

Лабораторная работа № 8. Дешифраторы. Часть 2.

Функции дешифраторов и шифраторов понятны из их названий. Дешифратор преобразует входной двоичный код в номер выходного сигнала (дешифрирует код), а шифратор преобразует номер входного сигнала в выходной двоичный код (шифрует номер входного сигнала). Количество выходных сигналов дешифратора и входных сигналов шифратора равно количеству возможных состояний двоичного кода (входного кода у дешифратора и выходного кода у шифратора), то есть 2^n , где n — разрядность двоичного кода (рис. 5.1). Микросхемы дешифраторов обозначаются на схемах буквами DC (от английского Decoder), а микросхемы шифраторов — CD (от английского Coder).

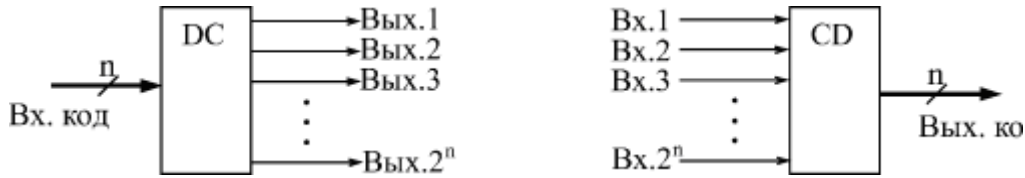


Рис. 1. Функции дешифратора (слева) и шифратора (справа)

На выходе дешифратора всегда присутствует только один сигнал, причем номер этого сигнала однозначно определяется входным кодом. Выходной код шифратора однозначно определяется номером входного сигнала.

Рассмотрим подробнее функцию дешифратора.

В стандартные серии входят дешифраторы на 4 выхода (2 разряда входного кода), на 8 выходов (3 разряда входного кода) и на 16 выходов (4 разряда входного кода). Они обозначаются соответственно как 2–4, 3–8, 4–16. Различаются микросхемы дешифраторов входами управления (разрешения/запрета выходных сигналов), а также типом выхода: 2С или ОК. Выходные сигналы всех дешифраторов имеют отрицательную полярность. Входы, на которые поступает входной код, называют часто адресными входами. Обозначают эти входы 1, 2, 4, 8, где число соответствует весу двоичного кода (1 — младший разряд, 2 — следующий разряд и т.д.), или А0, А1, А2, А5. В отечественных сериях микросхемы дешифраторов обозначаются буквами ИД. На рис. 5.2 показаны три наиболее типичных микросхемы дешифраторов.

