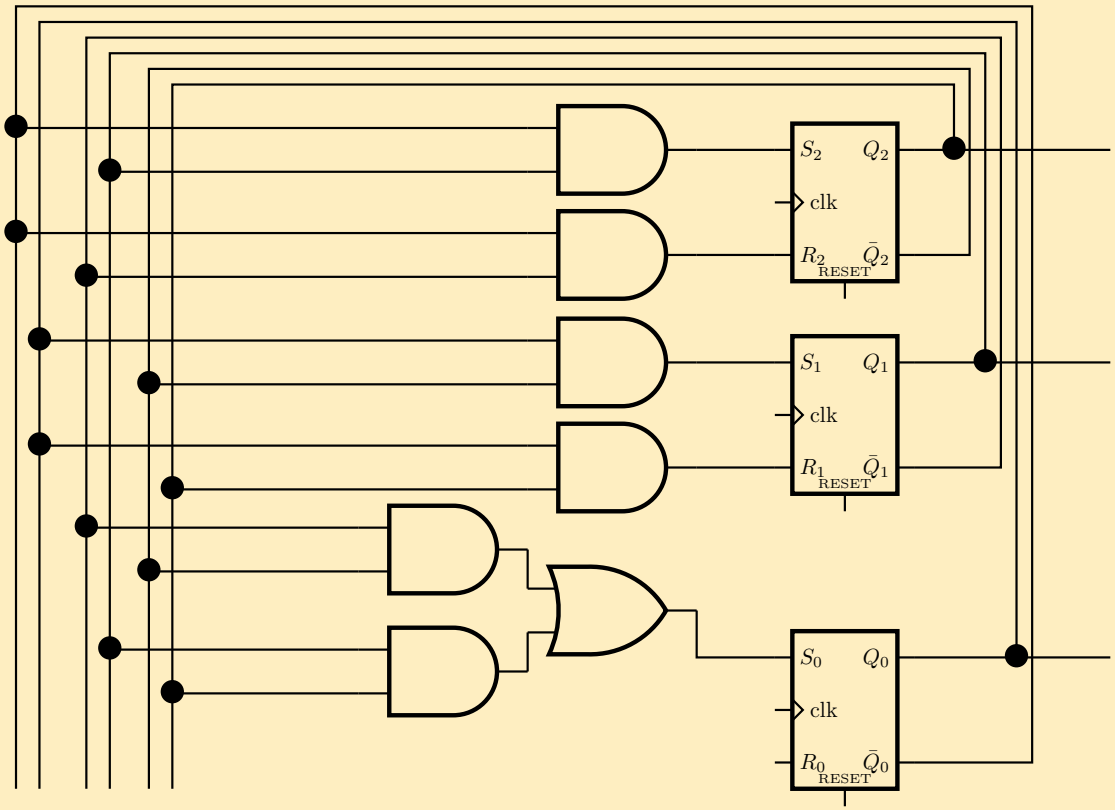
$$S_0 = \bar{Q}_2\bar{Q}_1 + Q_2Q_1$$

3-Bit Gray Counter (SR-FF Circuits)



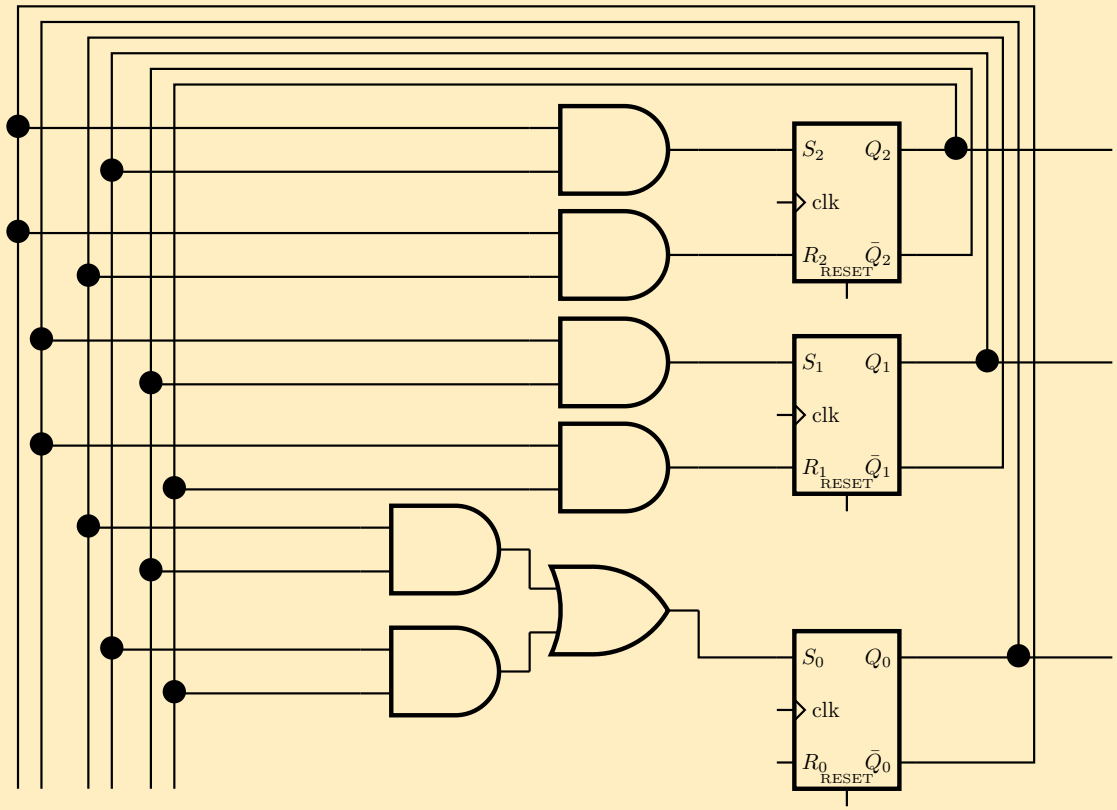
$$S_0 = \bar{Q}_2\bar{Q}_1 + Q_2Q_1$$

3-Bit Gray Counter (SR-FF Circuits)



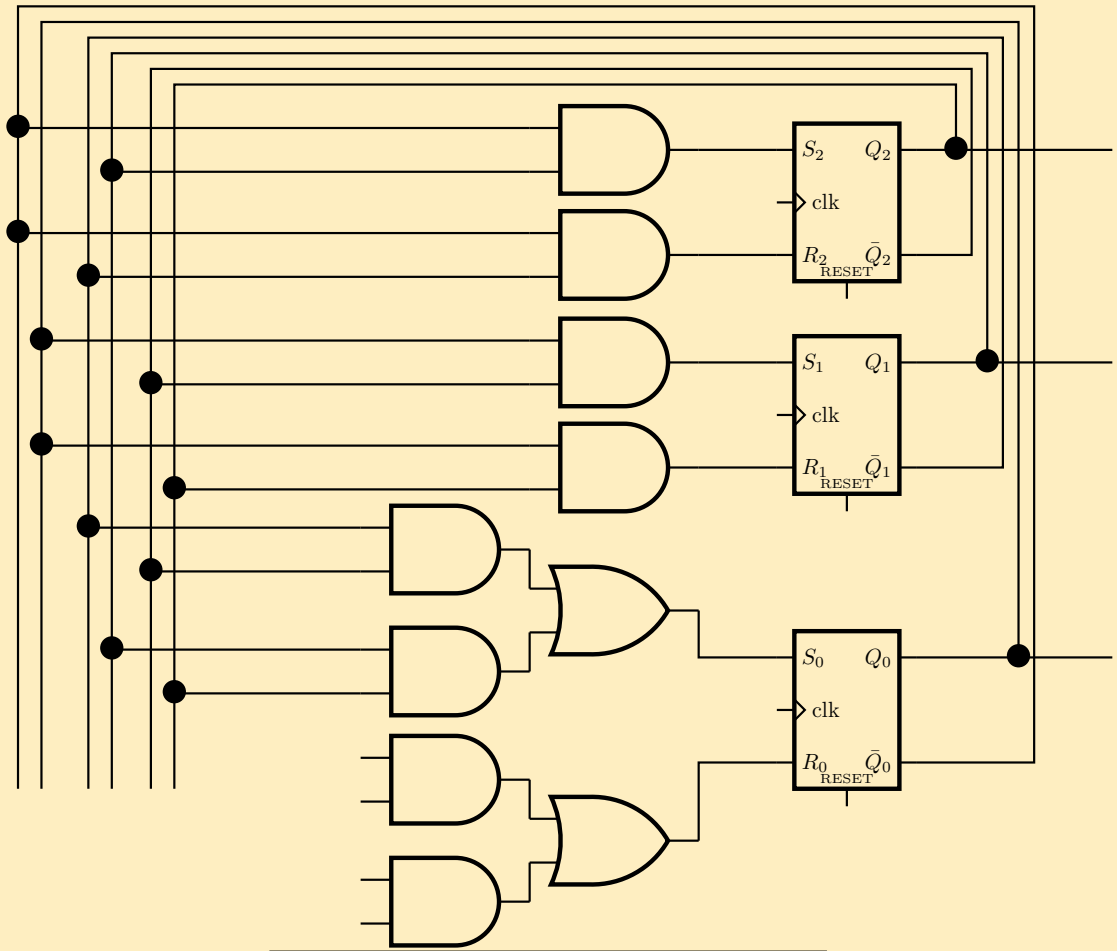
$$S_0 = \bar{Q}_2\bar{Q}_1 + Q_2Q_1$$

3-Bit Gray Counter (SR-FF Circuits)



