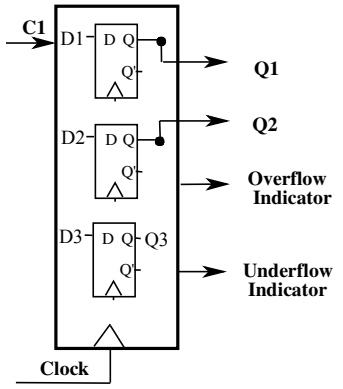
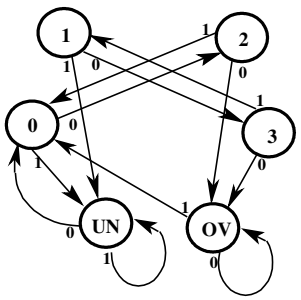
Tutorial 5: Two Mode Counter Design



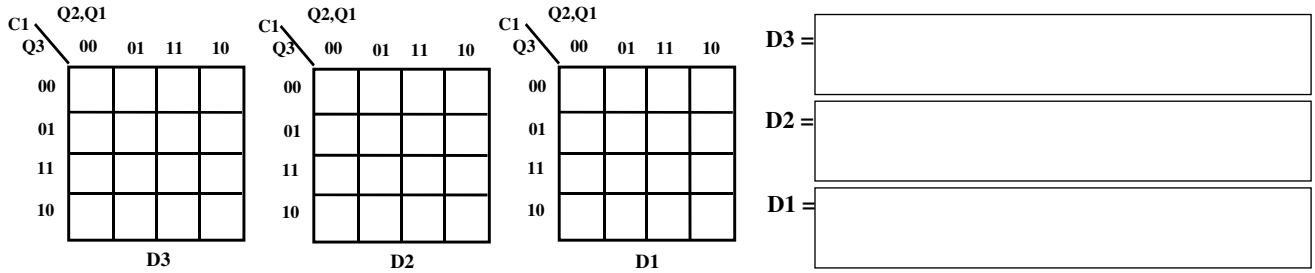


The task is to design a two-bit controlled counter which has two counting bits (**Q2,Q1**), has one control input **C1,** and also two extra outputs, one indicating **overflow**, the other **underflow**. When **C1=0** the counter counts up by 2s; i.e. **0** becomes **2**, **1** becomes **3**. In this mode the values **2** and **3** go to the **overflow** state. When the control input **C1=1**, the counter counts down by 2s, i.e. **3** becomes **1**, and **2** becomes **0**, and **1** and **0** go to the **underflow** state. The counter remains in one of these "error" states until the control bit **C1** is changed at which point it goes to state 0. The finite state machine shows this operating sequence. One way to design this circuit is to set up **six** states. These are the four output count states: **0**, **1, 2, 3,** and the two "error" states: **OV** (overflow) and **UN** (underflow). In order to provide six internal states, we need a minimum of three flip-flops. In the following incomplete transition table six flip-flop outputs are assigned, two are left as **don't care**s.

1 Complete the transition table by showing the **Q3, Q2, Q1** outputs of the "next" state.

	Current State				Next State			
C1	**State**	**Q3**	**Q2**	**Q1**	**State**	**Q3**	**Q2**	**Q1**
0	**0**	**0**	**0**	**0**	**2**			
0	**1**	**0**	**0**	**1**	**3**			
0	**2**	**0**	**1**	**0**	**OV**			
0	**3**	**0**	**1**	**1**	**OV**			
0	**UN**	**1**	**0**	**0**	**0**			
0	**X**	**1**	**0**	**1**	**X**	**X**	**X**	**X**
0	**X**	**1**	**1**	**0**	**X**	**X**	**X**	**X**
0	**OV**	**1**	**1**	**1**	**OV**			
1	**0**	**0**	**0**	**0**	**UN**			
1	**1**	**0**	**0**	**1**	**UN**			
1	**2**	**0**	**1**	**0**	**0**			
1	**3**	**0**	**1**	**1**	**1**			
1	**UN**	**1**	**0**	**0**	**UN**			
1	**X**	**1**	**0**	**1**	**X**	**X**	**X**	**X**
1	**X**	**1**	**1**	**0**	**X**	**X**	**X**	**X**
1	**OV**	**1**	**1**	**1**	**0**			

2. Fill in the K-Maps for the D-type flip-flop inputs **D3**, **D2**, and **D1**. (ie the next state **Q3**, **Q2**, **Q1** values) and determine the minimised expressions for **D3, D2,** and **D1** ;



3. Determine the necessary output circuits which provide the required output signals: **Overflow Indicator** and **Underflow Indicator**.

Optional:

4. Detemine the actual K-maps (replace the **X**s with their correct values) and find the final finite state machine diagram. Will the circuit get stuck in a wrong state?