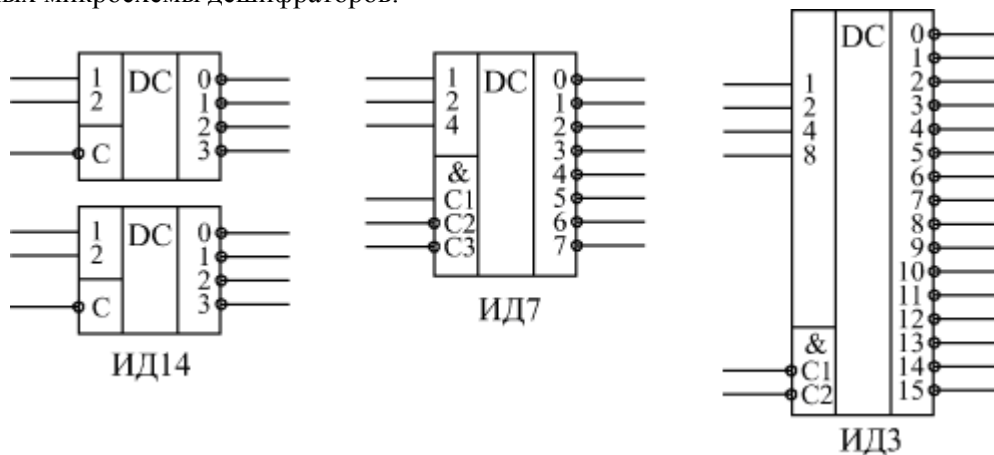


Рис. 2. Примеры микросхем дешифраторов

Код на входах 1, 2, 4, 8 определяет номер активного выхода (вход 1 соответствует младшему разряду кода, вход 8 — старшему разряду кода). Входы разрешения С1, С2, С3 объединены по функции И и имеют указанную на рисунке полярность. Для примера в табл. 5.1 приведена таблица истинности дешифратора ИД7 (3–8). Существуют и дешифраторы 4–10 (например, ИД6), которые обрабатывают не все возможные 16 состояний входного кода, а только первые 10 из них.

Первые три строки таблицы соответствуют запрету выходных сигналов. Разрешением выхода будет единица на входе С1 и нули на входах С2 и С5. Символ “Х” обозначает безразличное состояние данного входа (неважно, нуль или единица). Нижние восемь строк соответствуют разрешению выходных сигналов. Номер активного выхода (на котором формируется нулевой сигнал) определяется кодом на входах 1, 2, 4, причем вход 1 соответствует младшему разряду кода, а вход 4 — старшему разряду кода.

Наиболее типичное применение дешифраторов состоит именно в дешифрировании входных кодов, при этом входы С используются как стробирующие, управляющие сигналы. Номер активного (то есть нулевого) выходного сигнала показывает, какой входной код поступил. Если нужно дешифровать код с большим числом разрядов, то можно объединить несколько микросхем дешифраторов (пример показан на рис.3).

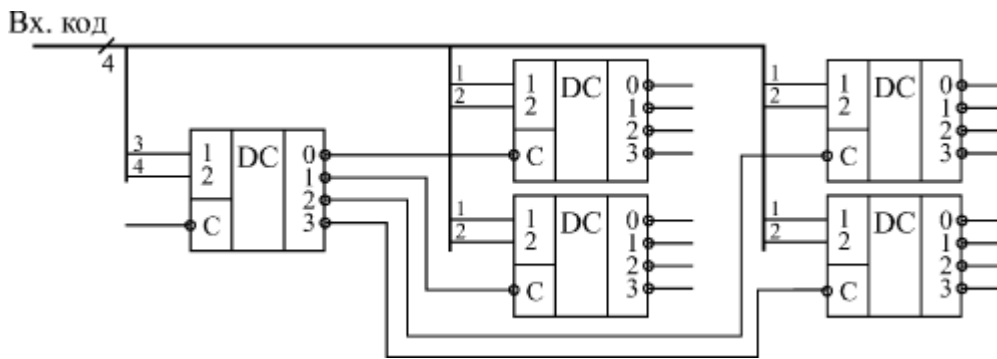


Рис. 3. Увеличение количества разрядов дешифратора

При этом старшие разряды кода подаются на основной дешифратор, выходы которого разрешают работу нескольких дополнительных дешифраторов. На объединенные входы этих дополнительных дешифраторов подаются младшие разряды входного кода. Из пяти микросхем дешифраторов 2–4 можно получить дешифратор 4–16, как показано на рисунке (хотя лучше, конечно, взять готовую микросхему). Точно так же из девяти микросхем 3–8 можно получить дешифратор 6–64, а из семнадцати микросхем 4–16 — дешифратор 8–256.

Модель данной схемы в программе Multisim:

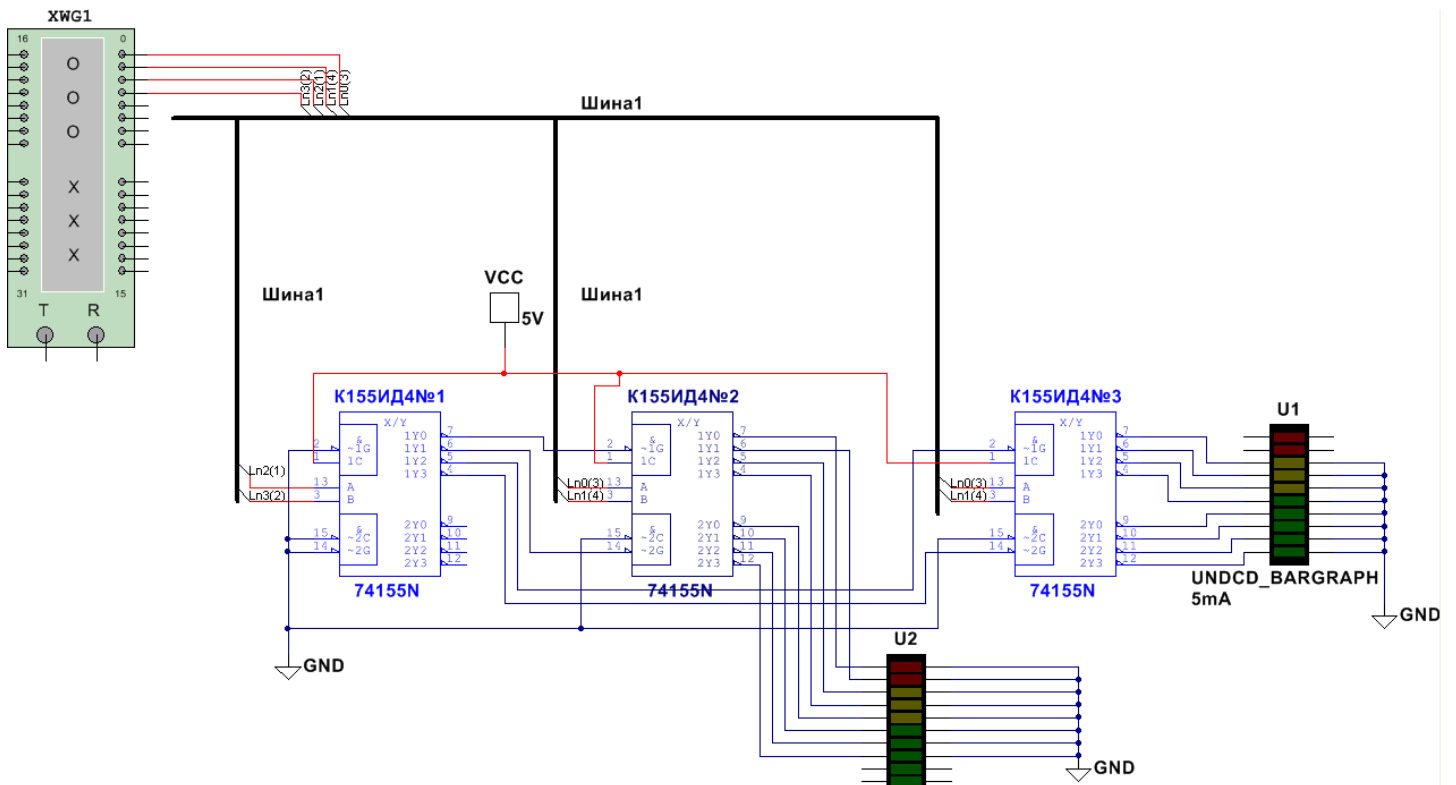


Рис. 4 – Модель схемы дешифратора 4→16 на пяти дешифраторах 2→4 (на трёх ИМС ИД4/74155)

Микросхема логики TTL 74155 является полным аналогом ИМС К155ИД4 (с которой вы будете иметь дело в практической реализации).

ИМС К155ИД4 содержит два дешифратора 2→4 с объединёнными адресными входами А и В и отдельными входами стробирования С и G (см. рис.5).

