



Рис.3. Управление динамиком в ПЭВМ типа IBM PC

DC - дешифратор 3 в 8. Входы G1 и G2 - разрешающие (если на них пришли нули, DC работает, в противном случае ни один из его выходов не реагирует). Кружки на выходе DC означает, что **активный уровень сигнала нулевой** (если сигнал есть, то на одном из выходов DC 0, если нет, то 1).

IOP 8255 - параллельный порт. Общий смысл назначения - буферное устройство, "шлюз" между шиной данных и какими-то внешними уст-вами, которые подключаются к одному из портов А, В или С. Умеет получать с шины данных байты или по запросу передавать их на системную шину данных, а принятые байты удерживать на одном из выходов, либо анализировать состояние одного из своих выходов и по запросу передавать байты от внешнего мира на шину данных.

СТ - таймер-счётчик. Умеет на одном из своих выходных каналов (0, 1 и 2) формировать различного вида (6 вар) сигналы. В нашем случае - последовательность импульсов заданной частоты. Чтобы там формировался нужный нам сигнал, нужно таймер запрограммировать (по шине данных закатать в него управляющие слова). На вход C2 подаётся опорная частота. В результате на выходе СТ способен выдавать меандровый сигнал (получается путём деления опорной частоты).

SP - динамик. Сигнал с СТ подаётся на звуковую карту, и мы его слышим. Внутри СТ есть катушка, которая связана с мембраной, она находится в магнитном поле. Если через неё пропустить ток, она двинется либо в одну сторону, либо в другую. Если тока нет, она вернётся в первоначальное положение. Подавая переменный ток на катушку, мы получаем переменное

движение мембраны -> звук. ЛИБО: имеется катушка и ферритовый стаканчик, в котором она расположена. В результате магн. поле сосредоточено перед этой конструкцией. На катушке лежит металлич. пластинка из магнитного материала. Если подавать переменное эл. поле на катушку, то пластинка будет то притягиваться, то отталкиваться от ферритового стаканчика, создавая колебания -> звук.

**Чтобы был звук, должен быть переменный ток!**

Элемент И, в частности, выполняет буферную функцию. Если на его выходе 0, то это значит, что течёт переменный ток, мембрана колеблется, иначе нет. Подавая нули и единицы, мы то качаем мембрану в крайнем положении, то отпускаем в нейтральное и слышим звук.

XData - шина данных (8-разрядная), XAddress - шина адреса (20-разрядная).

Процессор подключён с помощью дополнительной логики к шинам адреса и данных, **на схеме его нет!** Он формирует сигнал на этих шинах (адрес и данные), а также сигналы на !IORD (процессор желает что-то прочитать) и !IOWR (процессор желает что-то записать в порт ввода/вывода), управляющие направлением передачи данных. Если процессор пишет, то он передаёт данные на шину данных, а кто-то с неё их получает (чей адрес выставлен). Если процессор читает, то кто-то должен на шину данных сформировать сигнал, а процессор его получает.

**Активный уровень сигналов чтения и записи - нулевой (если 0, значит условие выставлено, если 1, значит нет)!**

У IOP и CT два адресных входа - A0 и A1 (младшие разряды адреса). Чтобы микросхема восприняла, что обращаются именно к ней, должен прийти сигнал !CS - сигнал активации. Если на !CS активный нулевой уровень, то дальше микросхема проверяет, что у неё на адресных входах, в каком направлении идёт передача данных и что у неё на входах данных или что она на них выдаст. Если !CS=1, то микросхеме всё равно, что пришло на !RD, !WR, A0, A1 и что там на шине данных - переводит свои выводы в 3-е, высокоимпедансное состояние.

!CS включает или выключает шинный интерфейс (шины данных, адреса, а также !IORD и !IOWR). **Если !CS не активен, то микросхема не общается с шиной данных, но при этом она продолжает заниматься своими делами!** К примеру, IOP на своих выходах держит тот байт, который ему был запрограммирован и занесён в выходы. CT, в свою очередь, продолжает

генерировать сигнал с заданной частотой, но по шине данных он ничего не передаёт.

Для того, чтобы передать какой-то сигнал микросхеме или чтобы прочесть какую-то инфу из неё, в программе внутри процессора надо использовать команды обращения к портам - команды IN и OUT (в них указывается адрес порта конкретной микросхемы).

