



ОБ ОДНОМ ПОДХОДЕ К ИЗУЧЕНИЮ ТЕМЫ «АРИФМЕТИЧЕСКИЕ СХЕМЫ: ПРОЕКТИРОВАНИЕ, СИМУЛЯЦИЯ И ОТЛАДКА»

Долинский М. С.¹, канд. техн. наук, доцент, dolinsky@gsu.by

¹Гомельский государственный университет им. Ф. Скорины, ул. Кирова, д. 119,
246019, Гомель, Республика Беларусь

Аннотация

В статье описывается технология изучения темы «Арифметические схемы: проектирование, симуляция и отладка» в курсах основ цифровой электроники на первом / втором курсах с помощью системы дистанционного обучения DL.GSU.BY и системы высокоуровневого проектирования цифровых устройств HLCCAD, разработанных в ГГУ им. Ф. Скорины под руководством автора. К основным достоинствам технологии относятся обучение, динамически адаптирующееся к конкретному студенту, многолетнее эффективное применение на практике.

Ключевые слова: основы цифровой электроники, арифметическая схема, симуляция, отладка, DL.GSU.BY, HLCCAD.

Цитирование: Долинский М. С. Об одном подходе к изучению темы «Арифметические схемы: проектирование, симуляция и отладка» // Компьютерные инструменты в образовании. 2023. № 4. С. 83–98 doi:10.32603/2071-2340-2023-4-83-98

1. ВВЕДЕНИЕ

Смешанное обучение (сочетание очного и дистанционного) активно развивалось и до COVID, однако с приходом этой болезни стало во много раз актуальнее. В работах [1, 2] обосновывается необходимость перехода к смешанному обучению, которое комбинирует возможности традиционного обучения и новых информационных технологий. В работах [3–8] описываются примеры такого смешанного обучения различным предметам: математике [3], английскому языку [4], географии [5], основам медицинских знаний [6], анатомии [7], химии [8].

Автор активно использует смешанное обучение для дисциплин, связанных с обучением программированию и основам цифровой электроники на факультете математики и технологий программирования ГГУ им. Ф. Скорины [9–11].

Для повышения эффективности смешанного обучения под руководством автора были разработаны система проектирования симуляции и отладки цифровых схем [12] и инструментальная система дистанционного обучения DL.GSU.BY [13].

Ранее автор описывал подходы к смешанному обучению для тем «Логические и комбинационные схемы» [14] и «Синтез комбинационных схем с использованием карт Карно» [15].

Данная работа представляет авторский подход к обучению теме «Арифметические схемы: проектирование, симуляция и отладка».

2. ФОРМУЛИРОВКА ПРОБЛЕМЫ

Автор много лет занимается преподаванием дисциплин, связанных с цифровой электроникой на математическом факультете ГГУ им. Ф. Скорины.

Всё это время стремительно меняются условия обучения — совершенствуются компьютерные средства и Интернет-технологии, в среднем снижается уровень подготовки и мотивации студентов, одновременно растут требования к знаниям и умениям выпускников вузов. Это приводит к необходимости менять процесс обучения так, чтобы, с одной стороны, теоретические знания давались более подробно и существенно более простым языком, чем это было принято ранее, с другой стороны, для закрепления полученных знаний использовались не только и не столько семинары со студентами, сколько решение ими практических заданий.

3. РЕШЕНИЕ ПРОБЛЕМЫ

3.1. Теоретические предпосылки

Данная работа посвящена описанию модифицированной методики обучения теме «Арифметические вычисления», учитывающей описанные выше тенденции. Перед этой темой студенты уже изучили темы «Введение в предмет», «Синтез комбинационных схем по таблицам истинности», «Комбинационные схемы».

Тема «Введение в предмет», в частности, предназначена и для освоения системы HLCCAD, с помощью которой можно редактировать, симулировать и отлаживать функциональные схемы цифровых устройств, и для ознакомления с основными логическими операциями NOT, AND, OR, XOR и соответствующими базовыми логическими элементами.

Тема «Синтез комбинационных схем по таблицам истинности», с одной стороны, закрепляет знания и навыки применения базовых логических элементов NOT, AND, OR, XOR для решения задач на проектирование цифровых устройств, с другой, — развивает навыки анализа (и мысленной симуляции) схем, составленных из этих базовых логических элементов, а с третьей стороны, знакомит с методом минимизации логических функций посредством карт Карно.

Тема «Комбинационные схемы» знакомит студентов с базовыми комбинационными схемами (дешифратор, шифратор, мультиплексор, сумматор), используемыми для проектирования и анализа цифровых устройств.

3.2. Суть заданий на проектирование

При изучении основ цифровой электроники очень важно, чтобы студенты получили хорошие практические навыки анализа и синтеза разнообразных цифровых устройств. Существенным подспорьем в этом являются средства симуляции и отладки. Для этих целей в ГГУ им. Ф. Скорины уже много лет разрабатывается, эксплуатируется и используется в учебном процессе система HLCCAD. Для выработки навыков создания схем цифровых устройств в системе HLCCAD и поиска ошибок в них, в основном, и введена в изучение тема «Арифметические схемы». В общем виде задание, которое должен выполнить студент,

выглядит так: если некоторое условие выполняется, вычислить одно арифметическое выражение, если условие не выполняется, вычислить другое арифметическое выражение. При этом в обоих выражениях используются, как правило, не менее четырёх переменных, две из которых имеют разрядность два байта, а две другие — один байт. Пример такого задания приведен на рисунке 1.

Задача № 1

Имя HLCCAD проекта: Arifm.PRD
Входное устройство: Arifm

Разработать устройство, которое вычисляет значение выражения
 $a/b+c-d*a$, если $a*d<0$
RES=<
 $\backslash b-(c+a/d)*b$, если $a*d\geq 0$

Информация о входах и выходах представлена в таблице.

Название	Размерность	Тип
c	8	ВХОД
d	8	ВХОД
a	16	ВХОД
b	16	ВХОД
RES	16	ВЫХОД

Примеры:
a= 8 a=10
b= 3 b= 7
c= 5 c=-9
d=-2 d= 3

RES=23 RES=49

Рис. 1. Пример задания темы «Арифметические вычисления»

При внимательном отношении к размерностям данных задача фактически заключается в том, чтобы вытащить на схему нужные блоки арифметических операций и корректно соединить их выходы и входы в соответствии с порядком действий в выражениях.

Очевидно, что для студента с любым уровнем предварительной подготовки задание понятно и посильно — надо просто нарисовать схему, которая выполняет указанные вычисления. При этом время выполнения задания как раз и будет определяться навыками поиска и устранения ошибок. А сам процесс выполнения таких заданий практически в чистом виде посвящён отработке технологий и выработке практических навыков поиска и устранения ошибок в схемах.

