

Файл **models_ex_iqMath.mbp** содержит примеры, показывающие работу, блоков группы Math.

Примеры предназначены для наглядного изучения пользователем функций блоков и возможности экспериментально усвоить, как работают приведённые в примере блоки.

В группе **Math** находятся блоки, предназначенные для проведения различных математических операций над сигналами. Корневое поле набора файла примера выглядит:

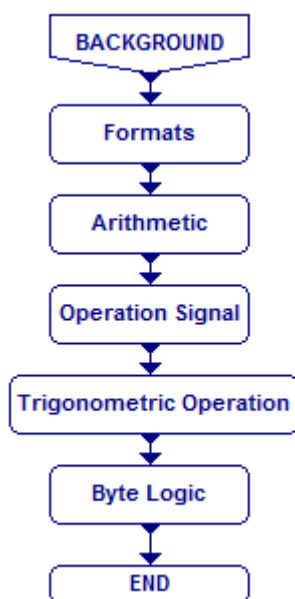


Рис. 1. Корневое поле набора схемы примера

Formats

Формула содержит пример преобразования форматов.

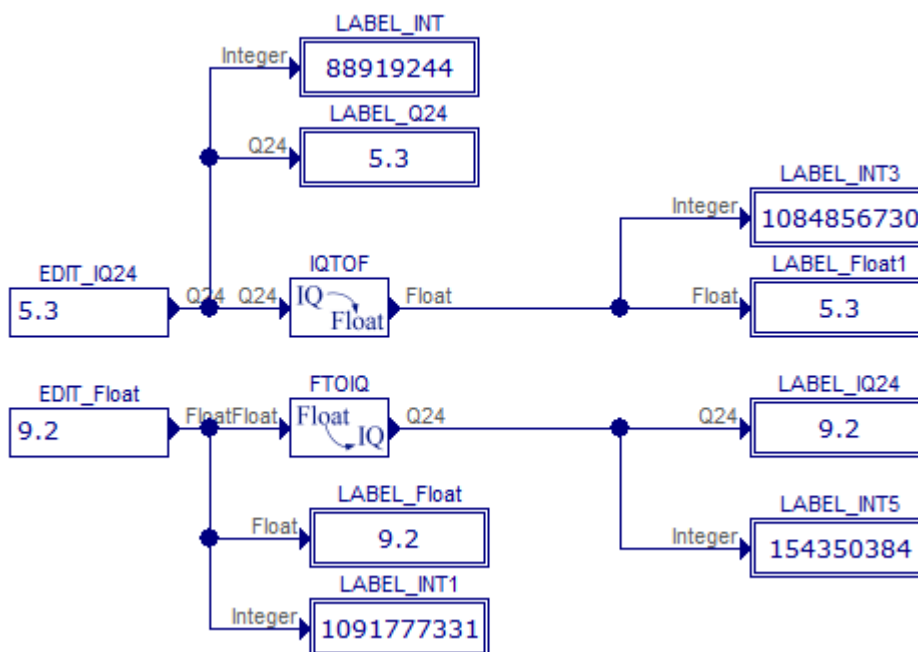


Рис. 2. Операции преобразования форматов

Пример показывает как представляется в процессоре одно и то же число в формате IQ24 и Float. И преобразование числа из формата IQ24 в Float, и из Float в IQ24.

После запуска симуляции можно изменить значения в органе управления Edit и наблюдать полученный результат.

Примечание

Микроконтроллеры Texas Instruments серии C2000 работают с 32-битовым словом, позволяющим представить положительные числа в диапазоне от 0 до $+2^{32}$ или числа со знаком (положительные или отрицательные) в диапазоне от -2^{31} до $(2^{31}-1)$ в дополнительном коде.

Для представления десятичных дробей используются числа, которые всем известны по работе с обычным калькулятором. Такие числа называются числами с плавающей точкой. Так как микроконтроллер работает с целыми числами для представления чисел с плавающей точкой необходимо перевести число в формат с псевдоплавающей точкой.