OUT 61h, al - в порт 61h передать инфу из регистра al. Будет сформирован сигнал !IOWR, процессор на шине данных выставит то, что лежало в регистре al, а на шине адреса выставляется код 61h -> 0110 0001 (**тетрады в двоичной системе счисления**)

0110 000**1**

**Красные биты - A0 и A1.** Кодируют номер порта, с которым будем взаимодействовать (в нашем случае это порт В: порт А - 00, порт В - 01, порт С - 10). 11 - регистр управляющего слова, задаёт режим работы микросхемы.

На самом деле адрес порта **0000 0000 0110 0001** (биты считаются справа налево) - адрес микросхемы IOP 8255.

**Зелёные биты (8-ой и 9-ый)** - разрешающие сигналы XA8 и XA9 (G1 и G2)

**Оранжевые (5-7-ой биты)** - номер выхода дешифратора (в данном случае выход 3 - вход !CS на микросхеме IOP 8255)

**Фиолетовые биты могут быть любыми.**

То есть, обращение идёт к IOP 8255, если биты с 5-го по 9-ый равны 00011 - диапазон адресов от 60h до 7fh (для схемы так, как она есть).

На самом деле диапазон адресов от 60h до 63h для компьютера.

Если же адрес порта выглядит как xxxx xx00 010x xxxx (с 5-го по 7-ой бит - обращение к выходу 2 дешифратора), то активен вход !CS на счётчике. **Диапазон адресов от 40 до 43h.**

Фиолетовые биты - A0 и A1. Кодируют номер порта, с которым будем взаимодействовать (нулевой, 1-ый или 2-ой). Нас интересует 2-ой, то есть код 10 (адрес порта 42h). 43h - режим работы 2-го канала (регистр управляющего слова).

Если на !CS счётчика равен 1, то микросхему не интересует, что на остальных входах. Входы, кроме входов с D по !WR - те выводы, которым СТ управляет по своему усмотрению в соответствии с режимом работы.

Выходом OUT2 управляют входы CE2 (разрешающий) и C2 (вход синхронизации).

Чтобы получилась нужная частота на OUT2 (100, 200, 300 Гц и тд), по шине данных нужно загрузить коэффициент деления  $K=1.19 \text{ МГц}/f$  (целое число, округлённый результат деления - двухбайтовое число). Чтобы его загрузить в СТ, нужно либо загрузить сначала старший, затем младший байт, либо наоборот.

Как работает K: в счётчик внутри СТ загружается K. С каждым приходом импульса K уменьшается. После K импульсов дойдёт до граничного события. По этому событию выход OUT2 перейдёт из 0 в 1 или из 1 в 0 **(на каждые K импульсов сигнала входной частоты происходит изменение сигнала на выходе).**

**Если канал OUT2 выключен (на CE2 0), то на нём постоянно 1.**

Выходы микросхемы IOP PB0 (заведён на вход CE2 счётчика) и PB1 (заведён на элемент И после СТ).

**Чтобы раздавался звук, нужно:**

1) запрограммировать таймер (занести туда K, чтобы на выходе получалась нужная частота)

2) чтобы таймер начал генерировать сигналы и чтобы они проходили через И, нужно, чтобы на PB0 и PB1 была 1 (то есть в порту PB по адресу 61h сформировать такое значение, чтобы в младших двух битах были 1).

**Если же PB0=0, то таймеру запрещено считать! Если PB1=0, то микросхема И будет выдавать 0 на выходе - импульсы не пройдут сквозь неё.**

Остальные же входы порта PB присоединены к другим узлам и устройствам внутри компа. В частности, они управляют передачей данных с клавиатуры.

Читаем то, что находится в порту В (команда IN al, 61h), потом изменяем два младших бита в прочитанном значении, и затем команда OUT 61h, al - модифицированное значение отправляем в порт 61h.

**А как ещё, имея перед глазами эту схему, можно управлять СТ??**

Ответ: Таймер нужно не использовать (PB0=0). Тогда можно программно управлять выходом PB1 и программно перекидывать микросхему И из 0 в 1 и тоже формировать выходной сигнал.