3.3. Библиотека компонентов Standard

В рамках темы «Арифметические вычисления» необходимо использовать такие базовые цифровые блоки как сумматор, вычитатель, умножитель, делитель. Кроме того, для выбора одного из вычисленных результатов требуется использовать мультиплексор. В некоторых заданиях требуется выбирать не из двух выражений, а из трёх. Это приводит к необходимости строить схему, которая будет вычислять значения, подаваемые на

адресные линии мультиплектора. На рисунках 2–6 представлены элементы библиотеки компонент Standard системы HLCCAD, используемые в заданиях на тему «Арифметические вычисления»: мультиплексор, сумматор, константа.



Рис. 2. Мультиплексор MSb8x2

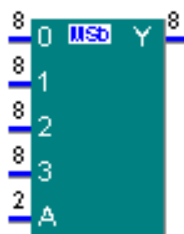


Рис. 3. Мультиплексор MSb8x4

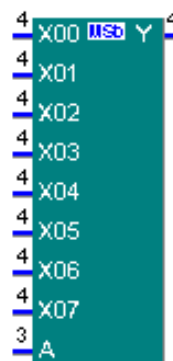


Рис. 4. Мультиплексор MSb4x8

Мультиплексор MS8b8x2 (рис 2) — это так называемый шинный мультиплексор. Он имеет на выходе 8-битную шину и на входе две 8-битных шины данных (0 и 1) и одну адресную линию A.

Если A = 0, то на выходную шину Y побитно передаются значения с входной шины 0.

Если A = 1, то на выходную шину Y побитно передаются значения с входной шины 1.

Мультиплексор MSb8x4 (рис 3) также является шинным мультиплексором. На выходе у него 8-битная шина Y, на входе — четыре 8-битных шины данных (0, 1, 2, 3) и 2-битная адресная шина.

Если на адресном входе стоит 00, то данные с шины 0 передаются на выход.

Если на адресном входе стоит 01 (левый бит старший), то данные с шины 1 передаются на выход.

Если на адресном входе стоит 10, то данные с шины 2 передаются на выход.

Если на адресном входе стоит 11, то данные с шины 3 передаются на выход.

Кроме того, любой из этих мультиплекторов может быть конвертирован в любой произвольный мультиплексор изменением количества разрядов в адресной линии и количества разрядов в шинах данных. Например, на рисунке 4 показан мультиплексор MSb4x8, имеющий 3 адресные линии и 8 четырёхбитных шин данных. На выходе также четырёхбитная шина. Первое число в обозначении (4) указывает разрядность входных и выходных шин данных, а второе число (8) указывает, сколько будет входных шин данных (одновременно определяя количество требуемых адресных линий как логарифм по основанию 2 от количества входных шин данных: 3 есть логарифм 8 по основанию 2).

Сумматор (рис. 5) по умолчанию является 8-разрядным. Однако эта разрядность (как и для всех устройств из проекта Standard) может быть произвольно изменена, кроме того, можно не отображать (в случае ненадобности) контакты переноса, переполнения и суммы.

Константы (рис. 6) могут задаваться несколькими способами. Во-первых, их можно формировать как объединение последовательностей единиц и нулей, используя, соответственно, константы единиц и нулей нужной разрядности. Кроме того, имеется элемент константа (const), в параметрах которого можно указать систему счисления, в которой задаётся константа (до первого символа '#' — по умолчанию — десятичная), а затем саму константу (между двумя символами '#'). Заметим, что не поддерживается задание отри-

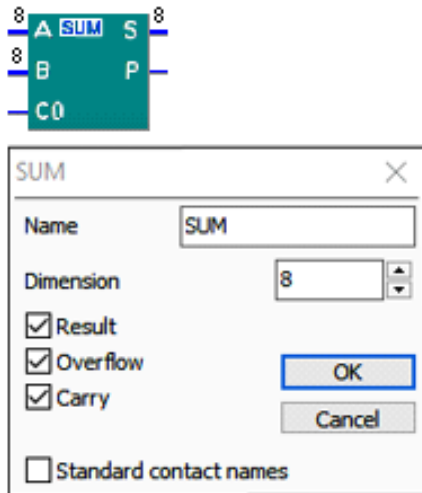


Рис. 5. Сумматор

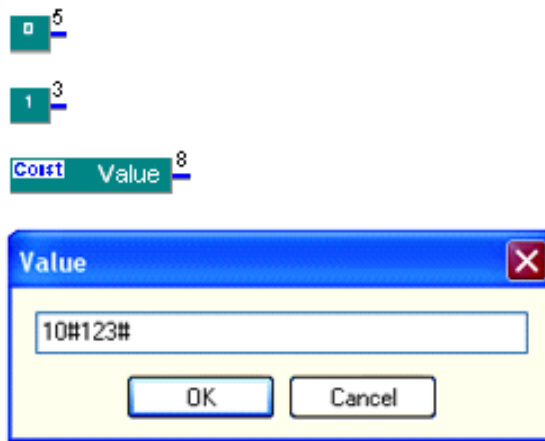


Рис. 6. Константы

цательных чисел в десятичной системе счисления. Это требует от студентов понимания, как представляются отрицательные числа в дополнительном коде и как они записываются в двоичной, восьмеричной или шестнадцатеричной системе счисления.

3.4. Библиотека схем SignedArithm

Компоненты библиотеки Standard (MUL, DIV, CMP) корректно работают только с положительными числами (то есть вычисляют результат правильно, только если на входе положительные числа). В то же время, задания на вычисления арифметических выражений требуют правильной работы как с положительными, так и с отрицательными числами. Для разрешения этой коллизии силами студентов была разработана библиотека схем знаковой арифметики SignedArithm. Она включает в себя следующие схемы с фиксированной разрядностью входных и выходных шин вычитание (iSUB), умножение (iMUL), деление (iDIV), сравнение (iCMP), повышение разрядности (распространение знака: CBW, CWD), представленные на рисунке 7.

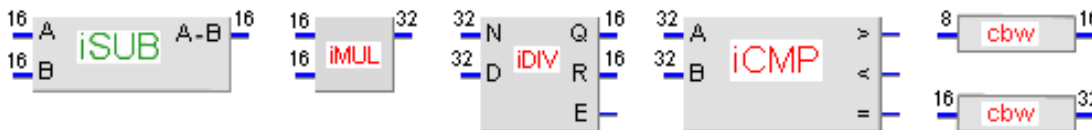


Рис. 7. Схемы знаковой арифметики

При решении задач студенты просто вытаскивают эти рисунки блоков на свои схемы. В то же время, кликнув по любому из них, каждый может ознакомиться, как именно выполнена соответствующая схема на базе стандартных компонент из библиотеки Standard. На рисунке 8 представлена схема 16-разрядного знакового вычитателя.