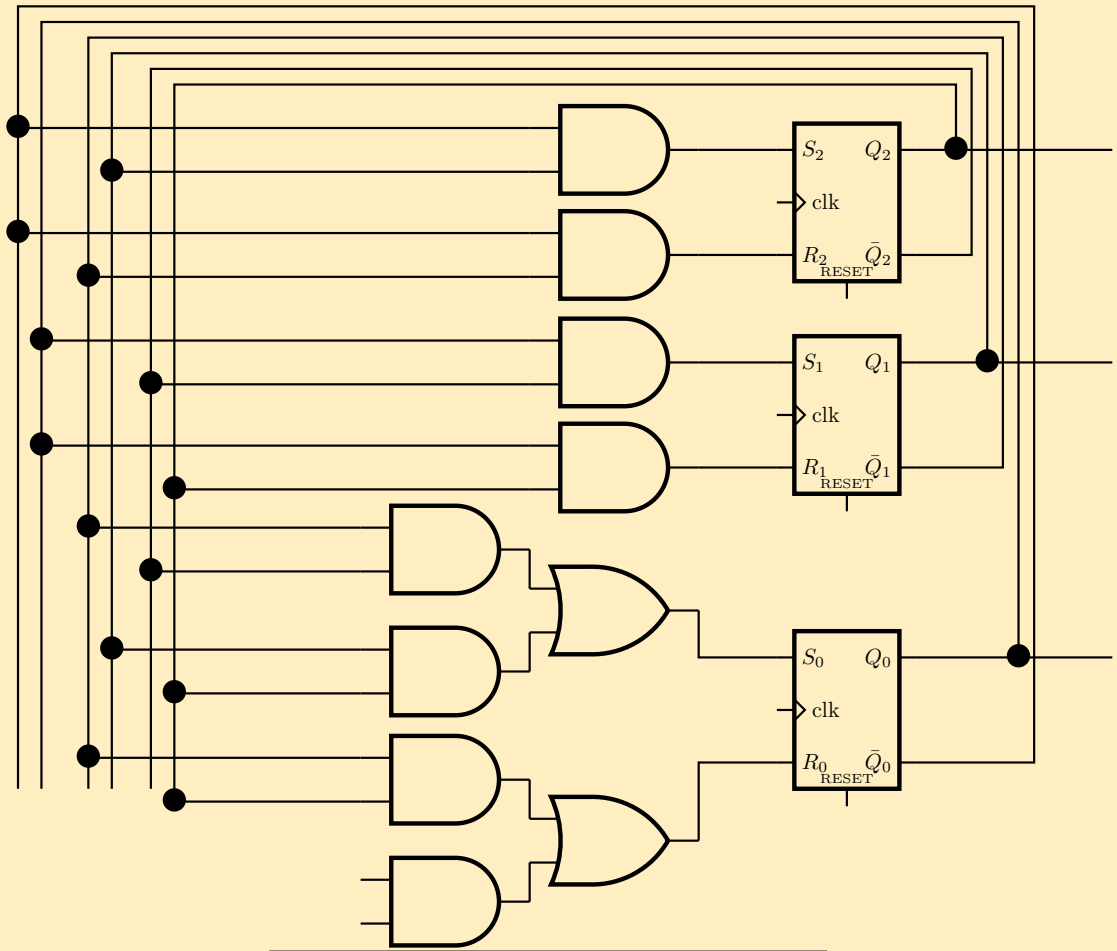
$$R_0 = \bar{Q}_2 Q_1 + Q_2 \bar{Q}_1$$

3-Bit Gray Counter (SR-FF Circuits)



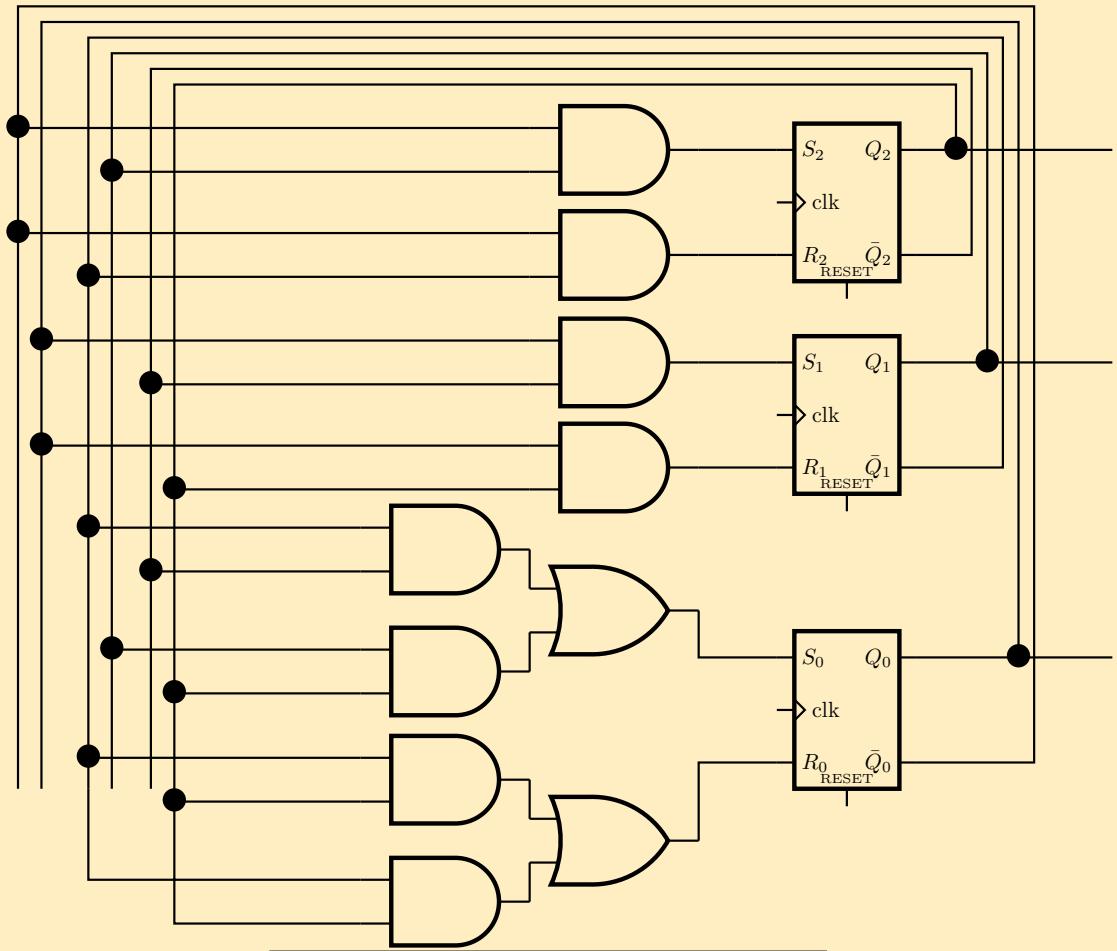
$$R_0 = \bar{Q}_2 Q_1 + Q_2 \bar{Q}_1$$

3-Bit Gray Counter (SR-FF Circuits)



