

АНАЛИЗ И СИНТЕЗ ПОСЛЕДОВАТЕЛЬНОСТНЫХ СХЕМ

Текущее состояние последовательностных логических схем (ПЛС) в отличие от состояния комбинационных логических схем определяется не только поступившими входными сигналами, но и предыдущими значениями входных сигналов. Другими словами, в случае ПЛС вектор выходных сигналов зависит не только от действующих на входах текущих значений сигналов, но и от того состояния, в котором находилась схема к моменту поступления данного набора входных сигналов.

Состояние ПЛС фиксируется с помощью элементов памяти, чем объясняется использование наряду с термином ПЛС термина «автомат с памятью».

Каноническая структура ПЛС содержит конечное число элементов памяти и управляющую комбинационную логическую схему. Управляющая КЛС вырабатывает сигналы, которые воздействуют на входы элементов памяти с целью их переключения. На вход управляющей КЛС поступают внешние сигналы, а также сигналы с выходов элементов памяти. Состояние ПЛС определяется совокупностью состояний всех элементов памяти, т.е. значениями их выходов. Переход ПЛС из старого состояния в новое происходит при поступлении внешних управляющих сигналов. Вследствие обратных связей между элементами памяти и управляющей КЛС значение нового состояния определяется как внешними управляющими сигналами, так и состоянием памяти до перехода. На выходе канонической структуры имеется комбинационная логическая схема, вырабатывающая выходные сигналы ПЛС.



Состояние ПЛС определяется общим состоянием всех триггеров. ПЛС с n триггерами может иметь до 2^n состояний.

По режиму работы различают 2 основные категории ПЛС:

1. **асинхронные ПЛС** - поведение определяется подачей сигналов на входы в определенное время; состояние схемы зависит от порядка изменения сигналов;
2. **синхронные ПЛС** - поведение определяется подачей на входах сигналов в дискретные, хорошо определенные моменты времени; синхронизация осуществляется с помощью импульсов, подаваемых тактовым генератором (clock).

Временная диаграмма тактового сигнала:



Базовыми ПЛС являются регистры и счетчики, использующие в качестве элементов памяти триггеры.

Этапы синтеза ПЛС.

1. Описание функционирования
2. Таблица переходов между состояниями, которая включает входные переменные в момент времени t , значения выходных функций в момент времени $t + 1$.

3. Минимизация логических функций
4. Реализация схемы.

Триггеры

Триггеры) - это последовательностные логические схемы, которые имеют два различных стабильных состояния 0 и 1.

Триггер может хранить двоичную информацию бесконечно долго, и в то же время его состояние может быть прочитано в любое время.

Триггер имеет 2 выхода: прямой и обратный.

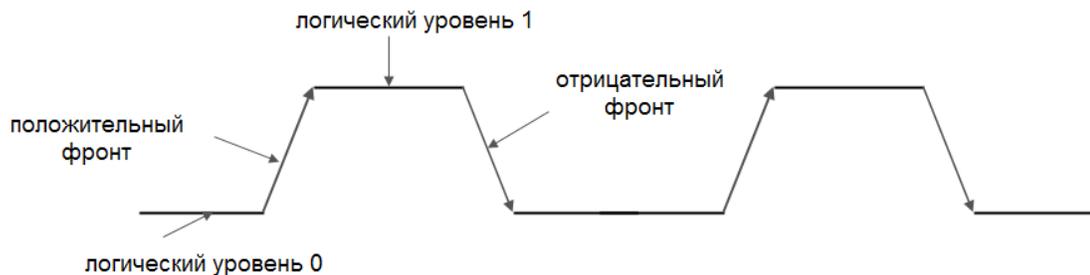
Как и любые ПЛС, триггеры могут быть асинхронными и синхронными.

Асинхронные триггеры переключаются (изменяют состояние) только при подаче сигналов данных, без тактового сигнала (CLK).

Синхронные триггеры переключаются только при активном тактовом сигнале.