Как известно, для представления отрицательного числа в дополнительном коде достаточно инвертировать его двоичное представление и прибавить к результату 1. На рисунке 8 показано, что для того чтобы из числа A вычесть число B, к числу A прибавляется дополнительный код числа B.

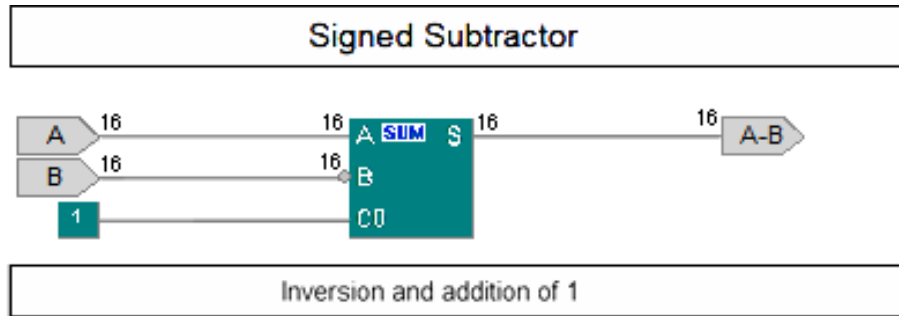


Рис. 8. Схема 16-разрядного знакового вычитателя

На рисунке 9 представлена схема 16-разрядного знакового умножителя. Вначале по знакам сомножителей вычисляется знак результата (если они разных знаков, то результат отрицательный, а если одинаковых — то положительный) и параллельно для каждого из сомножителей строится модуль соответствующего числа. Затем перемножаются полученные модули. И, наконец, на выход выдаётся полученное произведение (если результат положительный) и его дополнительный код, если результат должен быть отрицательным.

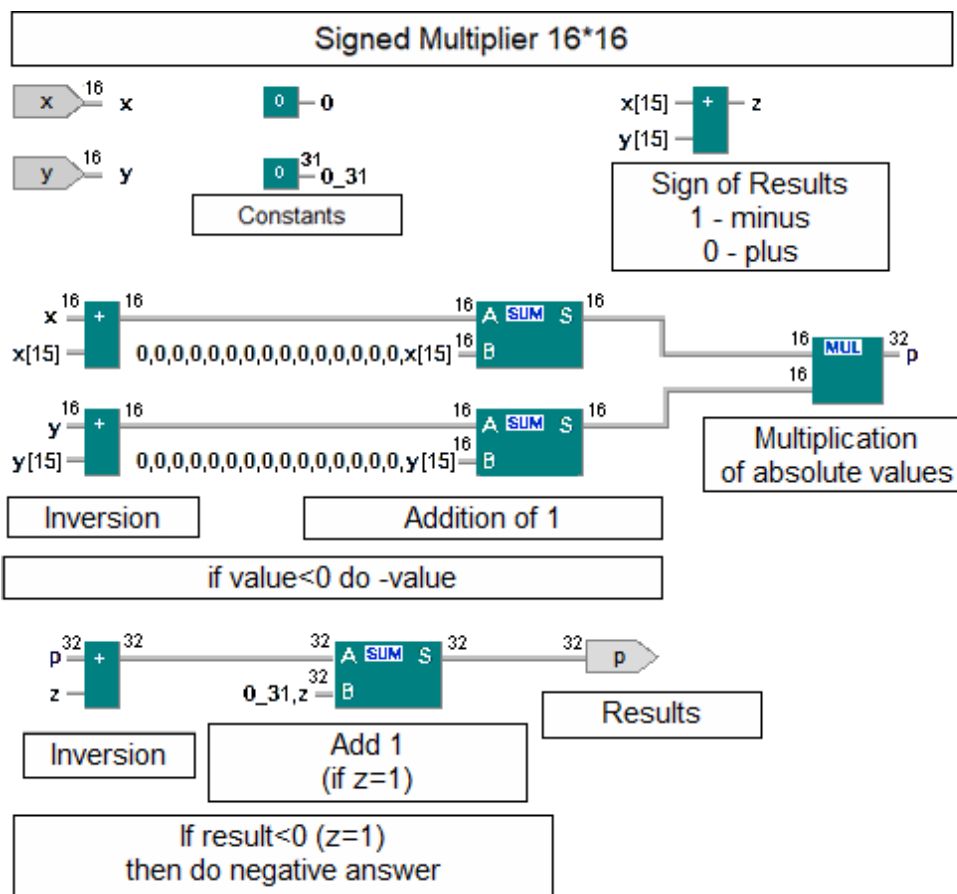


Рис. 9. Схема 16-разрядного знакового умножителя

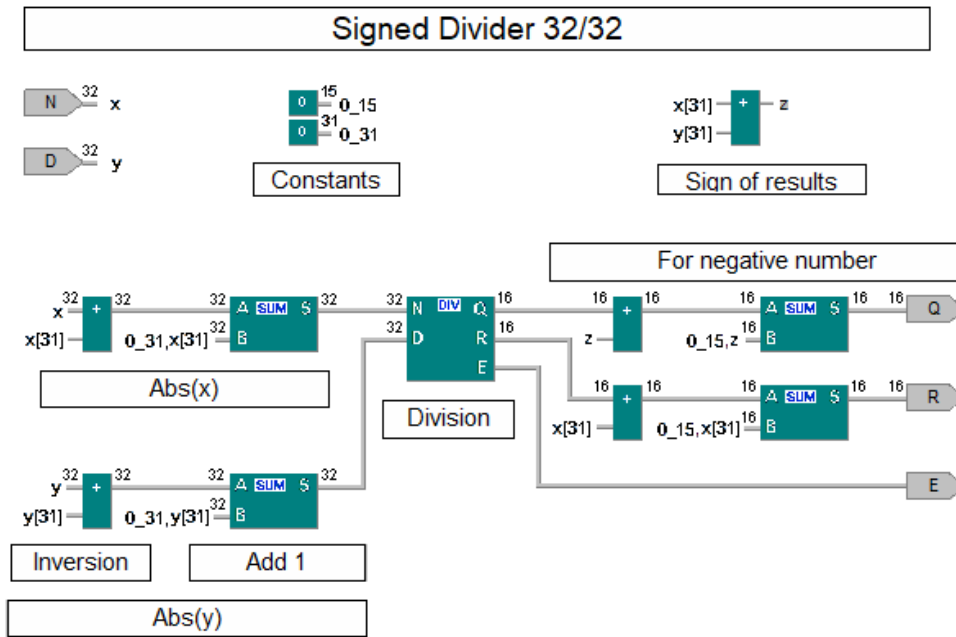


Рис. 10. Схема 32-разрядного знакового делителя

На рисунке 10 представлена схема 32-разрядного знакового делителя. Вначале по знакам делимого и делителя вычисляется результат (если они разных знаков, то результат отрицательный, а если одинаковых — то положительный) и параллельно для делимого и делителя строится модуль соответствующего числа. Затем выполняется деление модуля делимого на модуль делителя. И, наконец, на выход выдаётся полученный результат (если он положительный) и его дополнительный код, если результат должен быть отрицательным. При целочисленном делении строятся частное, остаток и признак ошибки (деление на 0), как и в стандартном компоненте DIV.

