Министерство образования и науки Российской Федерации Федеральное агентство по образованию

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по комплексу лабораторных работ дисциплины

«Проектирование цифровых устройств»

Выполнил студент группы ИВТ-31 /Крючков И. С/ Проверил /Мельцов В. Ю./

Киров 2022

1. Цель работы

Изучение основ проектирования арифметико-логических устройств ЭВМ. Разработка и тестирование АЛУ для выполнения операции деления, вычитания модулей и логической операции XOR-НЕ.

1. Задание
2. Разработка функциональной схемы операционного автомата деления;
3. Разработка функциональной схема ОА для определения результата операции |A| - |B|;
4. Разработка функциональной схемы ОА для выполнения логической операции XOR-НЕ;
5. Построение объединенной функциональной схемы ОА для выполнения данных операций.
6. Ход работы
   1. Операция деления

Выполнить операцию деления чисел в двоичной системе счисления с порядками с плавающей запятой в прямом коде вторым способом без восстановления остатков и вычитанием в обратном коде.

* + 1. Словесное описание алгоритма:

1. Принять операнды
2. Проверить делитель на равенство нулю: делитель равен нулю – прекратить операцию деления, установить признак ДНН, иначе перейти к п.3
3. Проверить делимое на равенство нулю: делимое равно нулю – прекратить операцию деления, выдать ноль, иначе перейти к п.4
4. Определить порядок частного вычитанием порядка делителя из порядка делимого, при этом возможны следующие исключительные ситуации.:
   1. ПРС: в результате вычитания порядков в знаковом разряде единица, но отсутствует единица переноса. При возникновении ПРС прекратить операцию деления, установить признак ПРС
   2. ПМР: в результате вычитания порядков в знаковом разряде ноль, но присутствует единица переноса. При возникновении ПМР прекратить операцию деления, выдать ноль
   3. Вр. ПМР: в результате вычитания порядков в знаковом разряде единица, присутствует единица переноса, остальные разряды заполнены нулями. При возникновении вр. ПМР продолжить алгоритм.
5. Определить знак частного, сложением по модулю два знаковых разрядов делимого и делителя. При дальнейших операциях использовать модули мантисс
6. Анализируем знак остатка:
   1. Если остаток отрицателен, прибавить к остатку делитель
   2. Если остаток положителен, вычесть делитель из остатка
7. Проанализировать знак полученного остатка:
   1. Если остаток положителен, в младший разряд частного занести «1»
   2. Если остаток отрицателен, в младший разряд частного занести «0»
8. Выполнить сдвиги: частного на один разряд влево, делителя на один разряд вправо
9. Если заполнены все разряды частного перейти к п.10, иначе перейти к п.6.
10. Проанализировать результат:
    1. Если в знаковом разряде единица – сдвинуть мантиссу на 1 разряд вправо и увеличить порядок на 1:
       1. Если был зафиксирован признак временного ПМР – он устранится. Перейти к п.11
       2. Если в знаковом разряде единица, но отсутствует единица переноса – установить признак ПРС, прекратить операцию
    2. Если получена нормализованная мантисса частного:
       1. Если был обнаружен признак временного ПМР – прекратить операцию и выдать ноль, так как ПМР истинный, иначе перейти к п. 11
11. Присвоить знак, определенный в п.5
12. Выдать результат
13. Завершить операцию деления
    * 1. Управляющие и осведомительные сигналы:

y0 – сброс триггеров T3, T4, T5, T6, T7, T8, T9, T10, запись в T1, сброс RG4

y1 – запись в регистры RG1, RG2, в триггер T2

y2 – обнуление счетчика CT1, обнуление регистра RG3

y3 – запись в регистр RG4

y4 – запись в счетчик CT1 и триггер T3

y5 – сигнал о необходимости вычитания содержимого регистра RG2

y6 – уменьшение счетчика CT1 на единицу

y7 – сдвиг вправо регистра RG1, сдвиг влево регистра RG3, переключение триггера T6

y8 - увеличение счетчика CT1 на единицу

y9 - сброс триггеров T1, T2

y10 – выдача результата на шину

y11 – установка триггера T4 в единицу

y12 – установка триггера T5 в единицу

y13 – установка триггера T7 в единицу

y14 – установка триггера T8 в единицу

y15 – запись в T9, T10

X – проверка наличия операндов на входной шине

P0 – проверка делителя на 0

P1 – проверка делимого на 0

P2 – проверка на ПРС

P3 – проверка на истинное ПМР

P4 – проверка на временное ПМР

P5 – знак разности порядков

P6 – проверка на окончание цикла деления

P7 – проверка на корректировку частного

Z – проверка возможности выдачи результата на выходную шину

* + 1. Функциональная схема алгоритма

Функциональная схема представлена на рисунке 1.