В зависимости от режима переключения триггеры подразделяются на:

1. **Latch** (одноуровневые триггеры, которые переключаются при логическом уровне 0 или логическом уровне 1 тактового сигнала).
2. **Flip-flop** (двухуровневые триггеры, ведущий-ведомый (master-slave), которые переключаются при положительном или отрицательном фронте тактового сигнала).



Асинхронный RS триггер (latch)

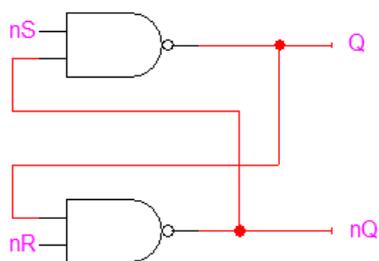
Асинхронный RS-триггер имеет 2 входа данных:

S (Установка, **Set**) и

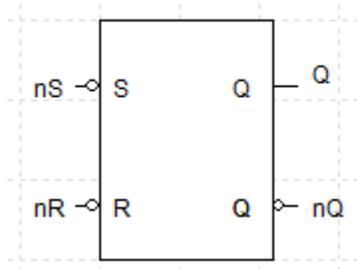
R (Сброс, **Reset**)

и два выхода Q и \bar{Q}

Логическая схема:



Логический символ



В этом случае сигналы R и S активны по 0 (обратная логика).

Уравнение состояния:

$$Q_{t+1} = \overline{Q_t} \overline{R} + S = \overline{Q_t} \overline{R} \cdot \overline{S}$$

Таблица переходов асинхронного RS-триггера

S_t	R_t	Q_{t+1}	nQ_{t+1}	Комментарии
0	0	-	-	Запрещенное состояние
0	1	1	0	Установка в 1 Set
1	0	0	1	Установка в 0 Reset
1	1	Q_t	nQ_t	Сохранение предыдущего состояния

Запрещенное состояние

С логической точки зрения, не имеет смысла записывать и удалять информацию одновременно, следовательно, $\overline{R}_t = 0$ и $\overline{S}_t = 0$ это запрещенное состояние.

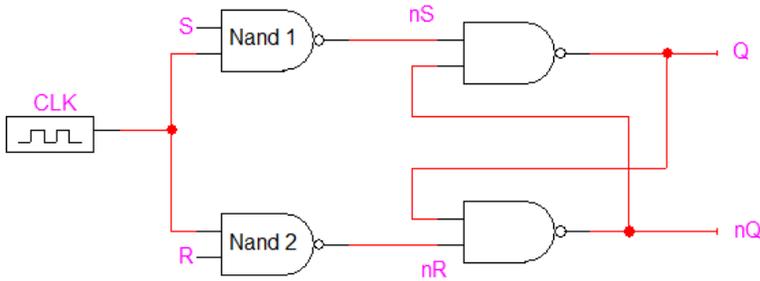
С технологической точки зрения логические элементы не идентичны физически. Когда эти элементы активируются одновременно, между ними возникает конкуренция (на каком из элементов входной сигнал достигнет выхода первым). Соревнование каждый раз будут выигрывать разные логические элементы, без возможности контролировать ситуацию.

При одновременной активации $\overline{R}_t = 0$ и $\overline{S}_t = 0$ на выходах обоих логических элементов И-НЕ установится логическое значение 1. Если сразу после этого будет подано $\overline{R}_t = 1$ и $\overline{S}_t = 1$, определить логическое значение на выходе будет невозможно из-за явления конкуренции между логическими элементами.

Асинхронный RS-триггер - это простейший элемент памяти, который может быть выполнен с помощью элементарных логических схем.

Синхронный RS триггер (latch)

Синхронный RS-триггер получается из асинхронного RS-триггера путем добавления дополнительных логических вентилях для того, чтобы реагировать на входные сигналы R и S только под действием управляющего сигнала, называемого тактовым импульсом (CLK).



В этом случае сигналы R и S активны на 1 (прямая логика).

Выходы синхронного триггера RS изменяются только при активном тактовом сигнале CLK.