Если проводить перевод вручную, то сначала необходимо выбрать точность представления числа, для этого задаётся часть бит от слова для хранения дробной части, остальная часть бит будет содержать целое число. Объясним на примере.

Пусть необходимо представить число π в формате с плавающей точкой. Пусть под дробную часть выделено 24 бита, под целую 7 бит (максимальное положительное число $2^{31}-1$). Для перевода числа в понятный для микроконтроллера вид необходимо произвести умножение:

$$3.1415926535897932384626433832795 \cdot 2^{24} = 52707178,533$$

Так как в слово может содержать только целое число - дробная часть отбрасывается. Полученное число есть число π с псевдоплавающей точкой. Для проведения арифметических операций необходимо чтобы все оперируемые числа были в формате 24. Если в микроконтроллере нет специальных функций для работы с числами представленными в формате с псевдоплавающей точкой, то ответственность за соблюдение формата ложится на пользователя. Также следует сказать, что под целую часть от слова осталось 7 бит, что означает максимальное число которое можно представить это 2^7 (на самом деле можно представить от -128 до 127,9(9)9 так как максимальное целое

положительное число $2^{31}-1$). Для перевода из формата с псевдоплавающей точкой в привычный вид необходимо разделить на 2^{24} .

$$52707178/2^{24}=3.14159262180328$$

Видим, что появилась погрешность представления числа величиной $1/2^{24}=6.0e-008$. Поэтому для операций с дробными цифрами, представленными в виде числа с псевдоплавающей точкой необходимо следить за необходимой точностью числа и не превышает ли целая часть числа максимальное число, которое может быть представлено оставшемся числом бит от слова. При превышении целой части произойдет переполнение формата.

Если прибавить к этому числу единицу то получим: -2^{31} :

$$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111 = 2147483647 = 2^{32}-1$$

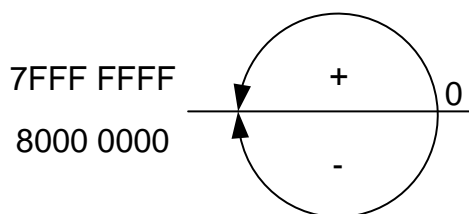
+1

$$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000 = -2147483647 = 2^{32}$$

Если случится переполнение формата в процессе вычисления – результат вычислений будет неверным.

В среде MexBIOS Development Studio используется представление числа, предложенное фирмой Texas Instruments библиотека функций IQmath для работы числами представленными в формате псевдоплавающей точки. Формат числа представленный, например, с точностью 24 бит называется формат IQ24. Пользователю необходимо указывать формат числа при задании переменной и отображении её с помощью графических средств. Делается это с помощью установке параметра Format в блоках, где присутствует этот параметр. Также отобразить числовой формат сигнала можно с помощью клавиши F2.

Большинство блоков рассчитывается в формате 24, что достаточно для решения большинства задач управления.



Arithmetic

Формула **Arithmetic** содержит несколько схем, показывающих работу блоков арифметических операций:

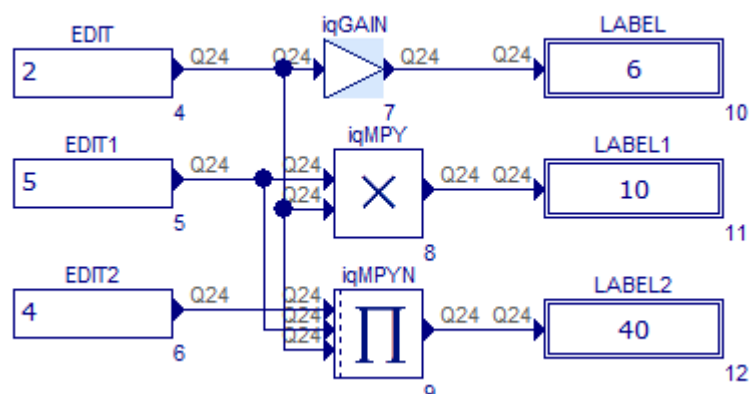


Рис. 3. Арифметические операции

Схема представленная выше показывает работу арифметических операций сложения, вычитания, умножения, деления.

