

3.10. Преобразователи кодов, шифрация и дешифрация

Следующим распространенным классом комбинационных устройств цифровой электроники являются **шифраторы и дешифраторы (кодеры/декодеры)**. Подобные устройства относятся к классу **преобразователей кодов**.

Преобразователь кодов

✓ ПрК: n входов и m выходов, однозначное преобразование комбинаций значений входных сигналов в комбинации выходных сигналов по определенному правилу.

✓ Если число необходимых комбинаций равно K , то должно быть $K \leq 2^n$, $K \leq 2^m$.

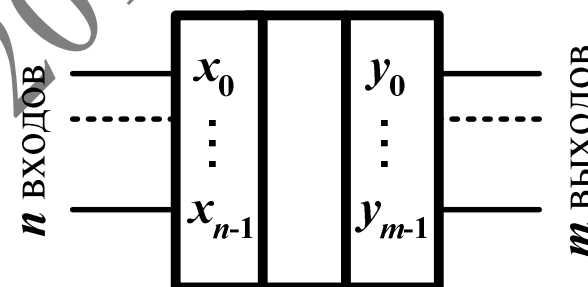
✓ Количество n , m и K определяется:

- характером системы источников, создающих комбинации входных сигналов
- характером системой потребителей комбинаций выходных сигналов,
- возможной взаимосвязью систем, характером задачи, которую решает преобразование и т.п.

✓ В общем случае возможна **избыточность**:

- всех комбинаций входных значений (всего может быть $N = 2^n > K$),
- всех возможных комбинаций выходных сигналов (максимальное количество $M = 2^m$).

Среди разнообразия возможных ПрК выделяют ряд подклассов, широко применяемых на практике как для решения часто встречающихся специальных задач, так и используемых в качестве *универсальных компонент* для построения сложных ЦЭУ.



✓ Преобразователи кодов в цифровой электронике разделяют на два класса:

шифраторы и дешифраторы

- Можно встретить разные трактовки, объясняющие принцип такого деления.
- Есть «узкие» и «широкие» интерпретации термина шифратор/дешифратор.

Одна из самых общих (широких) трактовок подразумевает:

а) Есть некоторый «внешний» формат информации заложенной в комбинациях сигналов подаваемых в электронную систему, либо выводимых из нее.

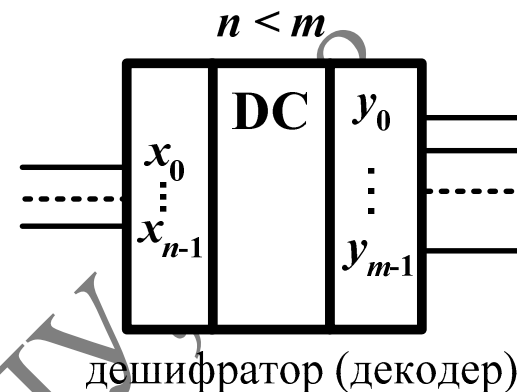
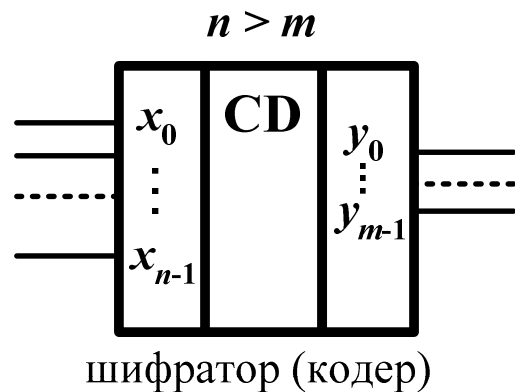
- Информация может быть очень сложно структурирована (характером источников либо потребителей) с большой избыточностью.
- Информация: хранение, обработка, передача. По возможности, в простом и компактном виде.

б) Для каждой комбинации сигналов, отражающей «внешнюю» информацию должен быть сопоставлен более простой, упорядоченный по структуре и компактный набор значений сигналов – код комбинации (например, номер комбинации). → *Кодированная форма* подразумевает меньшую избыточность представления информации. *Восстановленная (декодированная) форма*, удобную для приемников, предполагает большую избыточность.

в) Если избыточность наборов на входе преобразователя больше, чем на выходе, то это характерная ситуация для кодирования информации. Обратное характерно для декодирования.

