

# NSU-2023-T05L2e05

В этом упражнении вам необходимо построить 4-битный сумматор BCD. BCD (Binary Coded Decimal) – это формат, часто используемый во встраиваемых контроллерах многих устройств, которые имеют числовые дисплеи. Этот формат устроен следующим образом:

BCD упаковывает в один байт две **десятичные** цифры, каждая закодирована 4 битами. Представление для цифр 0,1,...,9 такое же, как и для соответствующих шестнадцатиричных. Затем эти две группы по 4 бита конкатенируются в один байт. Например, BCD для десятичного числа 13 будет 0b00010011, а для 95 – 0b10010101.

BCD сумматор получает две цифры BCD и входящий перенос, и выдает результат и выходящий перенос. Например, результат сложения переноса 1, 8 (0b1000) и 5 (0b0101) должен быть равен 4 (0b0100) с исходящим переносом 1.

В присоединенном файле `NSU-2023-T05L2e05.circ` предоставлен верхнеуровневый дизайн 2-значного BCD сумматора в виде главной макетной платы `bcd2`. Ваша работа состоит в том, чтобы закончить микросхему `bcd-add`, которая и должна служить 4-разрядным сумматором. Вы можете заметить, что контролы, размещенные на главной макетной плате, полезны для тестирования вашей микросхемы.

**Вот идея для дизайна.** Если мы сложим два 4-битных числа BCD как двоичные, ответ будет корректным в двоичном представлении (с учетом переноса). Но, поскольку мы заинтересованы в десятичном представлении, мы должны преобразовать значения 0xA, 0xB, ..., 0xF в 0x0, 0x1, 0x2, ..., 0x5 соответственно, и выдать перенос равным 1. Это легко достичь, добавляя 0x6 к двоичному результату (но только если сумма равна 0xA или более). Другой случай, требующий внимания – это когда после двоичного сложения уже выставлен перенос, то есть, если мы складывали  $8 + 8$ ,  $8 + 9$ ,  $9 + 9$  или  $9 + 9 + 1$  (последний случай учитывает входящий перенос 1). В этих случаях мы также должны откорректировать сумму, добавляя к ней 6. В конечном итоге, числа могут складываться как двоичные, но когда сумма больше 9 или был исходящий перенос, мы должны добавить 6 к сумме.

Это упражнение не имеет ограничения на число компонентов, но вам может быть полезно знать, что справочное решение тестера построено на четырех вентилях и двух полных 4-разрядных сумматорах. Использование библиотеки *Arithmetic* разрешено, но рекомендуется использовать ваши собственные сумматоры.

## **Как отправлять вашу работу на проверку**

Не перемещайте входные и выходные контакты, потому что Logisim присоединяет к ним тестовую схему, основываясь на их положении, а не по имени (это неудобно, но мы ничего не можем с этим сделать).

Проверьте устройство, нажимая входные контакты при помощи ручных контролов и записывая ваши наблюдения. Когда вы уверены, что устройство работает, ответьте на это сообщение, присоединив файл схемы с вашим решением. Не меняйте строку темы (Subject:); убедитесь, что номер билета в этой строке (т.е. часть, которая начинается со \*\*) не изменялся.