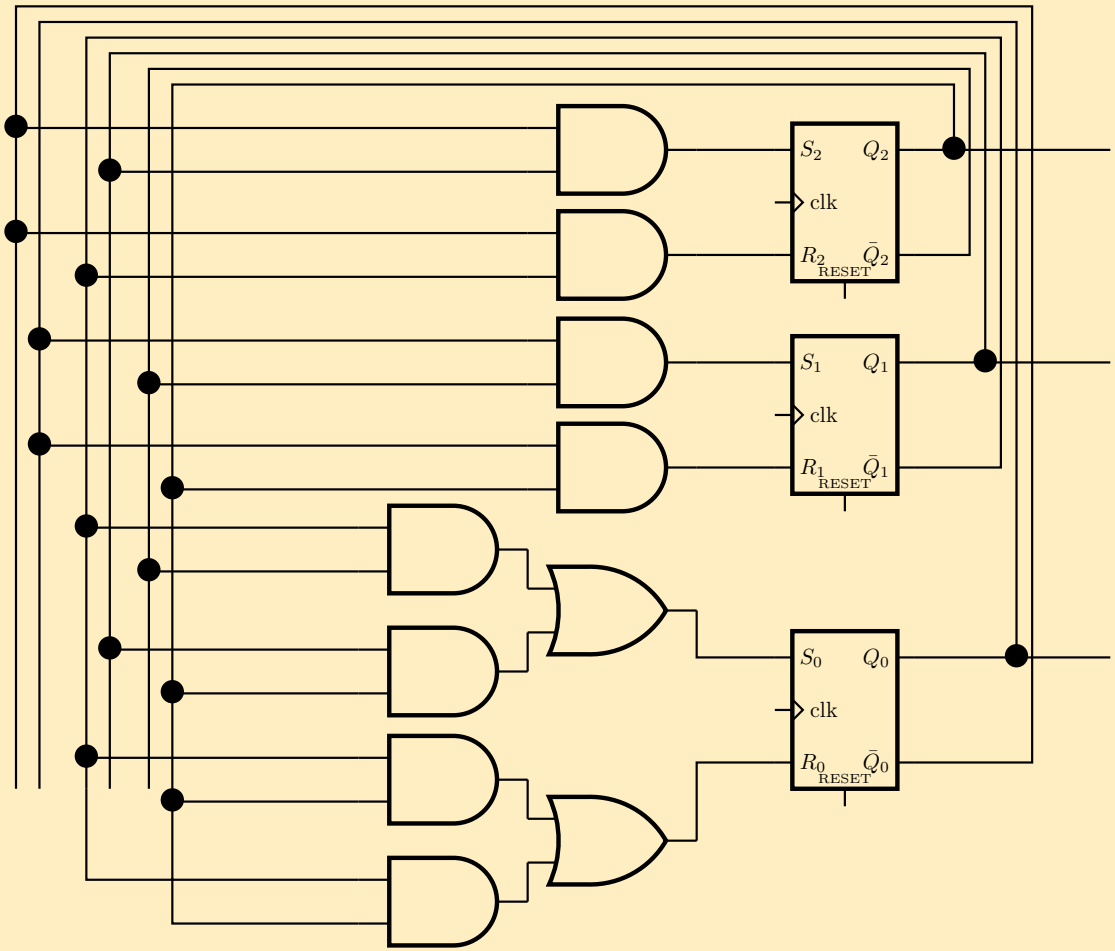
$$R_0 = \bar{Q}_2 Q_1 + Q_2 \bar{Q}_1$$

3-Bit Gray Counter (SR-FF Circuits)



$$R_0 = \bar{Q}_2 Q_1 + Q_2 \bar{Q}_1$$

3-Bit Gray Counter (SR-FF Circuits)



3-Bits Gray Counter, SR-FF I

Listing 15: demo/08_gray3sr/gray3.v

```
`timescale 1ns/1ns

module gray3(
  output x2,x1,x0,
  input init,
  input clk
);
  wire q2,q2n,q1,q1n,q0,q0n;
  wire s2,r2,s1,r1,s0,r0;

  andGate as2(s2,q1,q0n);
  andGate ar2(r2,q1n,q0n);
```

3-Bits Gray Counter, SR-FF II

```
  andGate as1(s1,q2n,q0);
  andGate ar1(r1,q2,q0);

  xnorGate as0(s0,q2,q1);
  xorGate  ar0(r0,q2,q1);

  srffi srff0(q2,q2n,s2,r2,1'b0,init,clk);
  srffi srff1(q1,q1n,s1,r1,1'b0,init,clk);
  srffi srff2(q0,q0n,s0,r0,1'b0,init,clk);

  assign x2 = q2;
  assign x1 = q1;
  assign x0 = q0;

endmodule
```

3-Bits Gray Counter, SR-FF III

3-Bits Gray Counter, SR-FF I

Listing 16: demo/08_gray3sr/gray3_tb.v

```
`timescale 1ns/1ns

module gray3_tb;

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

  logic init;
  wire x2, x1, x0;
  gray3 g(x2, x1, x0, init, clk);

  initial begin
```

3-Bits Gray Counter, SR-FF II


```
    $dumpfile("gray3.vcd");
    $dumpvars;

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

    #2000ns;
    $finish;
  end

endmodule
```

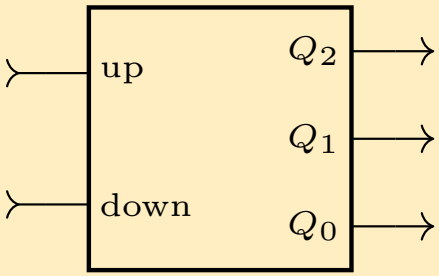
(Carmi) Lecture 16 reached here

- We need to repeat the gray 3 bits variants

3-Bit Gray Counter with Up/Down Controls

(1-hot D-FFs)

3-Bit Gray Counter with Up/Down (1-hot)

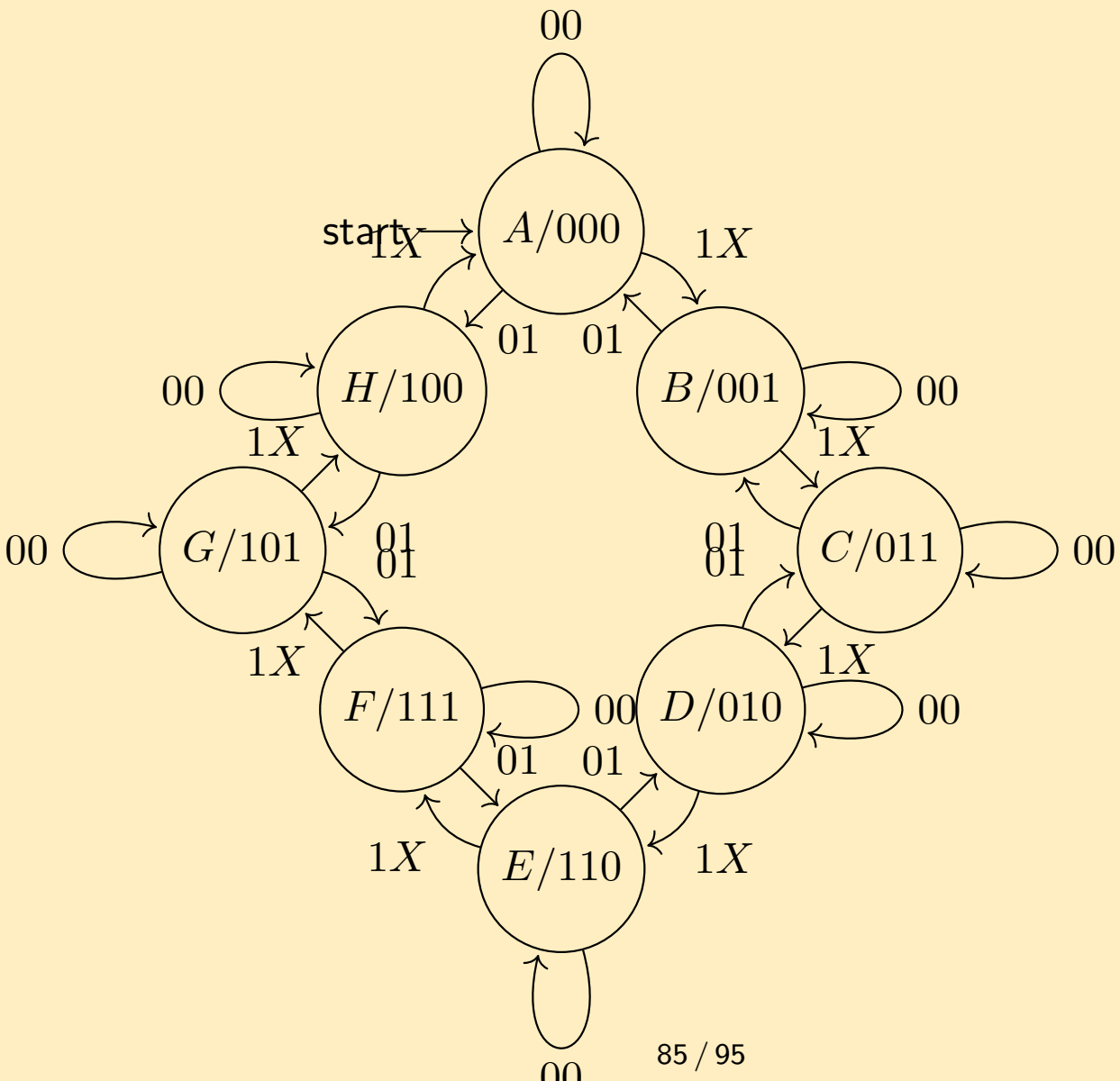


The Controls

The control lines are up and down

up	down	Behavior
1	X	go up
0	1	go down
0	0	stay

3-Bit Gray Counter with Up/Down (1-hot)