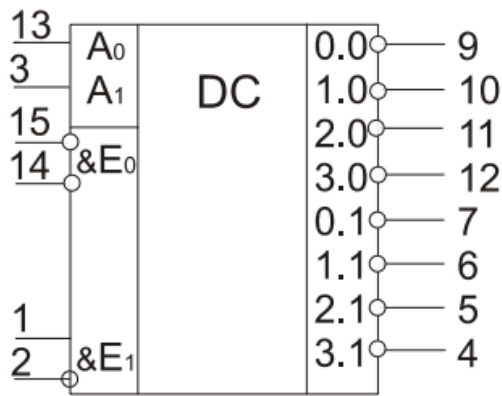


Рис. 5 – УГО К155ИД4

Таблица состояний К155ИД4

| Входы | | | | Выходы | | | | Входы | | | | Выходы | | | |
|-------|-------------|-------|-------|--------|-----|-----|-----|-------------|-------------|-------|-------|--------|-----|-----|-----|
| E_1 | \bar{E}_1 | A_1 | A_0 | 0.1 | 1.1 | 2.1 | 3.1 | \bar{E}_0 | \bar{E}_0 | A_1 | A_0 | 0.0 | 1.0 | 2.0 | 3.0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | X | X | X | 1 | 1 | 1 | 1 | 0 | X | X | X | 1 | 1 | 1 | 1 |
| X | 1 | X | X | 1 | 1 | 1 | 1 | X | 1 | X | X | 1 | 1 | 1 | 1 |

Разрешающей комбинацией для одного дешифратора является наличие лог. «0» на обоих входах системы \bar{E}_0 , а для другого – лог. «1» на одном и лог. «0» на другом входе схемы \bar{E}_0 .

Практическое задание Часть 1 – сборка схемы в программе Multisim

Упростим схему – смоделируем дешифратор 3→8 на трёх дешифраторах двух ИМС К155ИД4 схему в программе Multisim.

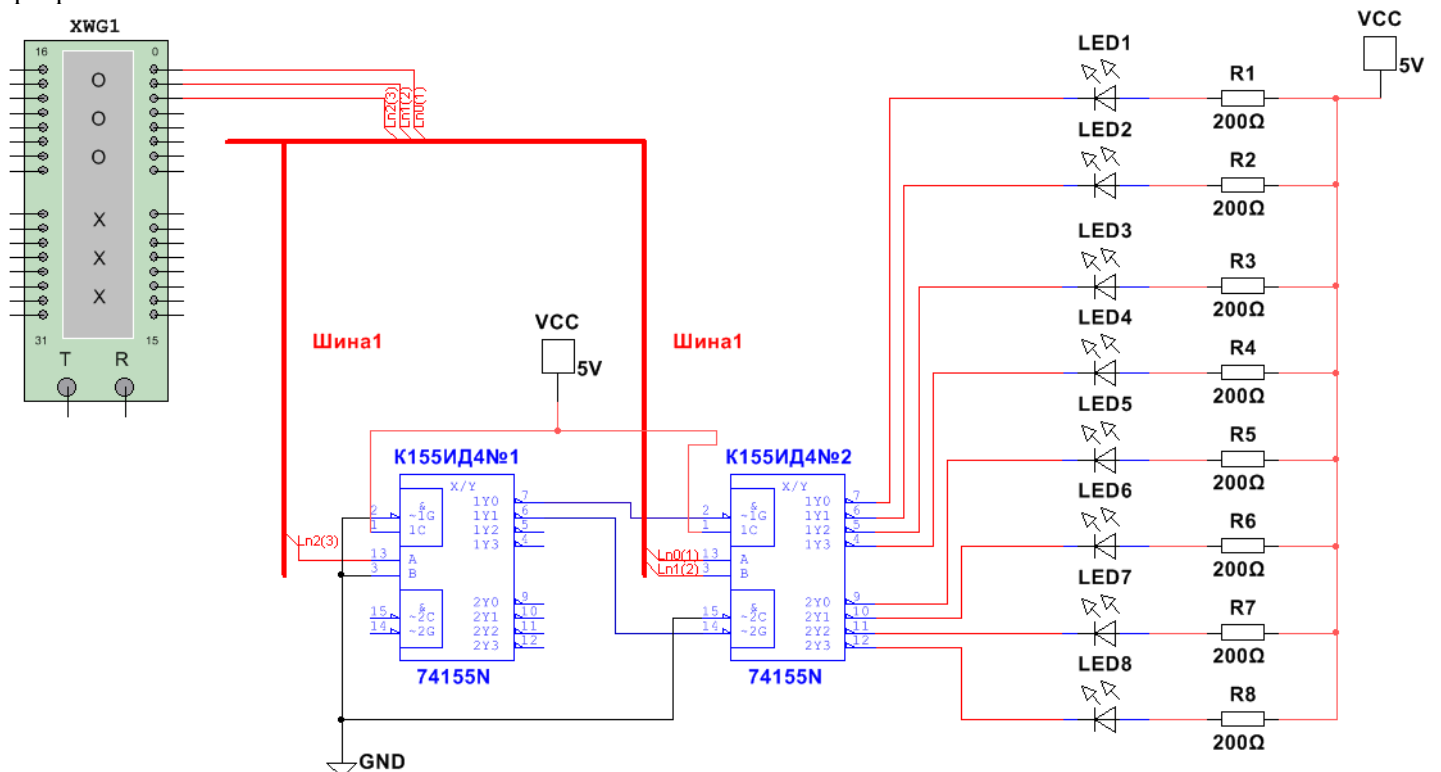
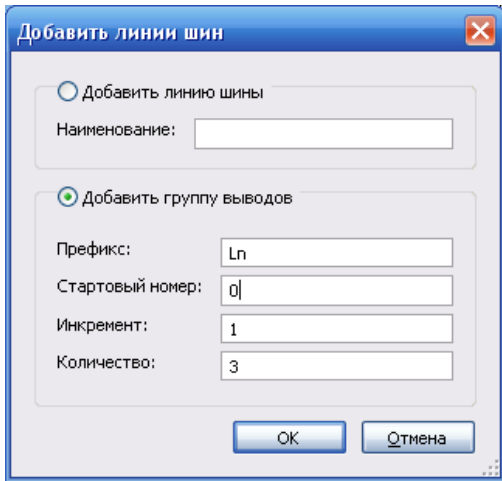