После запуска симуляции можно изменить значения в органе управления Edit и наблюдать полученный результат.

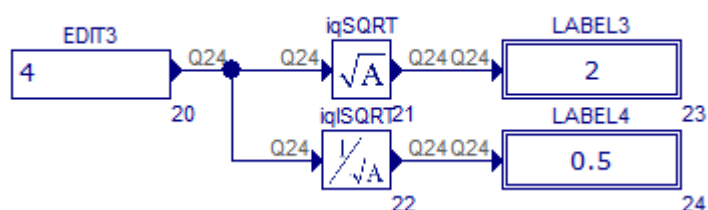


Рис. 4. Арифметические операции

Схема показывает работу блока извлечения квадратного корня и функции вычисления величины обратной квадратному корню из числа.

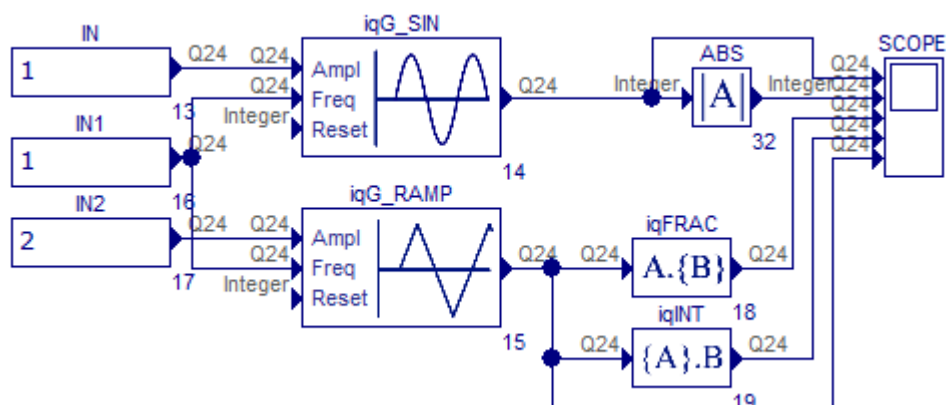


Рис. 5. Схема показывающая работу блоков ABS, FRAC, INT.

Схема показывает работу блоков **ABS** (модуль числа), **FRAC** (выделение дробной части числа), **INT** (выделение целой части числа). Блоки **QG_SIN**, **QG_RAMP** вспомогательные блоки. Для наглядности иллюстрации работы блоков сигналы были смещены с помощью настройки **Offset** осциллографа.

Блоки подключены к буферному осциллографу, после запуска симуляции необходимо нажать на кнопку запуска отображения графиков на осциллографе.

Полученные графики:

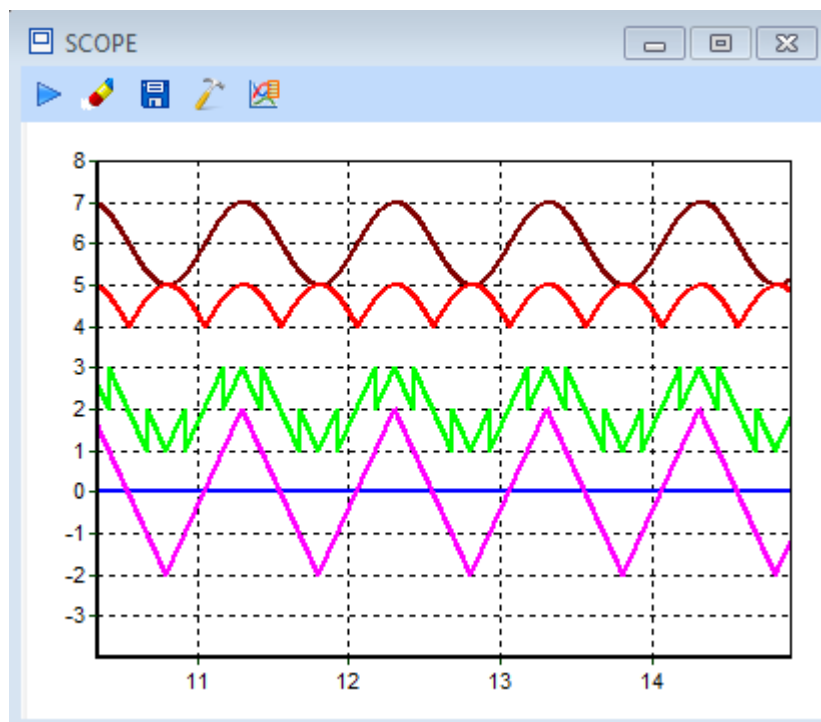


Рис. 6. Осциллограмма работы блоков ABS, FRAC, INT

Operations_signals

Формула **Operations_signals** содержит схему, показывающих работу блоков специальных операций над сигналами:

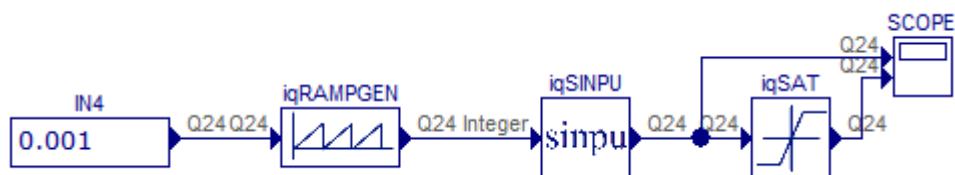


Рис. 7. Схема, показывающая работу блока SAT (ограничение входного сигнала на заданном уровне)

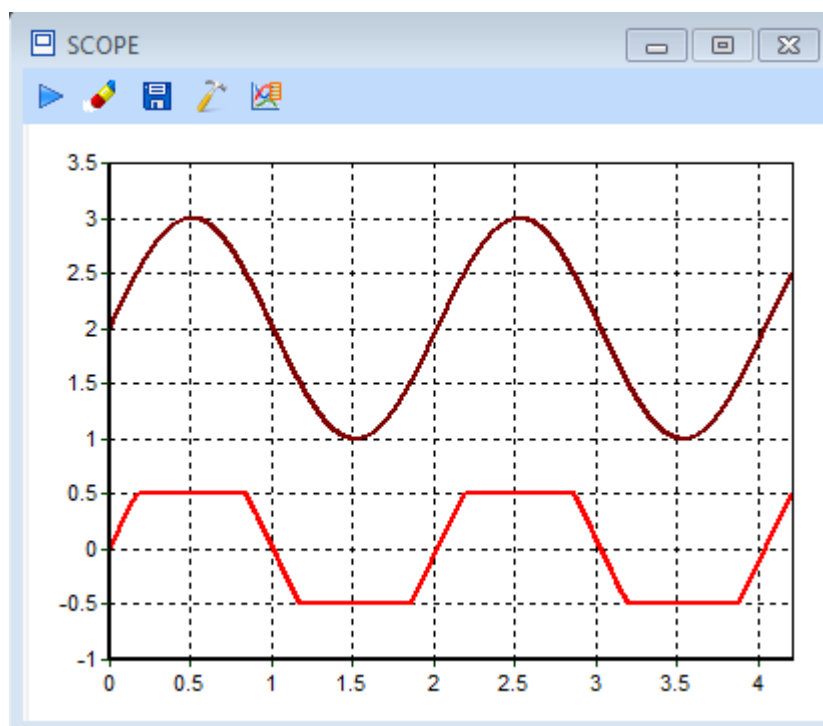


Рис. 8. Результат работы блоков SAT

Trigonometric_Operation

Формула **Trigonometric_Operation** содержит формулы по вычислению тригонометрических функций:

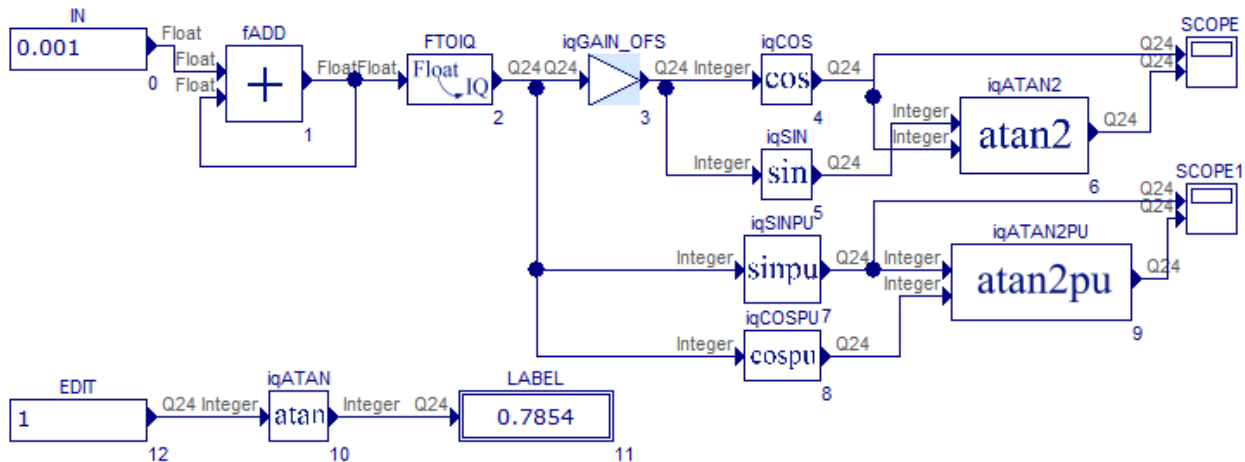


Рис.9. Схема, показывающая работу блоков тригонометрических функций

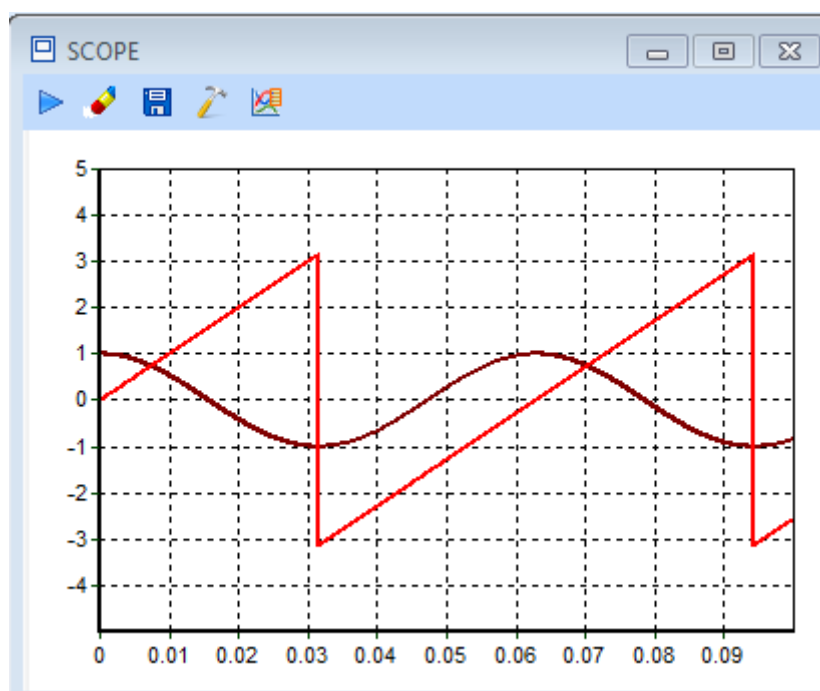


Рис10. Результат работы блоков iqCOS и iqATAN2