State Codes

State	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
A	0	0	0	0	0	0	0	1
B	0	0	0	0	0	0	1	0
C	0	0	0	0	1	0	0	0
D	0	0	0	0	0	1	0	0
E	0	1	0	0	0	0	0	0
F	1	0	0	0	0	0	0	0
G	0	0	1	0	0	0	0	0
H	0	0	0	1	0	0	0	0

Recall for D-FF, $Q(t+1) = D$

3-Bits Gray Counter, Equations (1-hot)

Rather trivial

$$D_0 = \overline{\text{up}} \cdot \overline{\text{down}} \cdot Q_0 + \text{up} \cdot Q_4 + \overline{\text{up}} \cdot \text{down} \cdot Q_1$$

$$D_1 = \overline{\text{up}} \cdot \overline{\text{down}} \cdot Q_1 + \text{up} \cdot Q_0 + \overline{\text{up}} \cdot \text{down} \cdot Q_3$$

$$D_3 = \overline{\text{up}} \cdot \overline{\text{down}} \cdot Q_3 + \text{up} \cdot Q_1 + \overline{\text{up}} \cdot \text{down} \cdot Q_2$$

$$D_2 = \overline{\text{up}} \cdot \overline{\text{down}} \cdot Q_2 + \text{up} \cdot Q_3 + \overline{\text{up}} \cdot \text{down} \cdot Q_6$$

$$D_6 = \overline{\text{up}} \cdot \overline{\text{down}} \cdot Q_6 + \text{up} \cdot Q_2 + \overline{\text{up}} \cdot \text{down} \cdot Q_7$$

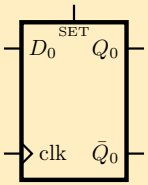
$$D_7 = \overline{\text{up}} \cdot \overline{\text{down}} \cdot Q_7 + \text{up} \cdot Q_6 + \overline{\text{up}} \cdot \text{down} \cdot Q_5$$

$$D_5 = \overline{\text{up}} \cdot \overline{\text{down}} \cdot Q_5 + \text{up} \cdot Q_7 + \overline{\text{up}} \cdot \text{down} \cdot Q_4$$

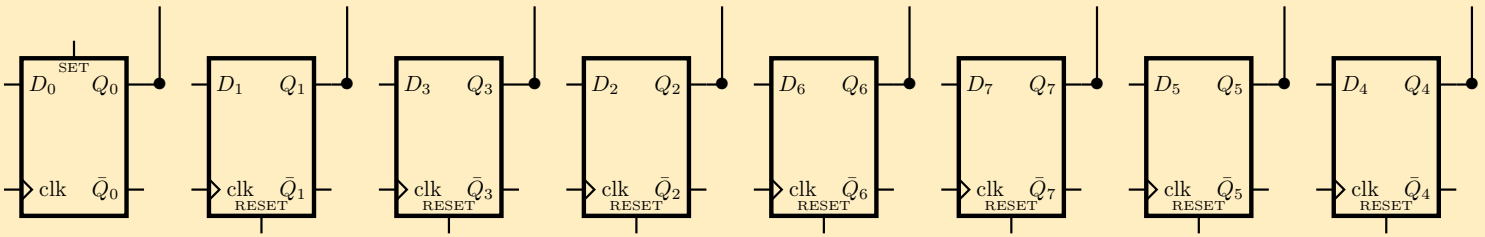
$$D_4 = \overline{\text{up}} \cdot \overline{\text{down}} \cdot Q_4 + \text{up} \cdot Q_0 + \overline{\text{up}} \cdot \text{down} \cdot Q_5$$

3-Bits Gray Counter, Circuit (1-hot)

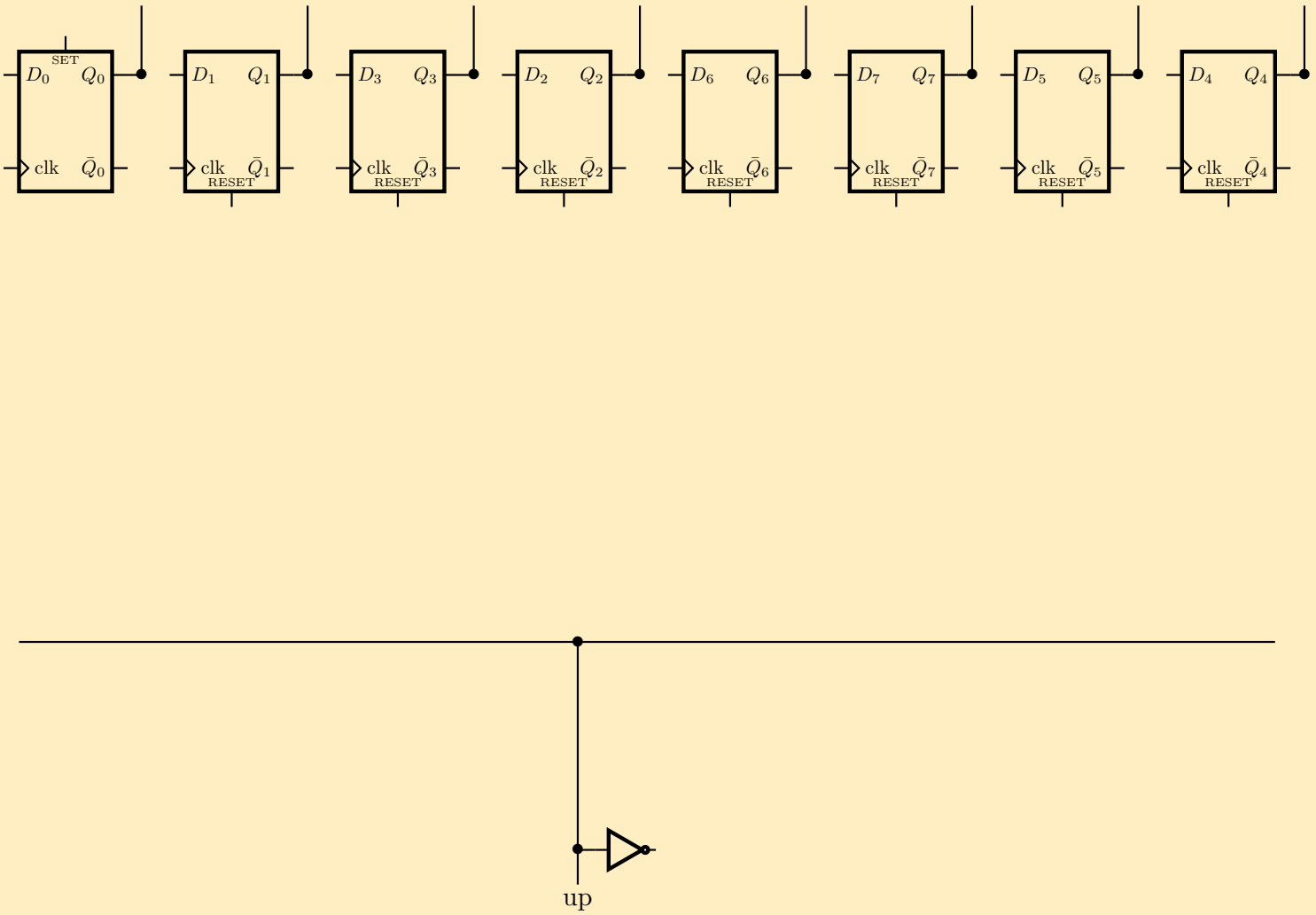
3-Bits Gray Counter, Circuit (1-hot)