✓ Следствие: простое правило деления:

- если $N = 2^n > M = 2^m$ ($n > m$, т.е. входов > выходов) → шифратор, кодер;
- если $N = 2^n < M = 2^m$ ($n < m$, т.е. входов < выходов) → дешифратор, декодер;



источники
(датчики)
сигналов

сложная комбинация сигналов
исходная информация, много сигналов (высокая избыточность)

кодирование



код (номер) комбинации
"сжатая" информация, мало сигналов (низкая избыточность)

→ передача, хранение,
обработка, принятие
решений и т.п.

Формирователь
кода

код (номер) комбинации
"сжатая" информация, мало сигналов (низкая избыточность)

декодирование



сложная комбинация сигналов
много сигналов (высокая избыточность)

→ управление
набором узлов,
индикация,

Возможна ситуация, когда $n = m$ и $N = M$. В этом случае компактность представления информации во входных и выходных сигналах одинакова и такое устройство в равной степени можно интерпретировать как шифратор или как дешифратор.

✓ Понятие о шифрации и дешифрации (кодировании и декодировании) и соответствующих устройствах (кодерах/декодерах) встречается не только в классификации элементов цифровой электроники, но и во многих других задачах, касающихся преобразования информации, в том числе информации, представленной в виде цифрового сигнала. Например:

- Линии передачи данных.
- Системы записи-воспроизведения данных.
- Криптография (возможно в сочетании с предыдущими).

! При этом представление о шифрации/дешифрации не соответствует, описанному выше принципу разделения, принятому для элементов цифровой электроники, а является **обратным**.

Подразумевается, что исходные сигналы отражают информацию в простой и понятной форме (задают номера буквенных символов и т.п.). Кодирование на входе системы предполагает усложнение и увеличение избыточности. На выходе системы происходит восстановление исходного сжатого вида информации происходит в процессе декодирования.

Исходная/восстановленная комбинация – более простая, а код – более сложная структура.

"Усложнение" передаваемых (записываемых) данных преследует следующие цели:

- Засекречивание данных.
- Обеспечение синхронизации потока данных на приемном конце.
- Распознавание системы структурирования данных.
- Помехозащищающее кодирование (обнаружение и исправление ошибок).

Примеры

ВШШФИКТ, СПбПУ, 2018

ПрК с прямым двоичным кодом

На практике очень распространены относительно простые и понятные типы ПрК, которые предполагают, что сложные информационные комбинации пронумерованы и в качестве кода комбинации использовать двоичное число, соответствующее номеру комбинации.

- Шифратор (кодер) по комбинации значений входных сигналов выдает ее номер в прямом двоичном коде.
- Дешифратор (декодер) по заданному значениями входных сигналов номеру в двоичном коде, воспроизводит соответствующую комбинацию значений выходных сигналов.



✓ Для шифраторов/дешифраторов «прямого двоичного кода», чаще всего выполняется:

- $K = M = 2^m$ (для шифраторов);
- $K = N = 2^n$ (для дешифраторов).

Если число требуемых комбинаций K не равно степени двойки, то $2^{n-1} < K < 2^n$, либо $2^{m-1} < K < 2^m$.