В случае, если $CLK = 0$, вентили Nand1 и Nand2 имеют на выходе логическое значение 1, что приводит к поддержанию состояния асинхронного RS-триггера.

Когда $CLK = 1$, вентили Nand1 и Nand 2 действуют как инверторы сигналов S и R.

Логический символ

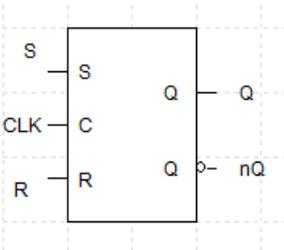


Таблица переходов синхронного RS-триггера

S_t	R_t	Q_{t+1}	nQ_{t+1}	Комментарии
0	0	Q_t	nQ_t	Сохранение предыдущего состояния
0	1	0	1	Установка в 0 Reset
1	0	1	0	Установка в 1 Set
1	1	-	-	Запрещенное состояние

Одновременная подача $R_t = 1$ и $S_t = 1$ в синхронном RS триггере запрещена.

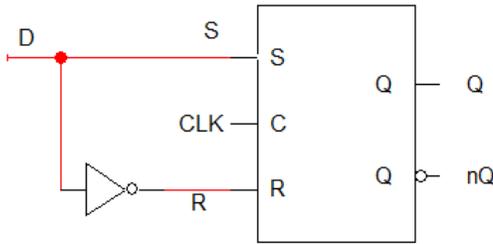
Синхронный D триггер (latch)

Синхронный D (**delay**) триггер имеет один вход D и 2 выхода Q и \bar{Q} . Состояние D триггера определяется сигналом, поданным на вход D. Он задерживает на один такт информацию, которую он получает на входе. Поэтому основное назначение триггера типа D – хранение двоичных слов. Используется в регистрах и схемах RAM памяти

Уравнение состояния:

$$Q_{t+1} = D$$

Логическая схема получается из RS триггера при условии $S = \bar{R}$



Логический символ

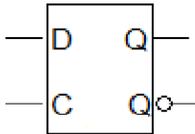


Таблица переходов D триггера

D_t	Q_{t+1}	nQ_{t+1}	Комментарии
0	0	1	Установка в 0
1	1	0	Установка в 1

Следующее состояние синхронного D триггера зависит только от сигнала, подаваемого на вход, и не зависит от текущего состояния триггера.

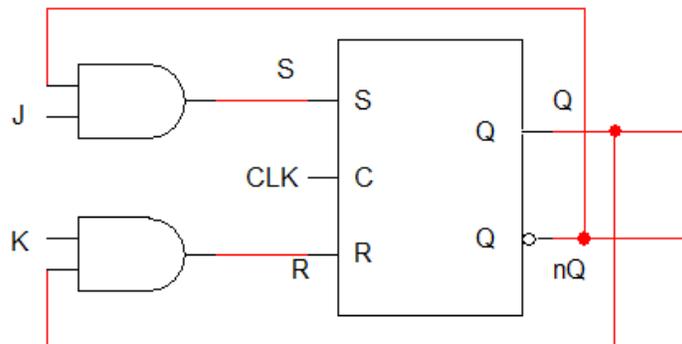
Синхронный JK триггер (latch)

JK триггер сохраняет свою функциональность и тогда, когда $RS=1$. Это достигается путем дополнительных линий обратной связи.

Уравнение состояния:

$$Q_{t+1} = J\bar{Q}_t + \bar{K}Q_t$$

Логическая схема



Поскольку выходы Q и nQ не могут быть равны одновременно логической 1, запрещенная комбинация $S = R = 1$ не может подаваться на входы S и R.

Когда $J=1$ (при $K=0$) триггер устанавливается в состояние логической 1. Если же $K=1$ (при $J=0$), то триггер устанавливается в состояние логического нуля. При $J=K=0$ триггер хранит прежнее состояние. Следовательно, входы J и K можно рассматривать как входы S и R соответственно. При $J=K=1$ триггер ведет себя особым образом, изменяя значение прежнего состояния на противоположное ($0 \rightarrow 1$, соответственно $1 \rightarrow 0$).