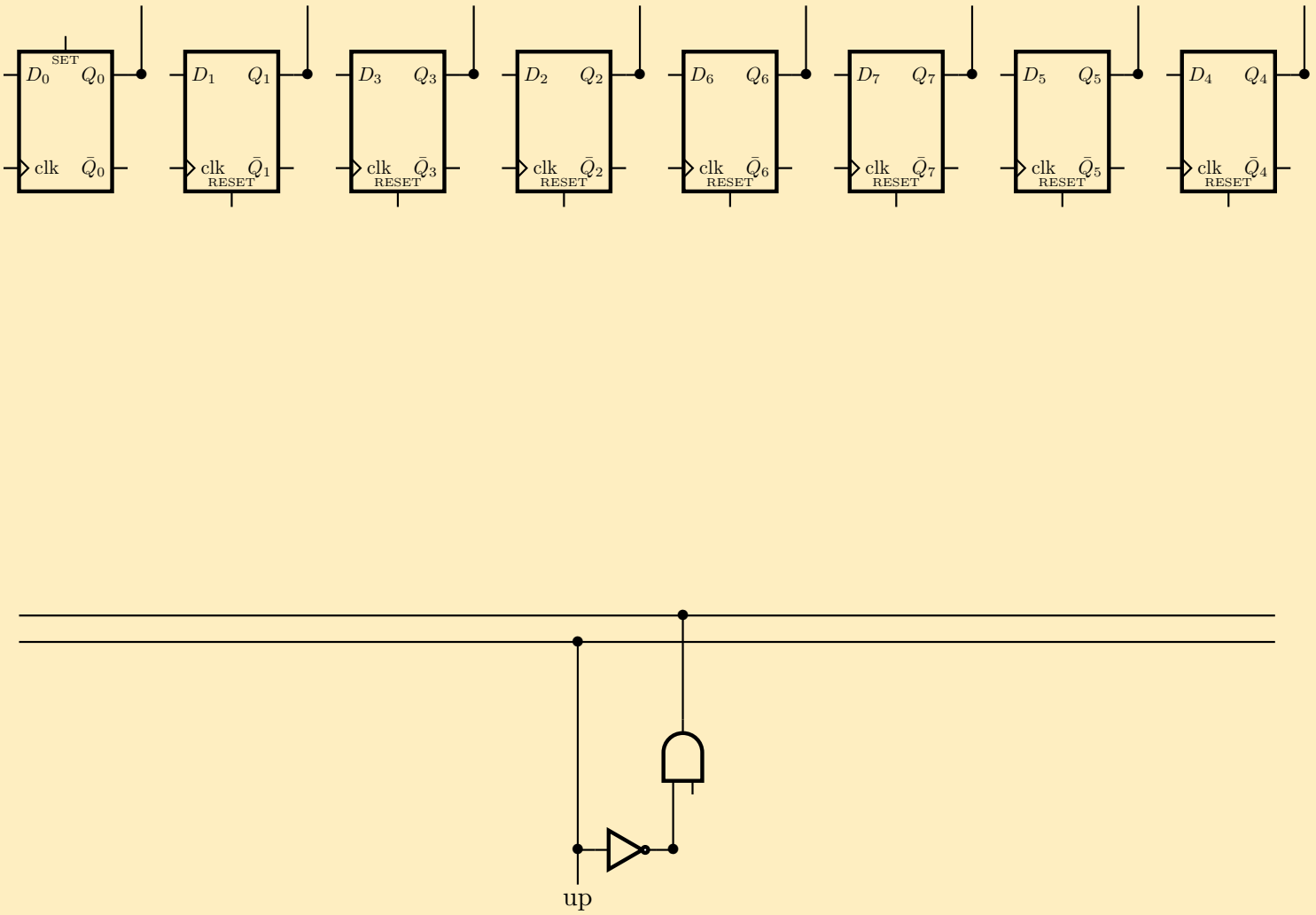
3-Bits Gray Counter, Circuit (1-hot)



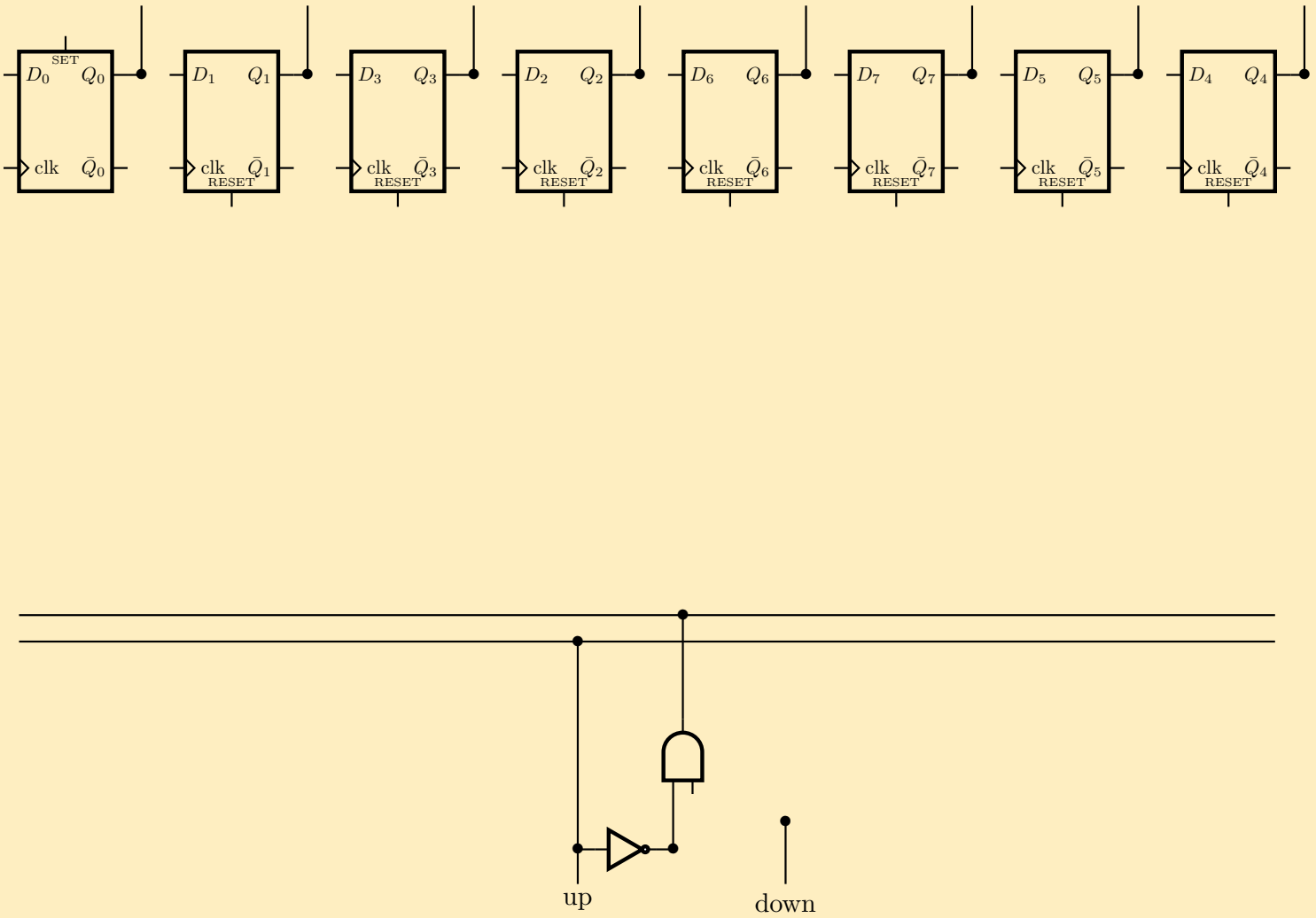
3-Bits Gray Counter, Circuit (1-hot)



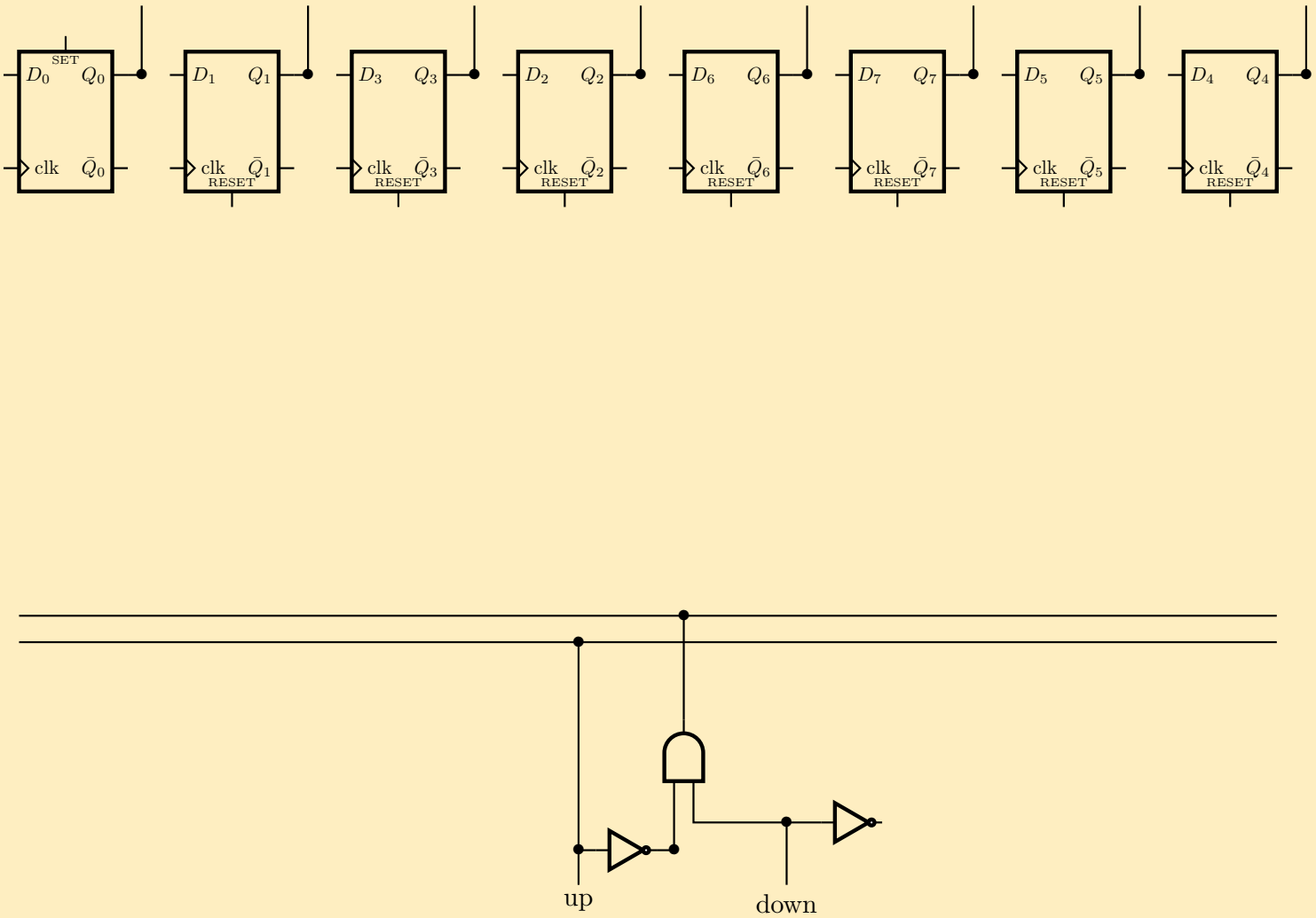
3-Bits Gray Counter, Circuit (1-hot)



