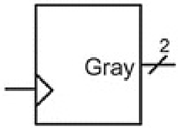
Example: 2-Bit Gray Code Up Counter (Part 1)

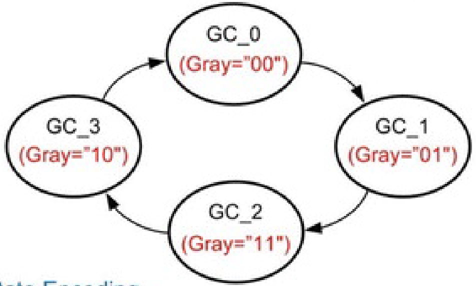
Word Description

We are going to design a <u>2-bit gray code up counter</u>. The counter will output an incrementing gray code pattern on every rising edge of the clock (“00”, “01”, “11”, “10). When the counter reaches “11”, it will start over counting at “00”. The output of the counter is called Gray.



State Diagram & State Transition Table

The state diagram for this counter is below. Notice that there are no inputs to the state machine. Also notice that the machine transitions in a linear pattern through the states and continually repeats the sequence of states. The outputs of this machine depend only on the current state, so they are written inside of the state circles. This is a Moore machine.



(Output)

Current State	Next State	Gray
GC_0	GC_1	“00”
GC_1	GC_2	“01”
GC_2	GC_3	“11”
GC_3	GC_0	“10”

State Encoding

When implementing this counter, we can use “state-encoded outputs”. This means that we choose the state codes so that they match the desired output at each state. This allows the machine to simply use the current state variables for the system outputs. Let’s name the current state variables Q1_cur and Q0_cur and the next state variables Q1_nxt and Q0_nxt. The state code assignments and updated state transition table are below.

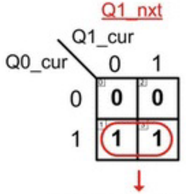
State	Code
GC_0	= “00”
GC_1	= “01”
GC_2	= “11”
GC_3	= “10”

Current State			Next State			Outputs
	Q1_cur	Q0_cur		Q1_nxt	Q0_nxt	Gray
GC_0	0	0	GC_1	0	1	“00”
GC_1	0	1	GC_2	1	1	“01”
GC_2	1	1	GC_3	1	0	“11”
GC_3	1	0	GC_0	0	0	“10”

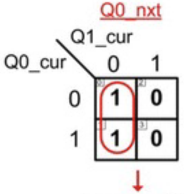
Example: 2-Bit Gray Code Up Counter (Part 2)

Next State Logic

The next state logic for this counter only depends on the current state variables since there are no inputs to the system. Care must be taken when synthesizing the next state logic because the order of the current state variable values in the state transition table is not in a binary count order as in prior examples.



Q1_nxt = Q0_cur



Q0_nxt = Q1_cur’

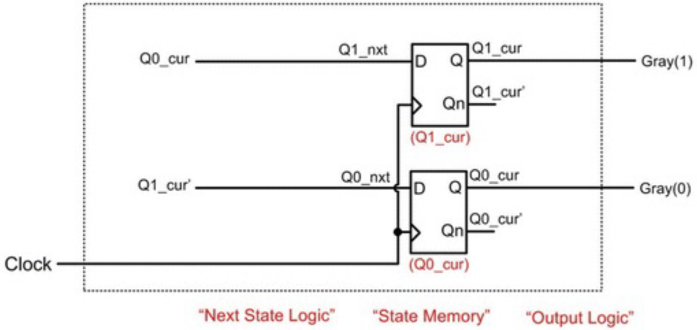
Output Logic

Since we are using state-encoded outputs, the outputs of the system will simply be the current state variables.