На рисунке 11 представлена разработанная студентами схема 32-разрядного компаратора, это правильно работающий проект, много лет используемый на практике. Несколь-

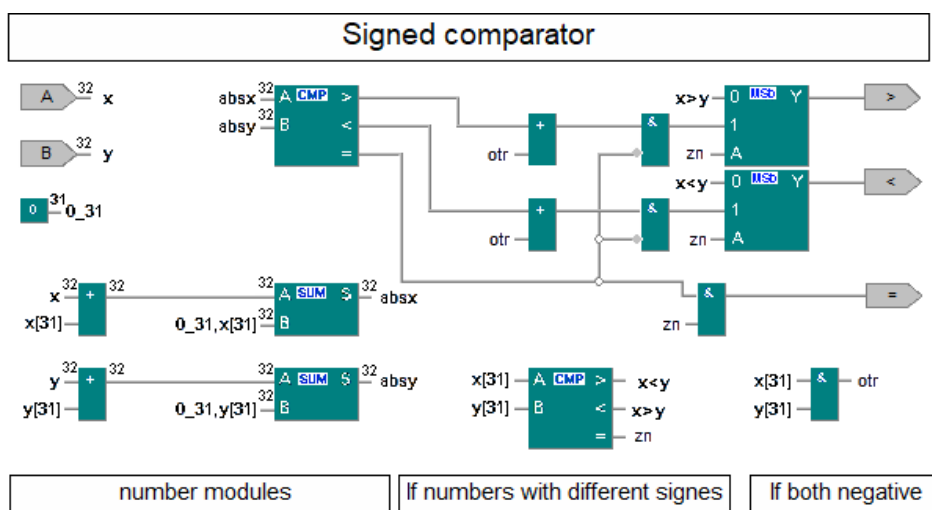


Рис. 11. Схема 32-разрядного знакового компаратора

ко лет спустя студентами же был предложен альтернативный вариант, существенно более простой (рис. 12).

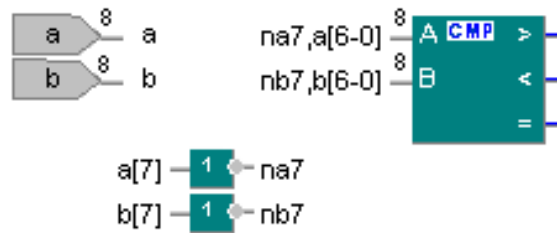


Рис. 12. Схема 8-разрядного знакового компаратора

Старшие разряды обоих сравниваемых чисел инвертируются, полученные числа сравниваются стандартным компаратором, сравнивающим положительные числа. Далее приводится объяснение этой схемы её автором (Климович Дмитрий, ПО-31, февраль 2009 года):

«Вопрос был в том, как сделать из беззнакового компаратора знаковый. Ведь можно просто прибавить к каждому из чисел 2^{n-1} (n — количество бит). Никто ведь не будет спорить, что от этого результат сравнения не изменится? Не изменится, но оба числа станут положительными и можно будет применить беззнаковый компаратор.

Так вот, когда мы инвертируем старший бит у наших чисел, то получаются не просто какие-то другие непонятные числа, а положительные числа, разность которых осталась прежней. Для 4-битов:

- положительные
 - 0xxx
 - +1000
 - 1xxx — увеличилось на 8 (2^3),
- отрицательные
 - 1xxx
 - +1000
 - 1|0xxx — увеличилось на 8,
 - ↑
 - «уходит», так как у нас 4 бита.

Мы получили 2 числа в беззнаковой форме, к которым можно применить беззнаковый компаратор.

Таким образом, инвертируя старший бит, мы прибавляем к обоим числам одно и то же число, делая их положительными. Результат сравнения от этого не меняется».

Поскольку схемы умножения, деления и сравнения имеют фиксированные разрядности операндов, нужно уметь изменять разрядность чисел, которыми мы оперируем. Для увеличения разрядности как раз и служат устройства CBW (из 8битов в 16 битов) и CWD (из 16 битов в 32 битов), представленные на рисунках 13 и 14.

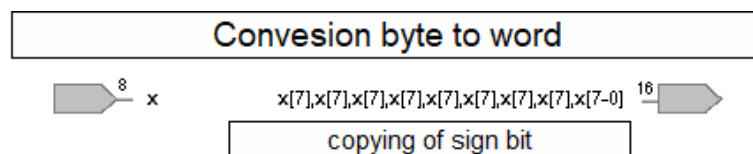


Рис. 13. Схема конвертирования байта в слово

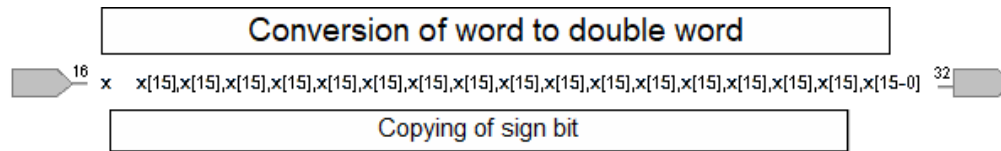


Рис. 14. Схема конвертирования слова в двойное слово

3.5. Рекомендуемая технология выполнения задания

Для операции сложения использовать сумматор SUM из проекта Standard (он правильно работает и с положительными, и с отрицательными числами). Для всех остальных арифметических операций (вычитание, умножение, деление, сравнение) берем устройства из проекта SignedArithm (соответственно iSUB, iMUL, iDIV, iCMP) (см. рис. 15).