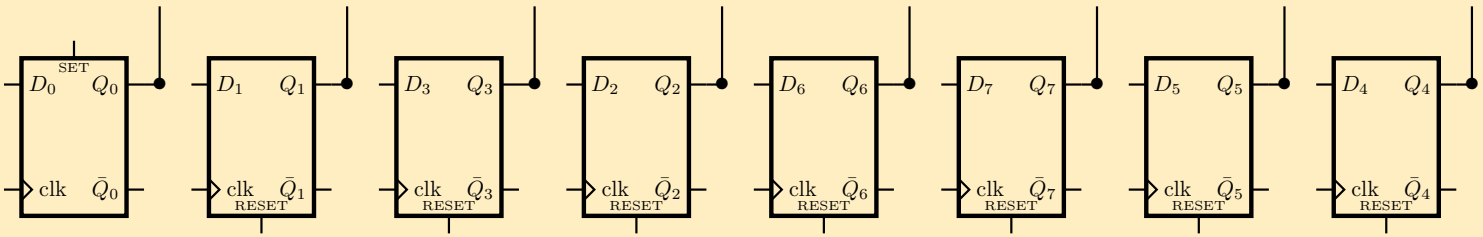
3-Bits Gray Counter, Circuit (1-hot)

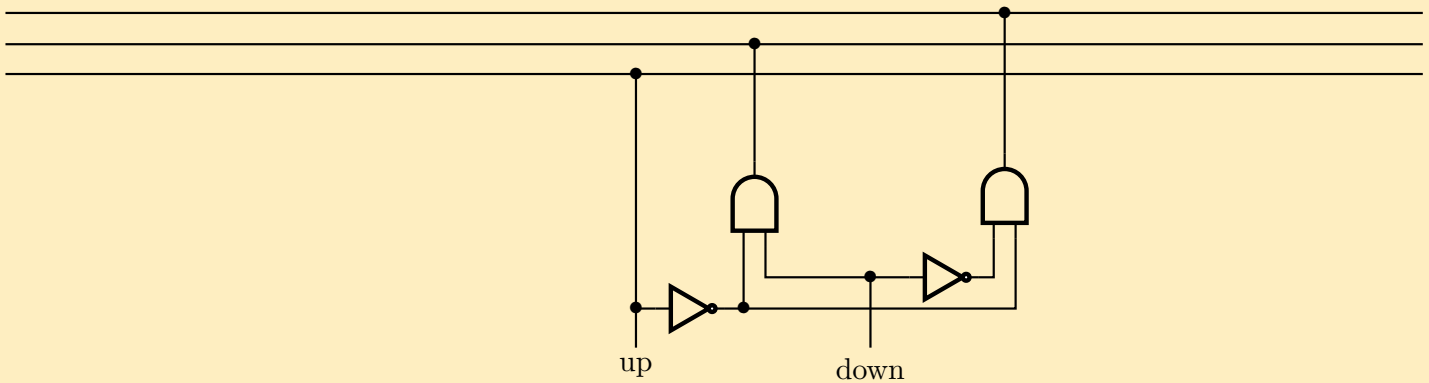


3-Bits Gray Counter, Circuit (1-hot)

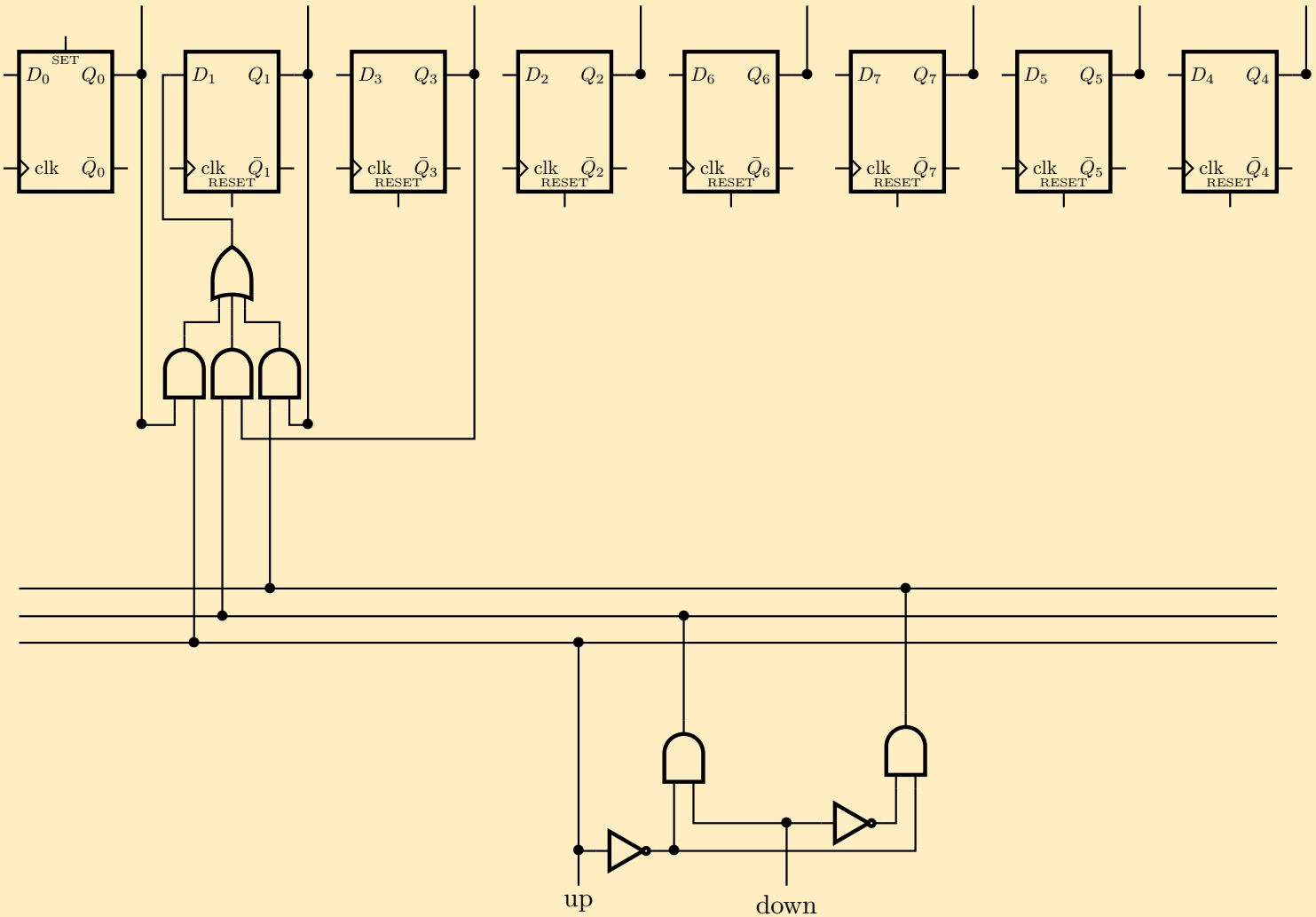


3-Bits Gray Counter, Circuit (1-hot)