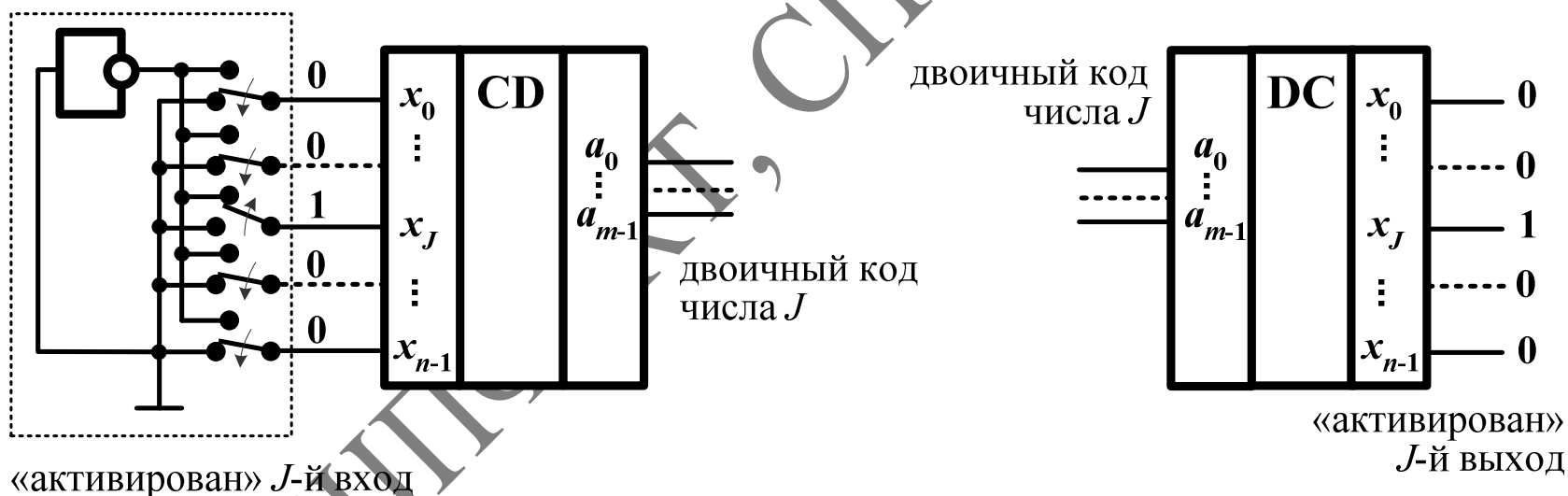
"Узкая" трактовка понятия шифратор/дешифратор

На практике (и в литературе) очень часто под шифраторами/дешифраторами (кодерами/декодерами) подразумевается только один подвид ПрК связующий прямой двоичный код с т. н. унитарным кодом.

✓ **Унитарный код** подразумевает, что информация представлена в виде набора K сигналов (на K выводах), только один которых имеет значение «1», а остальные «0». При этом двоичный код комбинации соответствует номеру активного сигнала (выводы).

✓ Унитарный код является избыточным (K сигналов в общем случае могли бы отражать 2^K комбинаций значений). Преобразование в прямой двоичный код снимает эту избыточность и снижает количество используемых сигналов.

В большинстве случаев в таких шифраторах $K = N = 2^M$, а для дешифраторов $K = M = 2^N$.

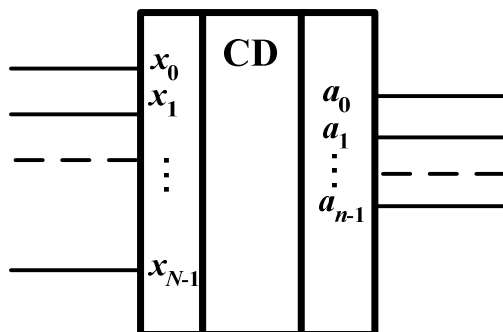


! Выходы шифратора часто трактуют как **адресные сигналы**, а входные сигналы дешифратора называют **адресными входами**.

При "узкой" трактовке, CD/DC более общего типа интерпретируют как ПрК.

3.11. Шифраторы и дешифраторы

✓ **Шифратор (кодер)** – это устройство преобразования кодов с n входов, m выходов, $n > m$.



По заданному правилу, для произвольной комбинации значений входных сигналов формирует определенную комбинацию выходных сигналов (сжатая кодированная форма информации).

! На практике часто под шифратором подразумевают только наиболее распространенный универсальный вариант шифратора это преобразователь т. н. унитарного кода в прямой двоичный код.

- Шифратор имеет $N = 2^n$ входов и n выходов (адресных). CD $N: n$
- В текущий момент времени, значение «1» может присутствовать только на одном из входов. Правило преобразования: на выходе шифратор формирует комбинацию сигналов a_{n-1}, \dots, a_0 , которая показывает двоичное число, соответствующее номеру входа со значением «1»:

$$\text{если } x_J = 1, \text{ то } (a_{n-1}, \dots, a_0)_2 = J.$$

Наглядный пример подобных устройств:

- Первичная регистрация нажатия кнопки на клавиатуре (каждой клавише соответствует свой входной сигнал, а CD формирует двоичный код с номером нажатой клавиши).
- Срабатывание датчиков вдоль маршрута предмета.