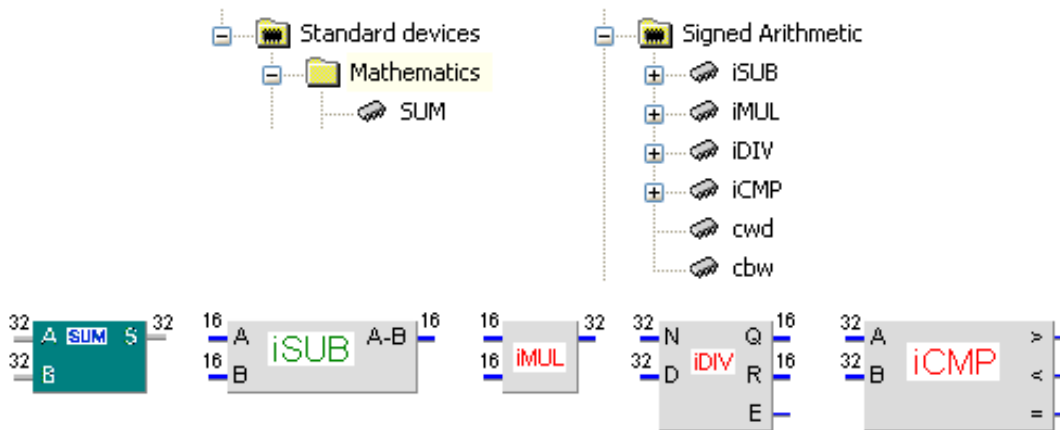


Рис. 15. Блоки используемых операций: арифметические и сравнения

НЕЛЬЗЯ менять разрядности устройств (iSUB, iMUL, iDIV, iCMP) из проекта SignedArithm (к сожалению, у нас там пока нет «защиты от дурака»).

Одна из проблем — разные разрядности исходных данных. Надо решать её выравниванием исходных данных до разрядности 16 битов с помощью устройства CBW (x8 => CBW => x16) и до 32 битов с помощью устройства CWD (x8 => CBW => x16 => CWD => x32) (рис. 16).

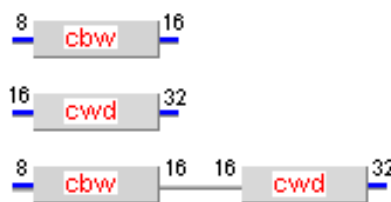


Рис. 16. Блоки повышения разрядности

Во всех задачах гарантируется, что ответ не превысит знакового целого в 16 битах.

Если нужно уменьшить разрядность (обычно после умножения), можно поступать так: обозначить выход (например Y, пусть он 32 битный), чтобы взять младшие 16 битов этого выхода, на 16-битном входе устройства достаточно написать Y[15-0] (предварительно кликнув по входной линии) (рис. 17).

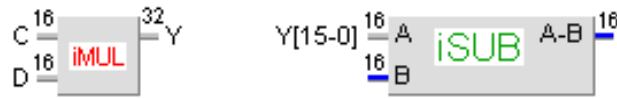


Рис. 17. Способ понижения разрядности

Для того чтобы выбрать один из нескольких результатов, используйте шинный мультиплексор (MSb8x2) из проекта Standard (см. рис. 18).

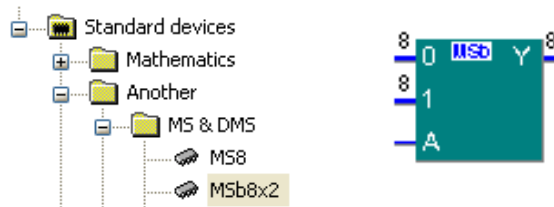


Рис. 18. Стандартный мультиплексор MSb8x2

Во всех задачах нужен 16-битный шинный мультиплексор, его можно получить из 8-битного с помощью внешнего редактора (кликнув правой кнопкой мыши по корпусу и выбрав строку «Внешний редактор») (рис. 19).

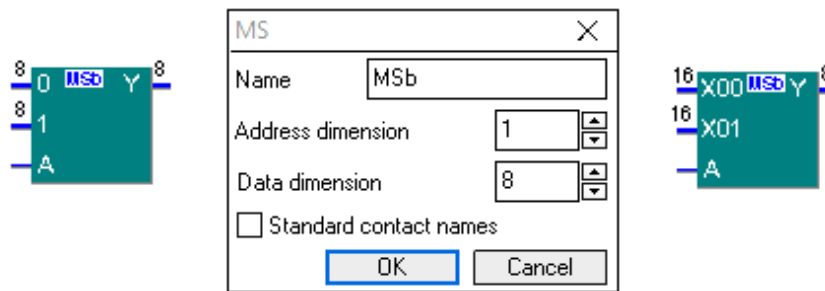


Рис. 19. 16-битный мультиплексор

На шину x01 нужно посылать результат
 res1, если условие выполнено (на линии A стоит 1),
 res0, если условие НЕ выполнено (на линии A стоит 0) (см. рис. 20).
 Условие Cond вычисляется с помощью схемы сравнения iCMP (см. рис. 21).

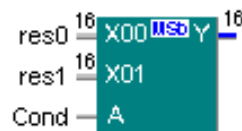


Рис. 20. Смысл входных контактов мультиплексора



Рис. 21. Вычисление условия

3.6. Отладка схемы

На рисунке 22 представлен фрагмент отладки схемы вычисления арифметического выражения. К схеме уже подключён тест, на котором запущено моделирование схемы. Поскольку обнаружена ошибка (значение на выходе схемы не равно эталонному значению, указанному в тесте), автоматически открывается окно отладчика, где представлена схема и на каждом входном и выходном контакте приводится имеющееся в текущий момент значение. Текущим моментом времени автоматически выбирается момент времени обнаружения первой ошибки.

Вычислив вручную выражение для текущих исходных данных, студент может точно определить, какое значение должно находиться на каждом из выходных контактов устройств схемы. Последовательно сверяя значения на выходах с результатами своих вычислений, легко можно найти первое расхождение — здесь и находится первая ошибка проектирования, которую необходимо исправить и вновь запустить моделирование. И так до тех пор, пока правильные результаты не будут получены для всех значений входных данных.