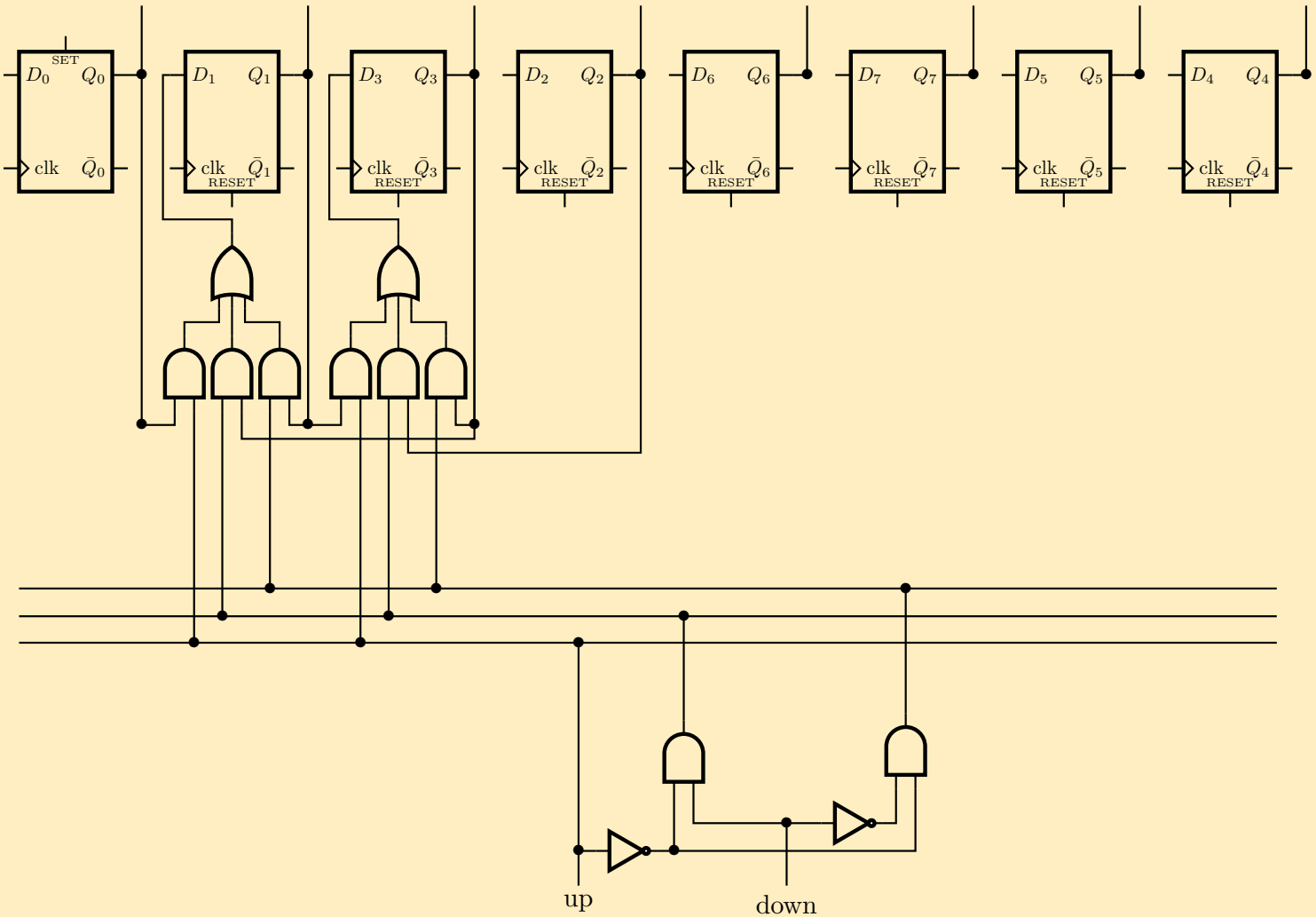




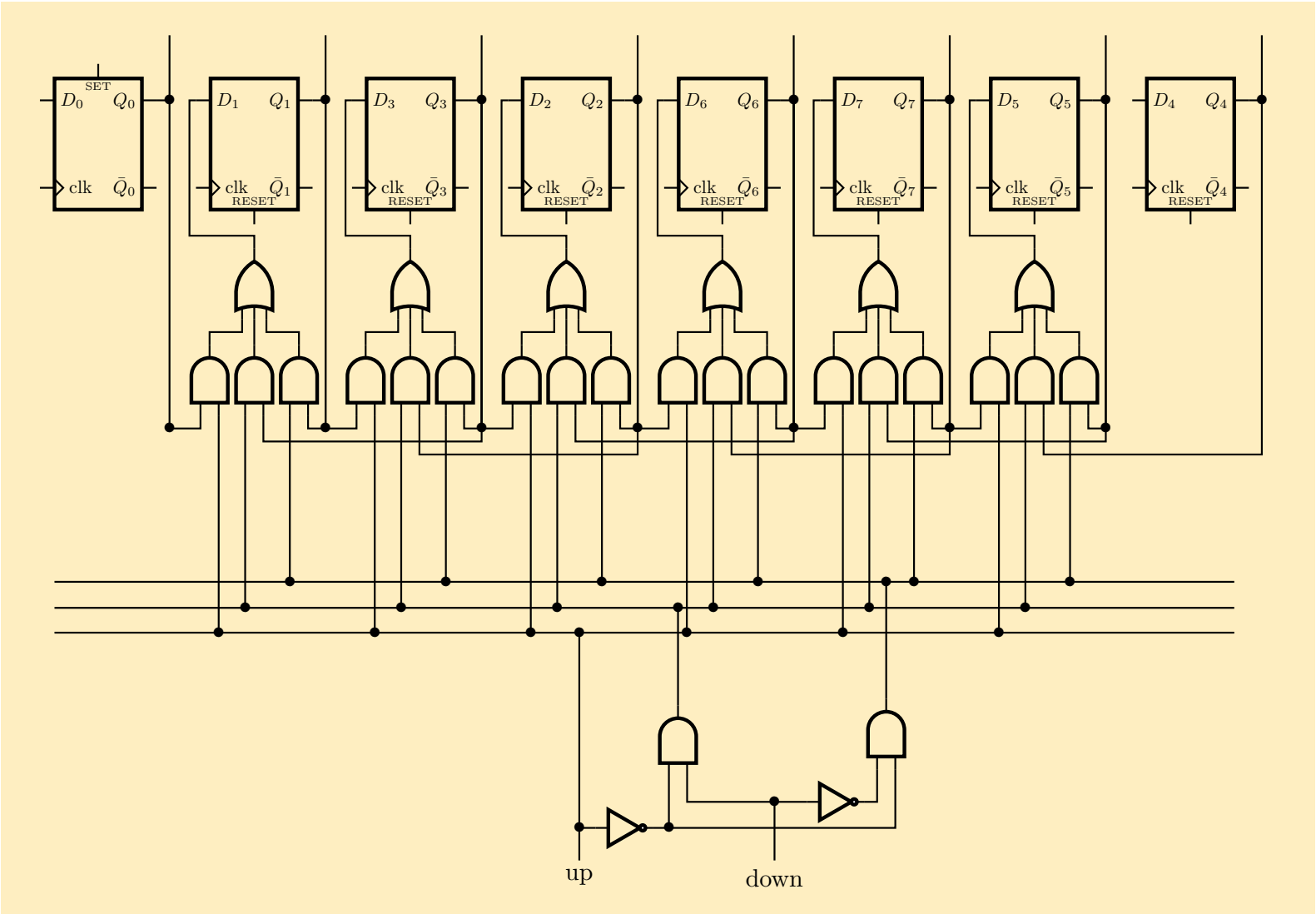
3-Bits Gray Counter, Circuit (1-hot)