! По правилу представления информации в унитарном коде, комбинации с двумя и более единицами являются запрещенными. Однако на практике на входы дешифратора можно подать

любую комбинацию сигналов. Как правило, реальные дешифраторы при этом отображают на выходе номер первого (либо последнего) из входов со значением «1».

Строение шифратора

x_3	x_2	x_1	x_0	a_1	a_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Унитарно-двоичный CD с 4-мя входами и двумя выходами может быть определен ТИ вида:

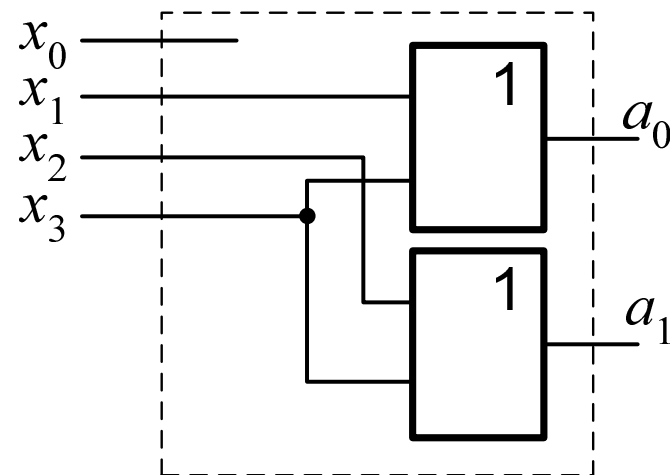
По КК с учетом оптимального доопределения функций (в запрещенных комбинациях) получим формулы для выходных сигналов в виде минимизированных сумм:

$$a_0 = x_1 + x_3, \quad a_1 = x_2 + x_3.$$

! Функции не зависят от x_0 .

Это связано с условием (в ТИ), что какой-то из сигналов обязательно должен иметь значение «1». Поэтому нулевые значения на трех «старших» входах однозначно определяют наличие значения «1» на входе x_0 (и $a_0 = a_1 = 0$).

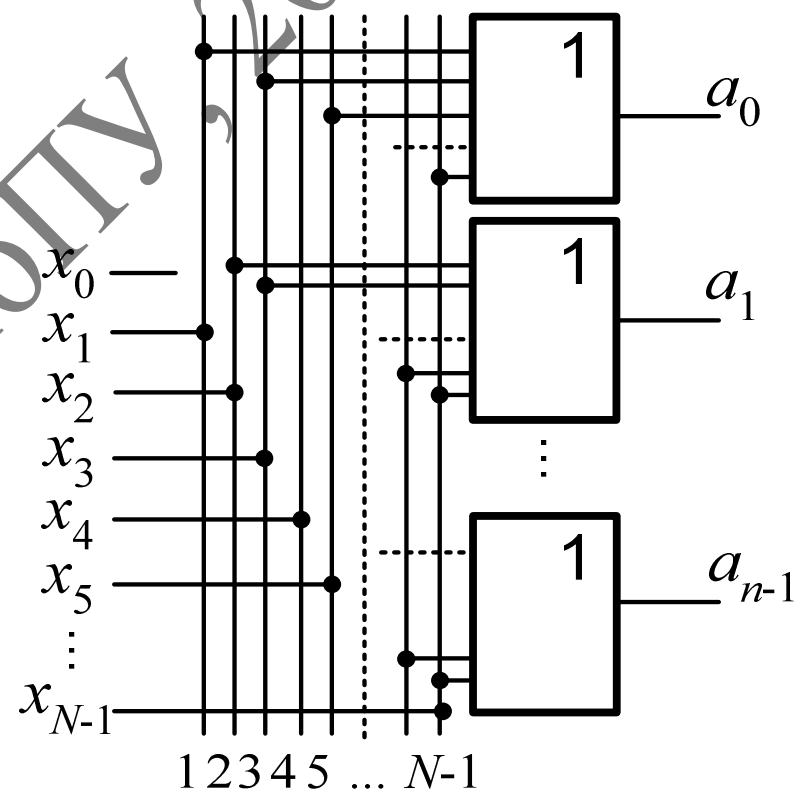
! Если допускается отсутствие 1 на входах все формулы должны быть дополнены слагаемым в виде минтерма $x_3 \cdot x_2 \cdot x_1 \cdot x_0$.