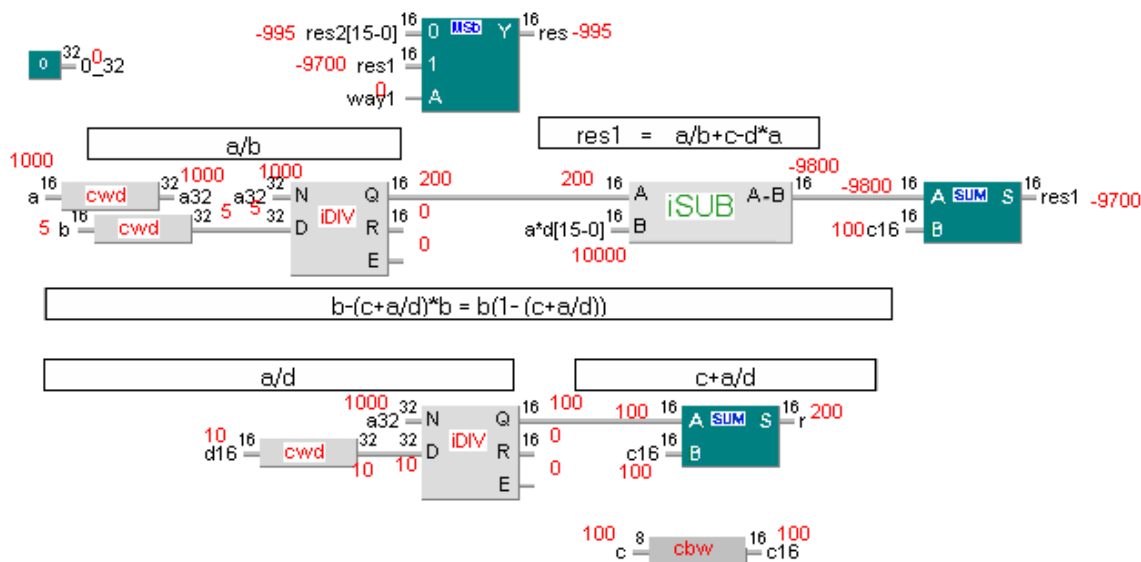


Рис. 22. Фрагмент отладки схемы вычисления арифметического выражения

После исправления всех ошибок требуется отправить полученный проект в систему DL для окончательной автоматической проверки.

4. ЗАДАНИЯ ПОВЫШЕННОЙ СЛОЖНОСТИ

Из 30 вариантов заданий на условные вычисления выражений есть несколько наиболее сложных, в условиях которых указано, что при вычислении арифметических выражений требуется определить модуль, знак или факториал числа (например приведенные ниже варианты 7, 21, 5).

Вариант 7

$$\text{RES} = \begin{cases} c/a+b/d-a*c, & \text{если } a=4 \\ (a+b-\text{sign}(c*d))/a, & \text{если } a \neq 4 \end{cases}$$

Вариант 21

$$\begin{aligned} & / x! + y / (a - b), \text{ при } x \geq 1 \text{ и } x \leq 7 \\ & \setminus 7! + x * y / (a - b), \text{ в остальных случаях.} \end{aligned}$$

Вариант 5

$$\begin{aligned} & / |(a-b)/c| + d * c, \text{ если } d > 0 \\ \text{RES} = & \setminus ||a| - |b|| * d / c + \text{sign}(a * b * c * d), \text{ если } d \leq 0 \end{aligned}$$

Для упрощения выполнения таких заданий имеются специальные обучающие задания, некоторые из которых приведены далее на рисунках 23–25. Фактически в каждом из них студенту предлагается схема, решающая поставленную задачу. При этом от студента требуется понимание того, как функционирует схема, для того чтобы завершить её выбором правильных надписей контактов.

В схеме вычисления модуля числа, представленной на рисунке 23, для числа вычисляется его дополнительный код и с помощью мультиплексора выбирается либо само число (если оно положительно), либо его дополнительный код (если исходное число было отрицательно).

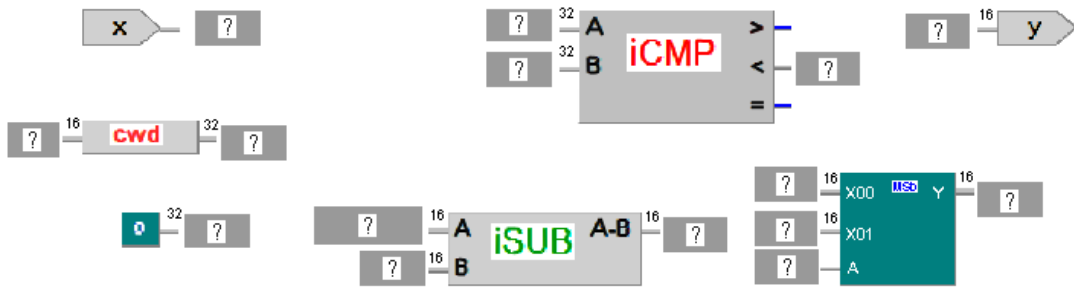


Рис. 23. Схема для вычисления модуля 16-разрядного числа

В схеме вычисления знака числа, представленной на рисунке 24, число отправляется на компаратор с нулём. В результате получаем три одиночные линии ($x > 0$, $x = 0$, $x < 0$), которые логически умножаются на константы 1, 0, и -1 (16 единиц) соответственно. Дизъюнкция этих трёх конъюнкций и определяет результат.

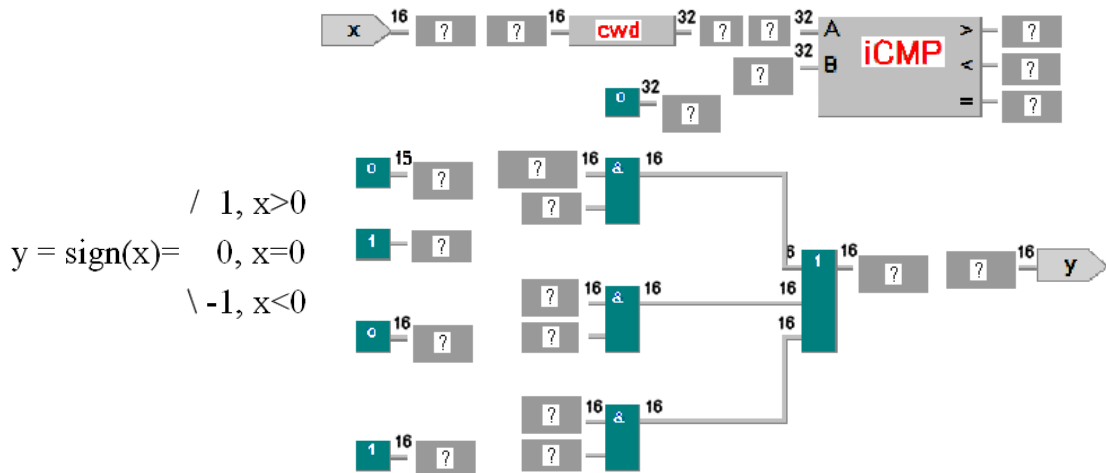


Рис. 24. Схема для вычисления знака 16-разрядного числа

В схеме вычисления факториала числа (в диапазоне от 0 до 15) используется мультиплексор с 16 шинами данных на входе, на каждую из которых подаётся соответствующая предварительно вычисленная константа от 0! до 15! (см. рис. 25).