Логический символ

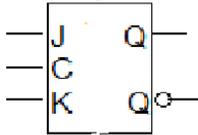


Таблица переходов JK триггера

J_t	K_t	Q_{t+1}	nQ_{t+1}	Комментарии
0	0	Q_t	nQ_t	Сохранение предыдущего состояния
0	1	0	1	Установка в 0 Reset
1	0	1	0	Установка в 1 Set
1	1	nQ_t	Q_t	Инвертирование состояния Q_t

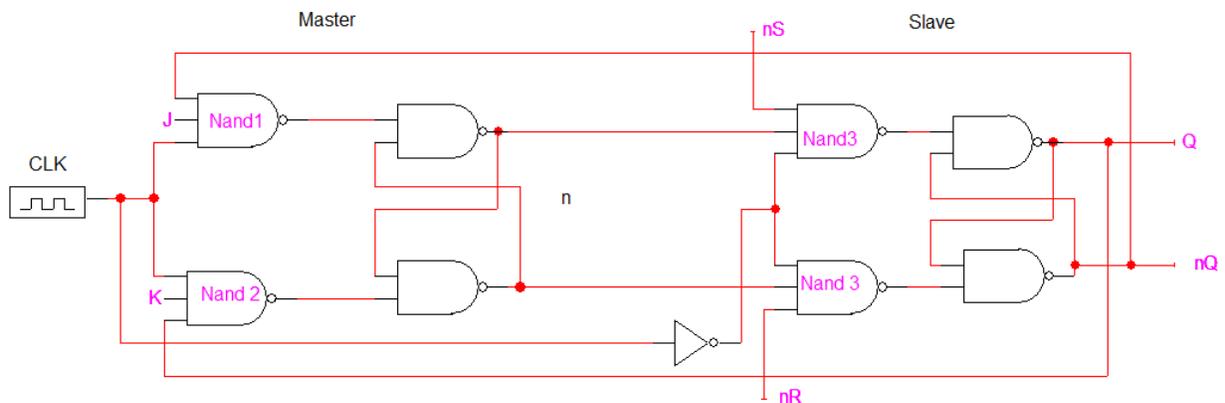
Синхронный JK триггер (flip-flop)

Линия обратной связи приводит к тому что JK триггер осциллирует (изменяет состояние), когда $J = K = 1$ и тактовый сигнал (CLK) остается в логической 1. Для того чтобы триггер переключался только один раз, длительность импульса на CLK должна быть больше, чем время распространения сигнала через логический вентиль, и короче, чем время прохождения сигнала через два логических элемента. В реальных схемах это сложно синхронизировать. Вот почему одноуровневая интерпретация триггера ненадежна.

Триггеры типа Flip-flop (ведущий-ведомый) представляют собой структуры, которые позволяют решить проблему переключения триггеров.

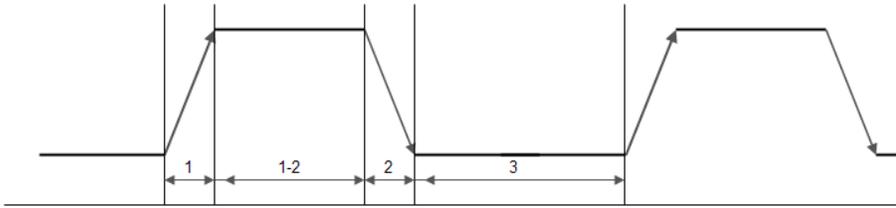
Принцип Flip-flop (ведущий-ведомый) может быть применен к любой триггерной схеме.

Структура flip-flop master-slave состоит из двух latch триггеров, одного “master” и другого “slave”.



В flip-flop триггерах на положительном фронте тактового сигнала информация входит в ведущий триггер, при этом ведомый триггер практически отключен. На

следующем отрицательном фронте информация передается от ведущего к ведомому триггеру, и появляется на выходах на отрицательном фронте тактового сигнала. Это обеспечивает хорошее разделение между входами данных и выходами триггеров.