В общем случае, при $N > 4$ (т. е. для $N = 8, 16, \dots$) таблица истинности, выражения для функций и логическая схема шифратора имеют структуру

x_{N-1}	x_{N-2}	...	x_1	x_0	a_{n-1}	...	a_1	a_0
0	0	...	0	1	0	...	0	0
0	0	...	1	0	0	...	0	1
0	1	...	0	0	1	...	1	0
1	0	...	0	0	1	...	1	1

$$\begin{aligned}
 a_0 &= x_1 + x_3 + x_5 + \dots; \\
 a_1 &= (x_2 + x_3) + (x_6 + x_7) + \dots; \\
 &\dots \\
 a_{n-1} &= (x_{N/2} + x_{1+N/2} + \dots + x_{N-1}).
 \end{aligned}$$



Надо N сумматоров, имеющих $N/2$ входов.

На 1-й сумматор – сигналы нечетных входов шифратора: (x_1, x_3, \dots) .

На 2-й сумм. – вторая, третья и остальные "нечетные пары": $(x_2, x_3, x_6, x_7, \dots)$.

На 3-й сумм. – вторая, четвертая и остальные "нечетные четверки": $(x_4, x_5, x_6, x_7, \dots)$ и т. д.

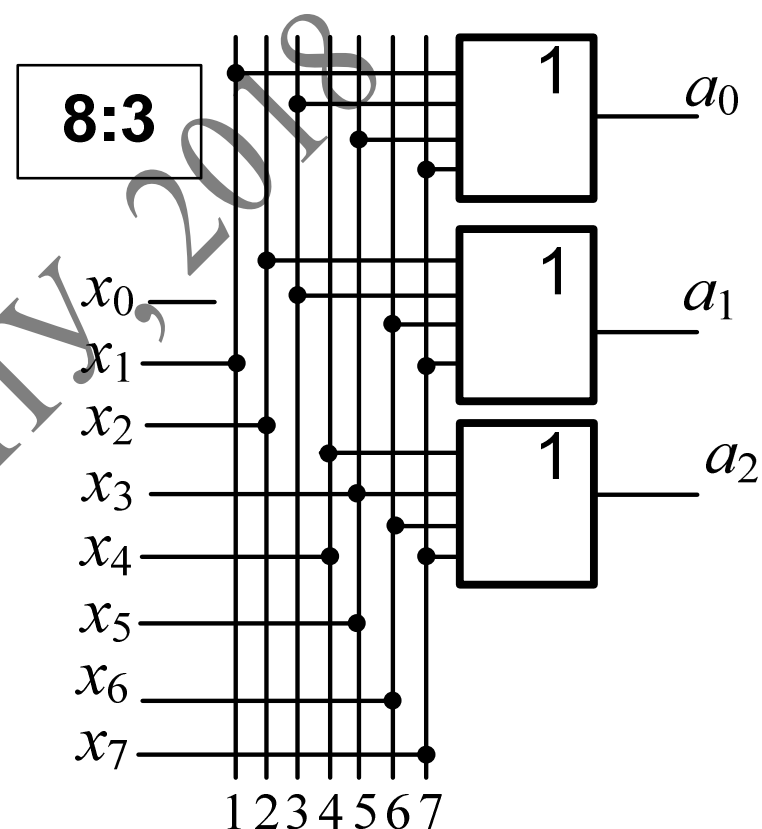
На последний сумматор подается вторая половина входных сигналов.

Пример: ЛС для CD 8:3

- На практике, обычно, востребованы CD, ориентированные под разнообразные конкретные источники сигналов и в каждом конкретном случае требуют разных дополнительных особенностей.
- DC (особенно в широкой трактовке) можно использовать по прямому назначению с конкретным источником сигналов, для которого он создан, и крайне сложно применить шифратор в качестве универсального элемента цифровой схемотехники.
- В номенклатуре выпускаемых микросхем общего применения класс шифраторов фактически не представлен, в отличие от MUX, DMX и DC.

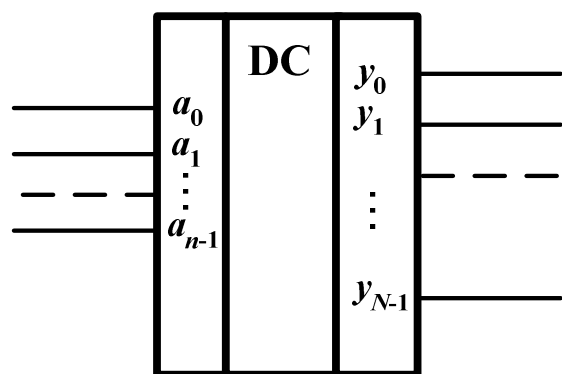
✓ Унитарно-двоичные шифраторы можно наращивать.