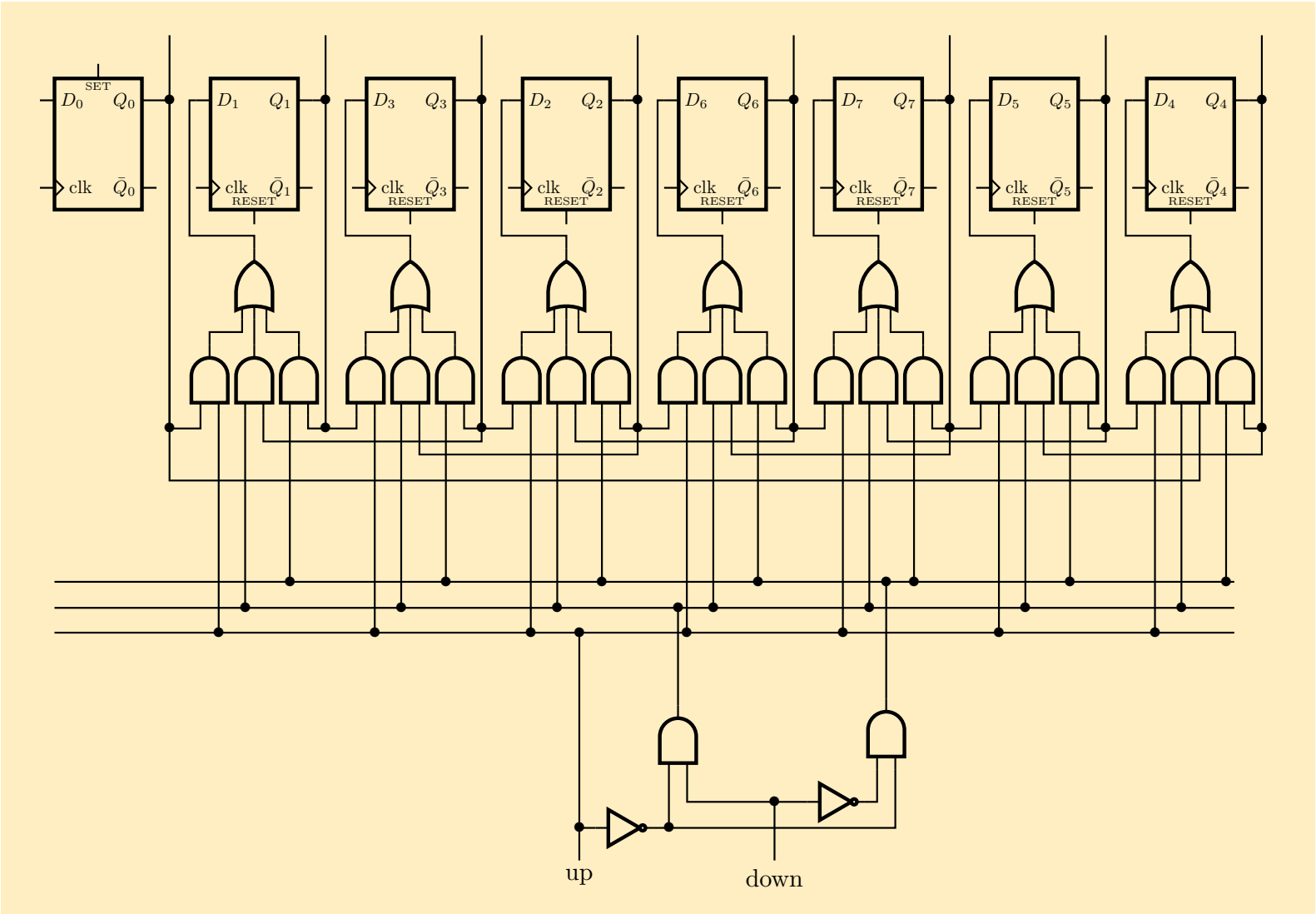
3-Bits Gray Counter, Circuit (1-hot)



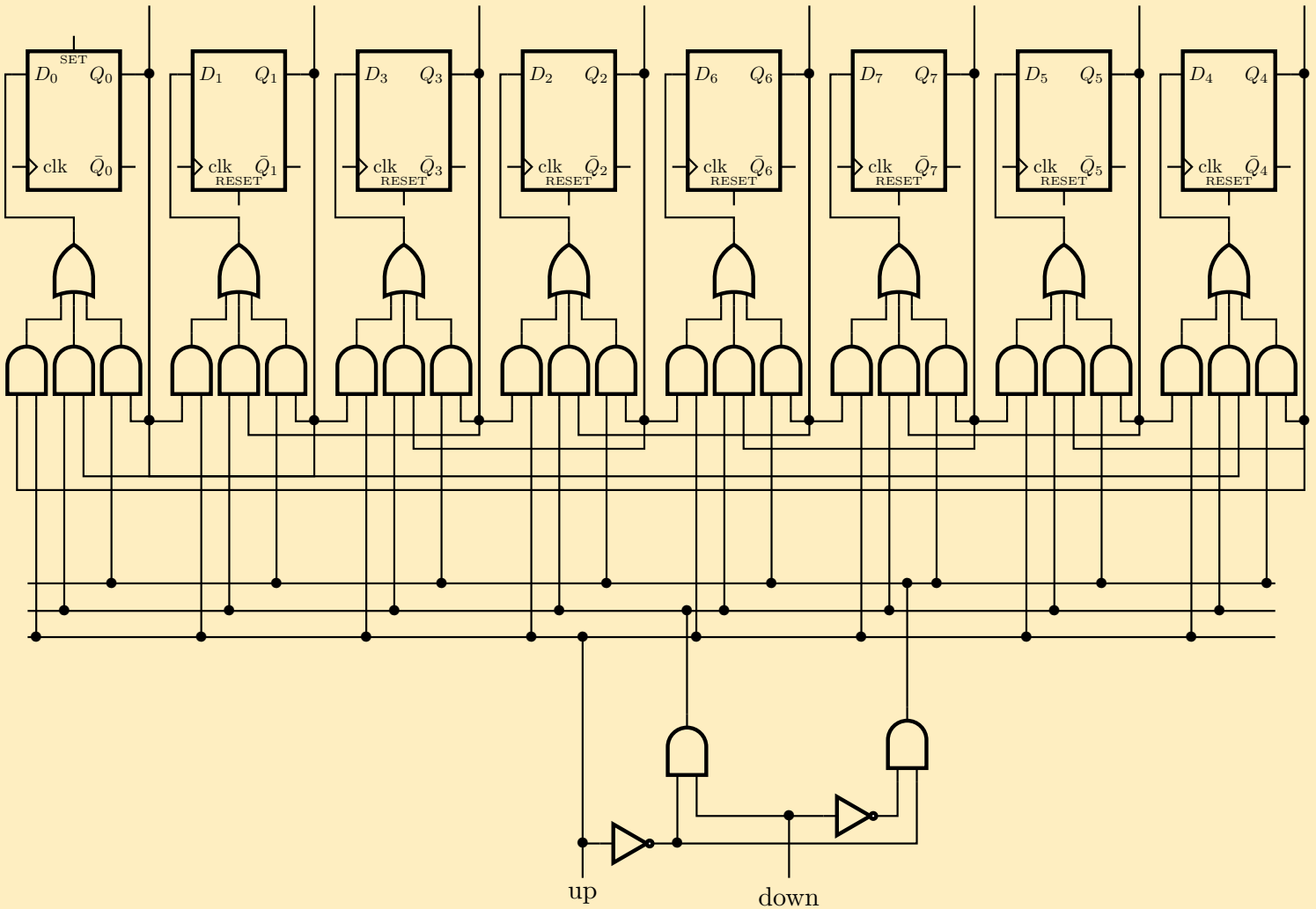
3-Bits Gray Counter, Circuit (1-hot)



3-Bits Gray Counter, Circuit (1-hot)



3-Bits Gray Counter, Circuit (1-hot)



3-Bits Gray Counter, Module (1-hot) I

Listing 17: demo/09_gray3hot/gray3.v

```
`timescale 1ns/1ns

module calc(
  output goto,
  input  up,
  input  down,
  input  prev,
  input  curr,
  input  next
);
  wire upn,downn;
  notGate n1(upn,up);
  notGate n2(downn,down);
```

3-Bits Gray Counter, Module (1-hot) II

```
  wire stay,goDown,goUp;
  and3Gate aStay(stay,upn,downn,curr);
  and3Gate aDown(goDown,upn,down,next);
  andGate aUp(goUp,up,prev);

  or3Gate o(goto,stay,goDown,goUp);
endmodule

module gray3(
  output [2:0] x,
  input  up,down,
  input  init,
  input  clk
);
```

3-Bits Gray Counter, Module (1-hot) III

```
wire [7:0] d,q;

dffi dff(q[0],,d[0],init,1'b0,clk);
for (genvar i = 1; i < 8; i++)
  dffi dff(q[i],,d[i],1'b0,init,clk);

calc c0(d[0],up,down,q[4],q[0],q[1]);
calc c1(d[1],up,down,q[0],q[1],q[3]);
calc c3(d[3],up,down,q[1],q[3],q[2]);
calc c2(d[2],up,down,q[3],q[2],q[6]);
calc c6(d[6],up,down,q[2],q[6],q[7]);
calc c7(d[7],up,down,q[6],q[7],q[5]);
calc c5(d[5],up,down,q[7],q[5],q[4]);
calc c4(d[4],up,down,q[5],q[4],q[0]);
```

3-Bits Gray Counter, Module (1-hot) IV

```
  encoder #(3) e(x,q);

endmodule
```

3-Bits Gray Counter, Testbench (1-hot) I

Listing 18: demo/09_gray3hot/gray3_tb.v

```
`timescale 1ns/1ns

module gray3_tb;

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

  logic init;
  logic up, down;
  wire [2:0] x;
  gray3 g(x, up,down, init, clk);

  initial begin
```

3-Bits Gray Counter, Testbench (1-hot) II


```
    $dumpfile("gray3.vcd");
    $dumpvars;

    init = 1;
    up = 1;
    down = 0;
    #70ns;
    init = 0;

    #2000ns;
    $finish;
  end

  always @(posedge clk) begin
```

3-Bits Gray Counter, Testbench (1-hot) III


```
    up = $urandom_range(0,1);
    down = $urandom_range(0,1);
  end

endmodule
```