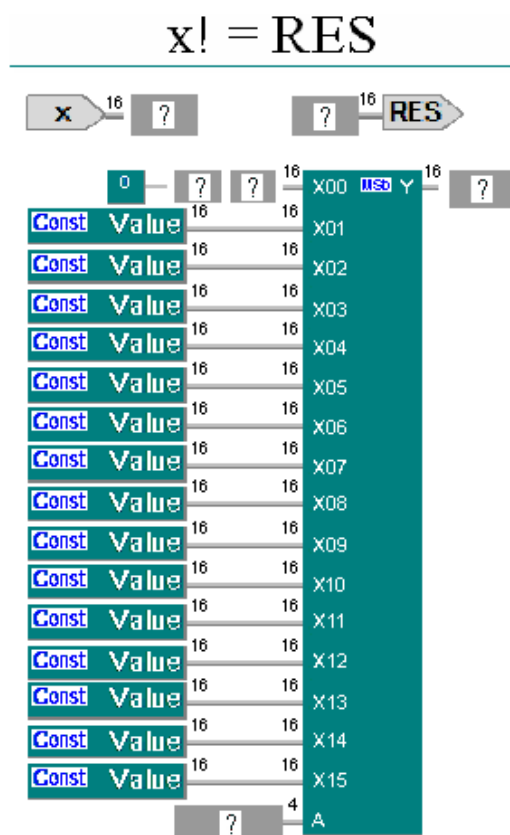


Рис. 25. Схема для вычисления факториала 16-разрядного числа (со значением от 0 до 15)

4.1. Функции системы дистанционного обучения DL.GSU.BY в учебном процессе

Система DL выполняет следующие базовые функции: предъявление задания студенту, приём и оперативная проверка (в течение минуты) схемы, спроектированной студентом, мгновенное сообщение студенту результата проверки схемы. В случае ошибки студенту предоставляется тест, на котором его схема выдала неправильный ответ. В тесте описываются входные воздействия и эталонные ответы. В реальном времени строится таблица результатов проверки работ всех студентов.

Система DL выполняет следующие дополнительные функции: предоставляет студенту полный доступ к теории по предмету, включая полный текст лекции по данной теме; поддерживает много лет функционирующий форум, в котором студенты задают свои вопросы, преподаватель отвечает и на первой странице форума пополняет тематически систематизированные ссылки на вопросы и ответы; поддерживает случайный выбор варианта контрольной работы для студента, запрет на доступ в Интернет и к сетевым папкам во время контрольных работ.

Для поддержки студентов, пропустивших занятия, или трудно усваивающих предмет, имеются видеозаписи решений студентами всех 30 вариантов контрольной работы, которые можно использовать во время самоподготовки.

5. ЗАКЛЮЧЕНИЕ

В данной статье описана разработанная автором и многократно апробированная методика изучения темы «Арифметические вычисления», ориентированная на работу в группах обучаемых с принципиально различными уровнями мотивации и предварительной подготовки. Серьезной технической основой методики является разработанная инструментальная система дистанционного обучения (DistanceLearningBelarus — <https://dl.gsu.by>). Внедрение данной методики обучения значительно повысило качество обучения, особенно наименее подготовленной и немотивированной категории студентов. В то же время, наиболее подготовленные студенты также удовлетворены таким подходом к обучению.

К альтернативным методам контроля и обучения можно отнести устные и письменные опросы, тесты, лабораторные стенды. Все они существенно проигрывают предложенной автором методике обучения по интенсивности и результативности учебного процесса. Программные аналоги связи систем DL.GSU.BY и HLCCAD автору неизвестны.

Список литературы

1. *Aznan N., Perdana R., Jumadi J., Nurcahyo H., Wiyatmo Y.* The Implementation of Blended Learning and Peer Tutor Strategies in Pandemic Era: A Systematic Review // Proc. of the 6th International Seminar on Science Education (ISSE 2020). 28–29 Nov. 2020. Yogyakarta, Indonesia. 2021. P. 906–914. doi:10.2991/assehr.k.210326.130
2. *Kakeshita T., Ohtsuki M.* Survey and Analysis of Computing Education at Japanese Universities: Non-IT Departments and Courses // Olympiads in Informatics. 2019. Vol. 13. P. 57–80.
3. *Jones K., Ravishankar S.* Higher Education 4.0. The Digital Transformation of Classroom Lectures to Blended Learning. Singapore: Springer, 2021.
4. *Zaugg H., Graham C., Lim C., Wang T.* Current and Future Directions of Blended Learning and Teaching in Asia // Chapter 16 in the book “Blended Learning for Inclusive and Quality Higher Education in Asia”. 2021. P. 301–327.
5. *Stahl G.* Redesigning Mathematical Curriculum for Blended Learning // Education. Sciences. 2021. Vol. 11. P. 165–177.
6. *Warman L. A. D.* The Effect of Google Classroom in Blended Learning on University Students’ English Ability // Journal of English for Academic. 2021. Vol 8, № 1. P. 12–23.
7. *Antwi-Boampong A.* Blended Learning Adoption in Higher Education: Presenting the Lived Experiences of Students in a Public University from a Developing Country // The Turkish Online Journal of Educational Technology. 2021. Vol. 20, № 2. P. 14–22.
8. *Oktaria S., Sasongko R., Kristiawan M.* Development of Blended Learning Designs using Moodle to Support Academics of The Curriculum in University of Bengkulu // JurnalStudi Guru dan Pembelajaran (Indonesia). 2021. Vol. 4, № 1. P. 118–126.
9. *Dolinsky M.* Teaching Algorithms and Programming First Year University Students on Base of Distance Learning System DL.GSU.BY // WSEAS Transactions on Advances in Engineering Education. 2022. Vol. 19. P. 52–57.
10. *Dolinsky M.* Experience of Blended Learning the Fundamentals of Digital Electronics for First/Second Year University students On Base of Distance Learning System DL.GSU.BY // International Journal of Education and Learning Systems. 2022. Vol. 7. P. 59–64.
11. *Dolinsky M.* Experience of Blended Learning in the Subject — Architecture of Computers // Journal of Information Technology and Digital World. 2022. Vol 4, № 3. P. 167–182.
12. *Dolinsky M. S.* Tool HLCCAD for Blended Learning the Fundamentals of Digital Electronics // International Journal of Circuits and Electronics. 2022. Vol. 7. P. 47–55.
13. *Dolinsky M.* Instrumental System of Distance Learning DL.GSU.BY and Examples of its Application // Global Journal of Computer Science and Technology Interdisciplinary. 2022. Vol. 22, № 1. P. 45–53.