Период времени 2 - это время установки т.е. период, в течение которого данные должны быть подготовлены до подачи тактового импульса.

Период времени 3 - время паузы.

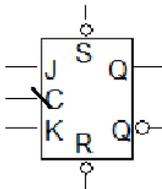
В течение периода времени 1 входные вентили еще не открыты, а вентили Nand3 и Nand4 блокируются и, таким образом, изолируют ведущий триггер от ведомого.

В зоне 1-2 элементы Nand1 и Nand2 открыты, и информация передается мастеру. Вентили Nand 3 и Nand 4 закрыты, и ведомый триггер хранит свою старую информацию.

Входы nS и nR являются асинхронными входами, которые действуют на последнем уровне логических вентилях, не зависят от тактового сигнала и имеют приоритет над синхронными входами J и K (когда один из них активирован, триггер будет работать асинхронно).

Асинхронные входы nS и nR могут присутствовать в триггерах любого типа.

Логический символ flip-flop триггера JK не отличается от символа latch триггера, но иногда в логическом символе может присутствовать наклонная линия (если триггер переключается на отрицательном фронте (направленная вниз), если триггер переключается на положительном фронте (направленная вверх)).



Т триггер (flip-flop)

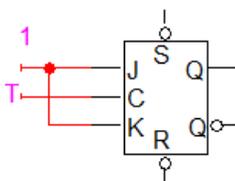
Т триггер (**Toggle**) получается из JK триггера типа flip-flop при соединении входов J и K.

Уравнение состояния:

$$Q_{t+1} = T\bar{Q}_t + \bar{T}Q_t$$

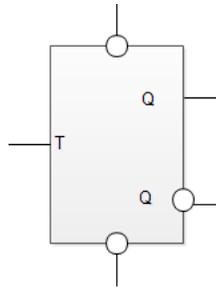
Т-триггер может быть реализован в асинхронном и синхронном исполнении.

Логическая схема асинхронного Т триггера

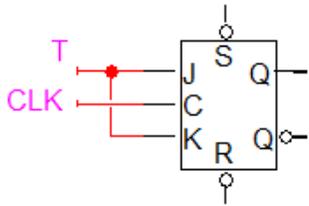


Сигнал CLK выполняет функцию входа Т триггера. Триггер меняет свое состояние (переключается), когда сигнал Т изменяется с 1 на 0 (на отрицательном фронте).

Логический символ асинхронного Т триггера



Логическая схема синхронного Т триггера



Синхронный Т-триггер будет переключаться, когда на вход Т будет подано логическое значение 1, а тактовый сигнал CLK изменится с 1 на 0 (отрицательный фронт).
Логический символ асинхронного Т триггера

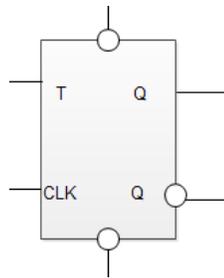
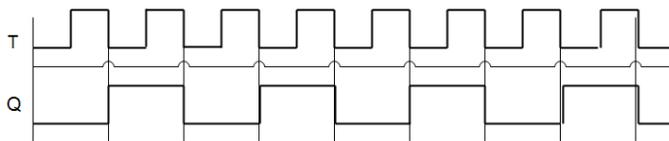


Таблица переходов Т триггера

T_t	Q_{t+1}	\bar{Q}_{t+1}	Комментарии
0	Q_t	\bar{Q}_t	Сохранение предыдущего состояния
1	\bar{Q}_t	Q_t	Инвертирование состояния Q_t

Триггер Т используется в счетчиках и в схемах деления частоты на 2.

Временная диаграмма Т триггера



Введем понятие функции возбуждения, характерной для каждого триггера. Она показывает какими должны быть входы чтобы осуществить необходимый переход из одного состояния в другое. Функция возбуждения используется для синтеза сложных последовательных схем с триггерами.

Таблицы возбуждения получают из таблиц переходов триггеров.