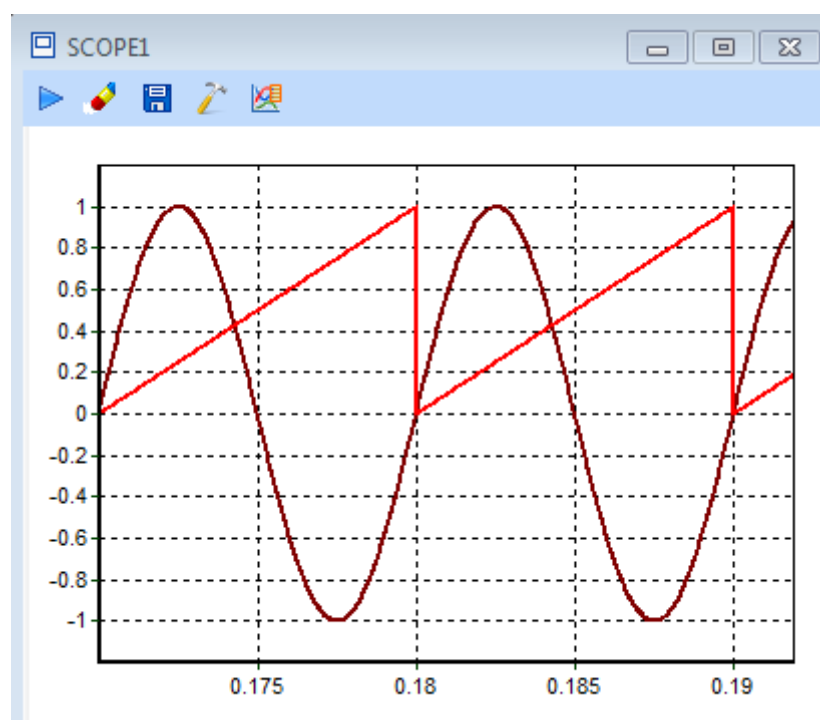


Рис.11. Результат работы блоков iqSINPU и iqATAN2PU

Byte_Logic

Формула **Operations_signals** содержит несколько схем, показывающих работу блоков битовых операций:

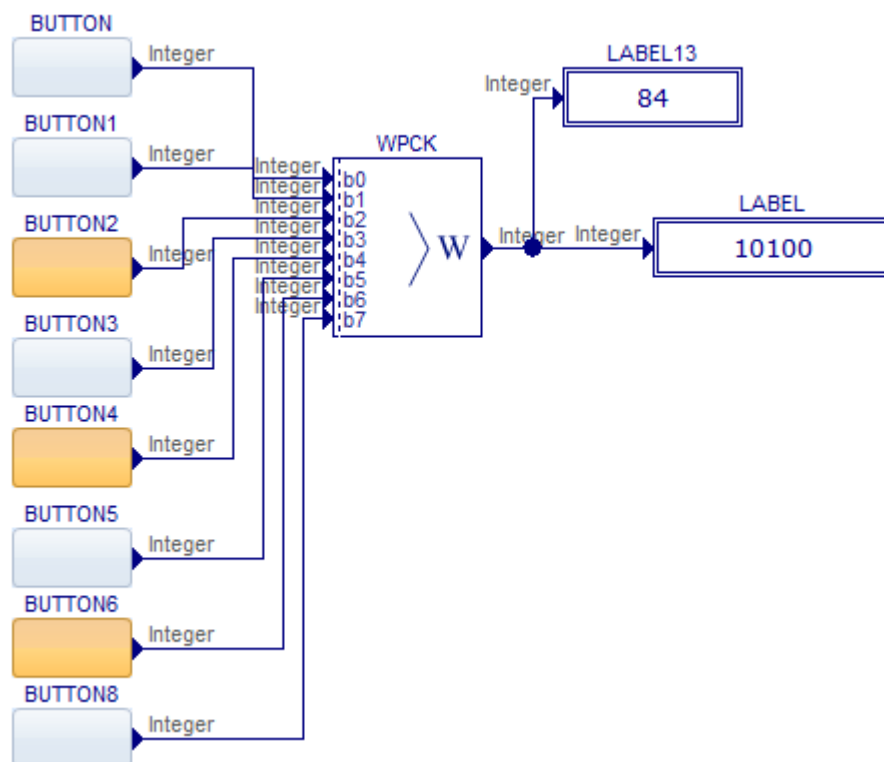


Рис. 9. Схема, показывающая работу блока WPKC

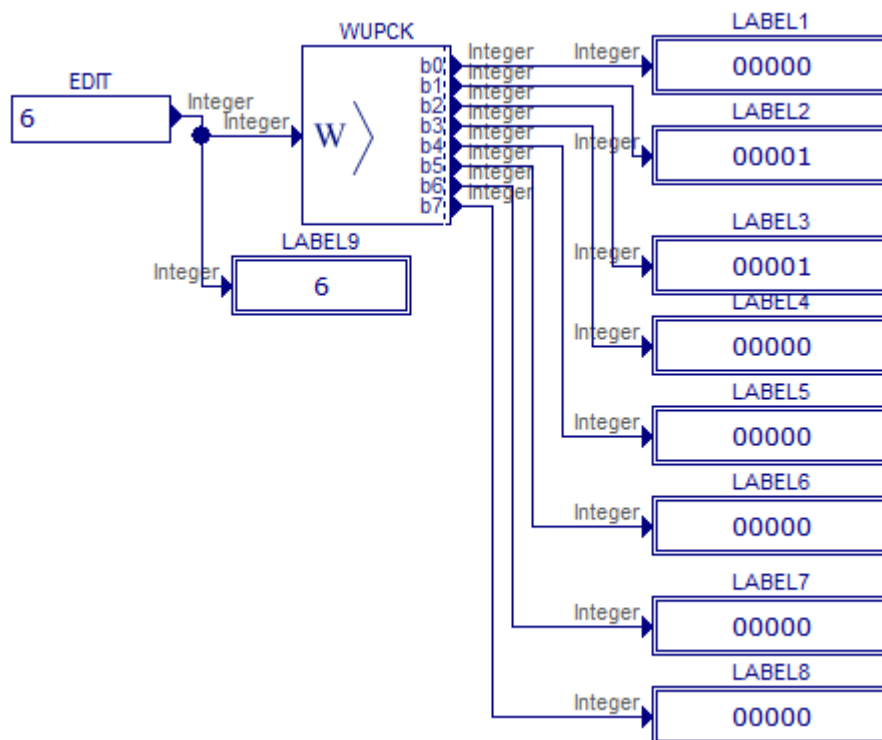


Рис. 10. Схема, показывающая работу блока WUPCK

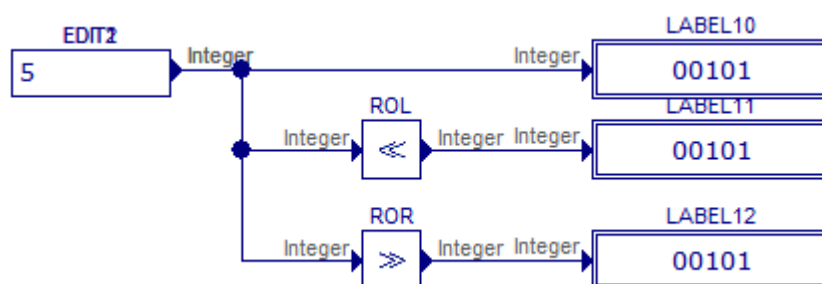


Рис. 11. Схема, показывающая работу блоков ROL (сдвиг влево) и ROR (сдвиг вправо)