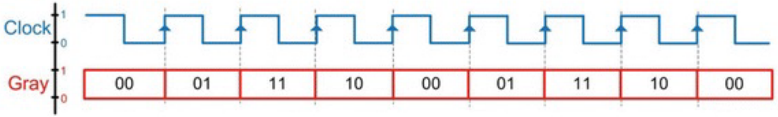
Gray(1) = Q1_cur

Gray(0) = Q0_cur

Logic Diagram



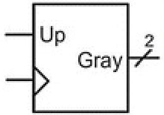
Timing Diagram



Example: 2-Bit Gray Code Up/Down Counter (Part 1)

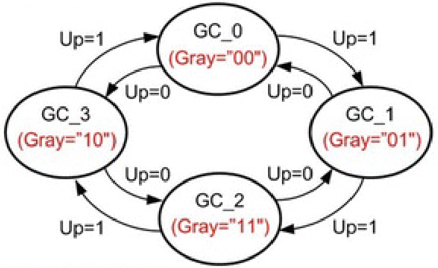
Word Description

We are going to design a 2-bit gray code up/down counter. When the system input "Up" is asserted, the counter will output an incrementing gray code pattern on every rising edge of the clock ("00", "01", "11", "10). When the input Up=0, the counter will output a decrementing gray code pattern. The output of the counter is called Gray.



State Diagram & State Transition Table

The state diagram for this counter is below. The outputs of this machine again only depend on the current state, so they are written inside of the state circles. This is a Moore machine.



Current State	Up (Input)	Next State	Gray (Output)
GC_0	0	GC_3	"00"
	1	GC_1	
GC_1	0	GC_0	"01"
	1	GC_2	
GC_2	0	GC_1	"11"
	1	GC_3	
GC_3	0	GC_2	"10"
	1	GC_0	

State Encoding

Again, this counter will use "state-encoded outputs". Let's name the current state variables Q1_cur and Q0_cur and the next state variables Q1_nxt and Q0_nxt. The state code assignments and updated state transition table are below.

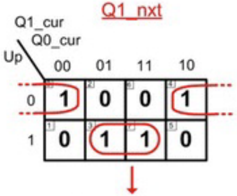
State	Code
GC_0	= "00"
GC_1	= "01"
GC_2	= "11"
GC_3	= "10"

Current State			Input	Next State			Outputs
	Q1_cur	Q0_cur	Up		Q1_nxt	Q0_nxt	Gray
GC_0	0	0	0	GC_3	1	0	"00"
GC_0	0	0	1	GC_1	0	1	"00"
GC_1	0	1	0	GC_0	0	0	"01"
GC_1	0	1	1	GC_2	1	1	"01"
GC_2	1	1	0	GC_1	0	1	"11"
GC_2	1	1	1	GC_3	1	0	"11"
GC_3	1	0	0	GC_2	1	1	"10"
GC_3	1	0	1	GC_0	0	0	"10"

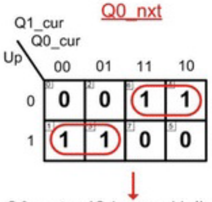
Example: 2-Bit Gray Code Up/Down Counter (Part 2)

Next State Logic

The next state logic for this counter depends on both the current state variables and the input Up. Again, care must be taken when synthesizing the next state logic due to the non-regular pattern of the current state codes in the state transition table.



Q1_nxt = (Q0_cur' · Up') + (Q0_cur · Up)



Q0_nxt = (Q1_cur · Up') + (Q1_cur' · Up)

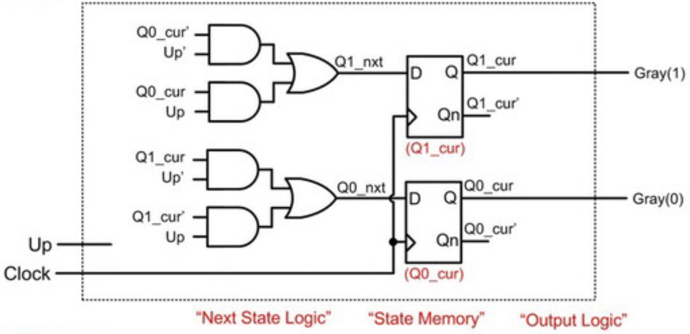
Output Logic

Since we are using state-encoded outputs, the outputs of the system will simply be the current state variables.

Gray(1) = Q1_cur

Gray(0) = Q0_cur

Logic Diagram



Timing Diagram