Однако в силу упомянутого отсутствия разнообразных микросхем этого класса в номенклатуре выпускаемых компонент рассмотрения приемов такого наращивания не представляется целесообразным.



Дешифратор (декодер) ПрК, n входов и t выходов ($n < t$),

По заданному правилу, для произвольной комбинации значений входных сигналов (кодированная форма информации) формирует определенную комбинацию выходных сигналов (декодированная форма информации).



Основной распространенный вариант преобразования кода, который реализуют дешифраторы, это преобразование прямого двоичного кода (позиционного) в унитарный код. Часто дешифраторами (декодерами) называют *только* этот такой элемент.

В этом случае дешифратор имеет $N = 2^n$ выходов.

Правило:

Комбинация адресных входов a_{n-1}, \dots, a_0 задает двоичное число $J = (a_{n-1}, \dots, a_0)_2$.

Сигнал на выходе с номером J принимает значение «1», в то время как остальные выходные сигналы принимают значение «0».

Наглядный пример применения подобных устройств – формирование сигнала для «подсвечивания» элемента "линейного" индикатора по заданному двоичным кодом номеру элемента. Для обозначения ранга дешифратора обычно указывают соотношение входов и выходов $n:N$.

Строение дешифратора (двоично-унитарного)

Двоично-унитарный DC преобразует один код (прямой двоичный) в другой (унитарный).

Однако нетрудно убедиться, что такое преобразование является очень близким к действию DMX 1: N , адресные входы которого соответствуют входам дешифратора.

Разница DC и DMX:

- DC формирует на «заданном» выходе значение «1»,
- DMX – формирует такое же значение, что и на его информационном входе.

! Т.е. строение такого DC $n:N$ повторяет структуру DMX 1: N , описанную в п. 39, при исключении из нее всех выводов входного сигнала I .

! В реальных устройствах, как правило, реализуют дополнительные сигналы управления работой схемы C (входы разрешения/блокировки) \rightarrow схему DC обычно можно заставить функционировать как DMX, если подавать на вход C сигнал I .

! Если на вход DMX подать фиксированный сигнал со значением «1», то он, очевидно, будет работать идентично дешифратору.

Поэтому

а) В номенклатуре реальных цифровых микросхем, как правило, одни и те же микросхемы предназначены для использования их в качестве как DMX, так и «двоично-унитарных» DC. Производители обычно не разделяют такие DC и DMX и часто называют микросхемы такого

назначения DMX. Примеры микросхем-DMX в п. 3.9, фактически являются примерами микросхем-DC двоично-унитарного преобразования.

б) Принципы наращивания DC идентичны способам наращивания DMX, с той разницей, что выходы DC предыдущего уровня подключаются к *входам разрешения* DC последующего уровня (в схеме на рисунке в п. 3.9 следует просто заменить информационные входы *I* на входы разрешения *C*).

Демультимплексор-дешифратор как универсальный логический элемент

На основе демультимплексоров-дешифраторов (двоично-унитарных) можно относительно просто реализовать *произвольную ЛФ* от *n* аргументов. Для этого удобно использовать таблицу истинности, в которой наборы аргументов расположены так, что соответствуют порядковому номеру строки, представленному в двоичном коде (так было сделано в ТИ, приведенных ранее).

- Аргументы функции подаются на адресные входы DC (DMX)
- В случае обычного дешифратора, для генерации функции, следует подсоединить часть выходов дешифратора, к сумматору, формирующему выходной сигнал. При этом следует выбрать те выводы дешифратора, номера которых совпадают с номерами строк таблицы истинности, в которых функция принимает значение «1».

! На практике часто встречаются дешифраторы с инверсными выходами. В этом случае при генерации логической функции можно использовать выходной перемножитель и номера строк таблицы истинности, в которых функция принимает значение «0».

Пример,

схемы реализации функции, заданной таблицей истинности на основе обычного дешифратора и дешифратора с инверсными выходами

№	X_2	X_1	X_0	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