14. Dolinsky M. One Approach to Study the Topic “Logic and Combinational Circuits” // WSEAS Transactions on Advances in Engineering Education, 2023. vol. 20 P. 14–24.
15. Dolinsky M. About One Approach to the Study of the Topic “Synthesis of Combinational Circuits Using Carnot Maps” // WSEAS Transactions on Advances in Engineering Education, 2023. Vol. 20. P. 60–69.

Поступила в редакцию 14.10.2023, окончательный вариант — 16.11.2023.

Долинский Михаил Семёнович, кандидат технических наук, доцент кафедры математических проблем управления и информатики, факультет математики и технологий программирования. Гомельский государственный университет им. Ф. Скорины, Гомель, Республика Беларусь, dolinsky@gsu.by

Computer tools in education, 2023

№ 4: 83–98

<http://cte.eltech.ru>

doi:10.32603/2071-2340-2023-4-83-98

On One Approach to Teaching the Theme “Arithmetic Circuits: Design, Simulation and Debugging”

Dolinsky M. S.¹, Cand. Sc., Associate Professor, dolinsky@gsu.by

¹ Francisk Skorina Gomel State University, 119 Kirova str., 246019, Gomel, Republic of Belarus

Abstract

This article describes the technology of teaching the theme “Arithmetic circuits: design, simulation and debugging” of basic digital electronics course to first/second-year students based on the DL.GSU.BY website. The main advantages of the technology include training adapted to the student, many years of experience in practical application and effectiveness. The following issues are consistently considered in the article: the theoretical foundations of the topic; library of standard components; a system of step-by-step learning to design arithmetic circuits, a technology for simulation and debugging.

Keywords: *basic digital electronics, arithmetic circuits, simulation, debugging, website DL.GSU.BY.*

Citation: Dolinsky M. S., “On One Approach to Teaching the Theme “Arithmetic Circuits: Design, Simulation and Debugging”,” *Computer tools in education*, no. 4, pp. 83–98, 2023 (in Russian); doi:10.32603/2071-2340-2023-4-83-98

References

1. N. Aznam, R. Perdana, J. Jumadi, H. Nurcahyo, and Y. Wiyatmo, “The Implementation of Blended Learning and Peer Tutor Strategies in Pandemic Era: A Systematic Review,” in *Proc. of the 6th International Seminar on Science Education (ISSE 2020), Yogyakarta, Indonesia, 28–29 Nov. 2020*, pp. 906–914, 2021; doi: 10.2991/assehr.k.210326.130
2. T. Kakeshita and M. Ohtsuki, “Survey and Analysis of Computing Education at Japanese Universities: Non-IT Departments and Courses,” *Olympiads in Informatics*, vol. 13, pp. 57–79, 2019; doi:10.15388/foi.2019.05

3. K. A. Jones and R. S. Sharma, *Higher Education 4.0. The Digital Transformation of Classroom Lectures to Blended Learning*, Springer Singapore, 2021; doi:10.1007/978-981-33-6683-1
4. H. Zaugg, C. R. Graham, C. P. Lim, and T. Wang, “Current and Future Directions of Blended Learning and Teaching in Asia,” *Blended Learning for Inclusive and Quality Higher Education in Asia*, pp. 301–327, 2021; doi:10.1007/978-981-33-4106-7_16
5. G. Stahl, “Redesigning Mathematical Curriculum for Blended Learning,” *Education Sciences*, vol. 11, no. 4, pp. 165–167, 2021; doi:10.3390/educsci11040165
6. L. A. D. Warman, “The Effect of Google Classroom in Blended Learning on University Students’ English Ability,” *Journal of English for Academic*, vol. 8, no. 1, pp. 12–23, 2021; doi:10.25299/jshmic.2021.vol8(1).6216
7. A. Antwi-Boampong, “Blended Learning Adoption in Higher Education: Presenting the Lived Experiences of Students in a Public University from a Developing Country,” *The Turkish Online Journal of Educational Technology*, vol. 20, no. 2, pp. 14–22, 2021.
8. S. D. Oktaria, R. N. Sasongko, and M. Kristiawan, “Development of Blended Learning Designs using Moodle to Support Academics of The Curriculum in University of Bengkulu,” *Jurnal Studi Guru dan Pembelajaran*, vol. 4, no. 1, pp. 118–126, 2021; doi:10.30605/jsgp.4.1.2021.548
9. M. S. Dolinsky, “Teaching Algorithms and Programming First Year University Students on Base of Distance Learning System DL.GSU.BY,” *WSEAS Transactions on Advances in Engineering Education*, vol. 19, pp. 52–57, 2022; doi:10.37394/232010.2022.19.6
10. M. Dolinsky, “Experience of Blended Learning the Fundamentals of Digital Electronics for First/Second Year University students On Base of Distance Learning System DL.GSU.BY,” *International Journal of Education and Learning Systems*, vol. 7, pp. 59–64, 2022.
11. M. Dolinsky, “Experience of Blended Learning in the Subject — Architecture of Computers,” *Journal of Information Technology and Digital World*, vol. 4, no. 3, pp. 167–182, 2022, doi:10.36548/jitdw.2022.3.004
12. M. S. Dolinsky, “Tool HLCAD for Blended Learning the Fundamentals of Digital Electronics,” *International Journal of Circuits and Electronics*, vol. 7, pp. 47–55, 2022.
13. M. Dolinsky, “Instrumental System of Distance Learning DL.GSU.BY and Examples of its Application,” *Global Journal of Computer Science and Technology Interdisciplinary*, vol. 22, no. 1, pp. 45–53, 2022.
14. M. Dolinsky, “One approach to study the topic «Logic and combinational circuits»,” *WSEAS Transactions on Advances in Engineering Education*, vol. 20, pp. 14–24, 2023.
15. M. Dolinsky, “About One Approach to the Study of the Topic «Synthesis of Combinational Circuits Using Carnot Maps»,” *WSEAS Transactions on Advances in Engineering Education*, vol. 20, pp. 60–69, 2023.

Received 14-10-2023, the final version — 16-11-2023.

Michael Dolinsky, Candidate of Sciences (Tech.), Associate Professor of the Department Mathematical Problems of Management and Computer Science, Faculty of Mathematics and Technologies of Programming, Francisk Skorina Gomel State University, Gomel, Republic of Belarus, dolinsky@gsu.by