Рис. 6 – Дешифратор 3→8 на двух ИМС К155ИД4/74155

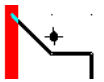
Для вставки шины нажмите Вставить → Шину (как неожиданно ☺) или Ctrl+U. После проведения шины (её не нужно подводить к контактам) зафиксируйте её дабл-кликом.

Рис. 7 – добавление проводов в шину

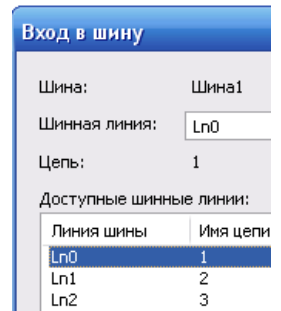


Затем дабл-клик по шине и добавьте группу выводов (см. рис. 7):
 Для разветвления шины снова нажмите Ctrl+U и кликните в нужном месте шины и продлите её куда требуется.

Подводить провода к шине нужно начиная от контакта компонента, проводники входят в шину под наклоном:



После выберите вход в шину:



Часть 2 – Подключение источника сигналов. Схема в Proteus.Isis.

Поскольку у нас нет функционального генератора, а построение делителя частоты сигнала, идущего от мультивибратора на инверторах, на триггерах заёмёт неоправданно (для данной тематики работы) много времени, воспользуемся микроконтроллером ATmega, запрограммированным на подачу сигнала, делённого на 2 и 4 программно.

Соберём в программе Proteus.Isis такую (аналогичную схеме на рис.6, но с микроконтроллером вместо функционального генератора) схему:

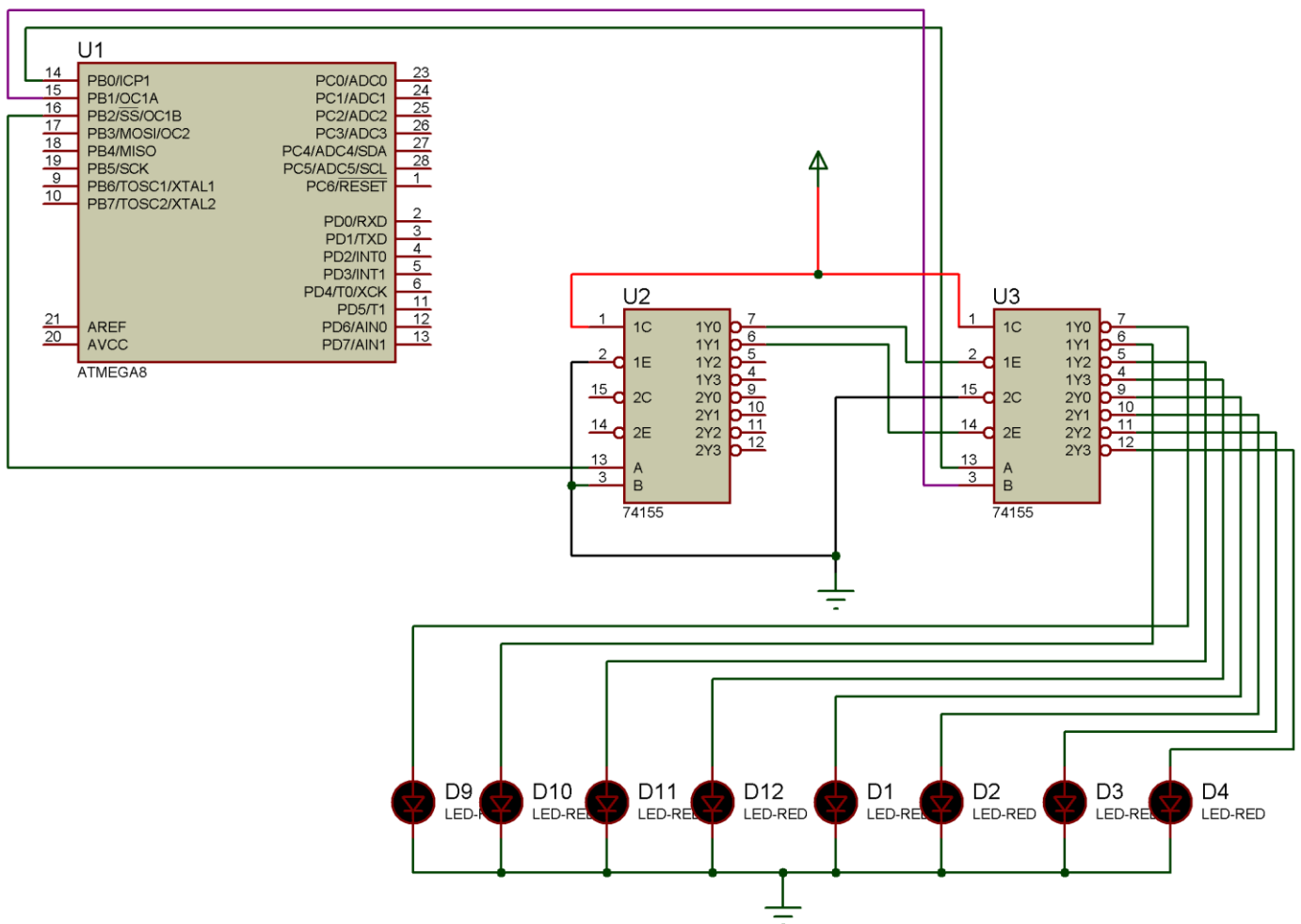


Рис. 8 – схема масштабированного дешифратора в Proteus.Isis

Часть 2 – создание программы

Теперь нам нужно создать программный код. Воспользуемся программой CodeVisionAVR. (см. [лабораторную работу "Программирование "Hallo, World!" для микроконтроллеров"](#))

Листинг программы:

```
#include <io.h>
#include <delay.h>
```

```
void main(void)
{
  DDRB =255;
  PORTB = 0;
  while (1)
  {
    PORTB++;
    if (PORTB>7) PORTB=0;
  }
}
```

Скомпилируем программу и прошьём микроконтроллер в схеме программы Proteus.

Часть 3 – создание программы

Соберём схему 8 на макетной плате