Триггер	Таблицы возбуждения	Таблицы переходов																																								
RS	<table border="1"> <thead> <tr> <th>Q_t</th> <th>Q_{t+1}</th> <th>S</th> <th>R</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>*</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>*</td> <td>0</td> </tr> </tbody> </table>	Q_t	Q_{t+1}	S	R	0	0	0	*	0	1	1	0	1	0	0	1	1	1	*	0	<table border="1"> <thead> <tr> <th>S_t</th> <th>R_t</th> <th>Q_{t+1}</th> <th>nQ_{t+1}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Q_t</td> <td>nQ_t</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>-</td> <td>-</td> </tr> </tbody> </table>	S_t	R_t	Q_{t+1}	nQ_{t+1}	0	0	Q_t	nQ_t	0	1	0	1	1	0	1	0	1	1	-	-
Q_t	Q_{t+1}	S	R																																							
0	0	0	*																																							
0	1	1	0																																							
1	0	0	1																																							
1	1	*	0																																							
S_t	R_t	Q_{t+1}	nQ_{t+1}																																							
0	0	Q_t	nQ_t																																							
0	1	0	1																																							
1	0	1	0																																							
1	1	-	-																																							
D	<table border="1"> <thead> <tr> <th>Q_t</th> <th>Q_{t+1}</th> <th>D</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Q_t	Q_{t+1}	D	0	0	0	0	1	1	1	0	0	1	1	1	<table border="1"> <thead> <tr> <th>D_t</th> <th>Q_{t+1}</th> <th>nQ_{t+1}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	D_t	Q_{t+1}	nQ_{t+1}	0	0	1	1	1	0																
Q_t	Q_{t+1}	D																																								
0	0	0																																								
0	1	1																																								
1	0	0																																								
1	1	1																																								
D_t	Q_{t+1}	nQ_{t+1}																																								
0	0	1																																								
1	1	0																																								
JK	<table border="1"> <thead> <tr> <th>Q_t</th> <th>Q_{t+1}</th> <th>J</th> <th>K</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>*</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>*</td> </tr> <tr> <td>1</td> <td>0</td> <td>*</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>*</td> <td>0</td> </tr> </tbody> </table>	Q_t	Q_{t+1}	J	K	0	0	0	*	0	1	1	*	1	0	*	1	1	1	*	0	<table border="1"> <thead> <tr> <th>J_t</th> <th>K_t</th> <th>Q_{t+1}</th> <th>nQ_{t+1}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Q_t</td> <td>nQ_t</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>nQ_t</td> <td>Q_t</td> </tr> </tbody> </table>	J_t	K_t	Q_{t+1}	nQ_{t+1}	0	0	Q_t	nQ_t	0	1	0	1	1	0	1	0	1	1	nQ_t	Q_t
Q_t	Q_{t+1}	J	K																																							
0	0	0	*																																							
0	1	1	*																																							
1	0	*	1																																							
1	1	*	0																																							
J_t	K_t	Q_{t+1}	nQ_{t+1}																																							
0	0	Q_t	nQ_t																																							
0	1	0	1																																							
1	0	1	0																																							
1	1	nQ_t	Q_t																																							
T	<table border="1"> <thead> <tr> <th>Q_t</th> <th>Q_{t+1}</th> <th>T</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	Q_t	Q_{t+1}	T	0	0	0	0	1	1	1	0	1	1	1	0	<table border="1"> <thead> <tr> <th>T_t</th> <th>Q_{t+1}</th> <th>nQ_{t+1}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Q_t</td> <td>nQ_t</td> </tr> <tr> <td>1</td> <td>nQ_t</td> <td>Q_t</td> </tr> </tbody> </table>	T_t	Q_{t+1}	nQ_{t+1}	0	Q_t	nQ_t	1	nQ_t	Q_t																
Q_t	Q_{t+1}	T																																								
0	0	0																																								
0	1	1																																								
1	0	1																																								
1	1	0																																								
T_t	Q_{t+1}	nQ_{t+1}																																								
0	Q_t	nQ_t																																								
1	nQ_t	Q_t																																								