Рисунок 1 – Функциональная схема ОА операции деления

* + 1. Содержательная граф-схема алгоритма

ГСА представлена на рисунке 2.



Рисунок 2 – ГСА алгоритма деления

* + 1. Экранные формы
       1. Штатная ситуация

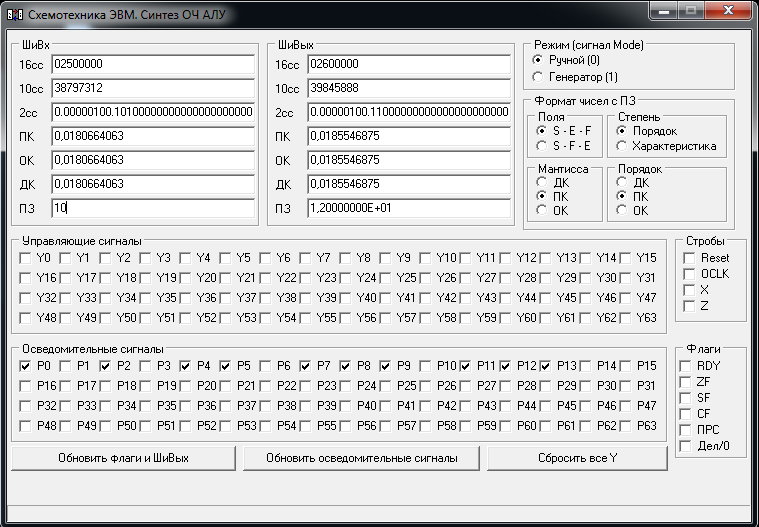


Рисунок 3 – Первой операнд штатной операции деления

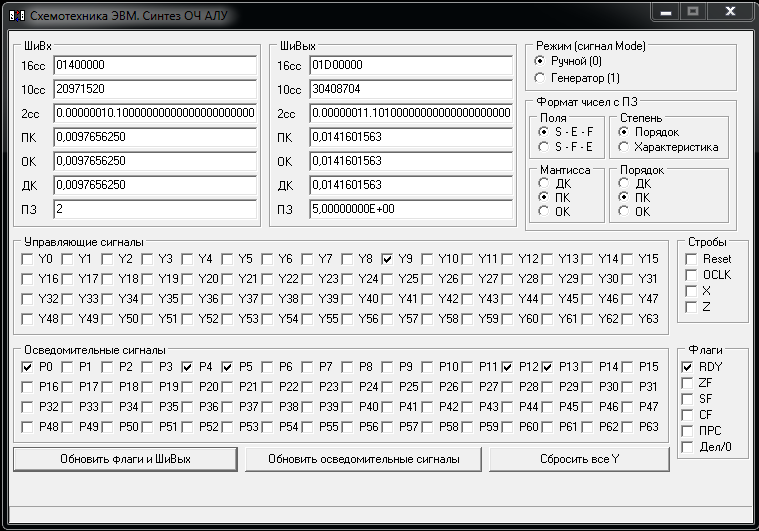


Рисунок 4 – Второй операнд и результат штатной операции деления

* + - 1. ПРС

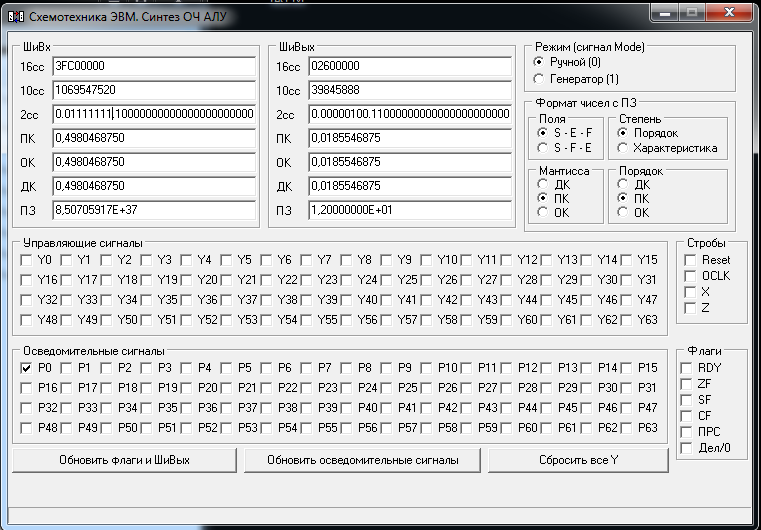


Рисунок 5 – Первый операнд ситуации ПРС

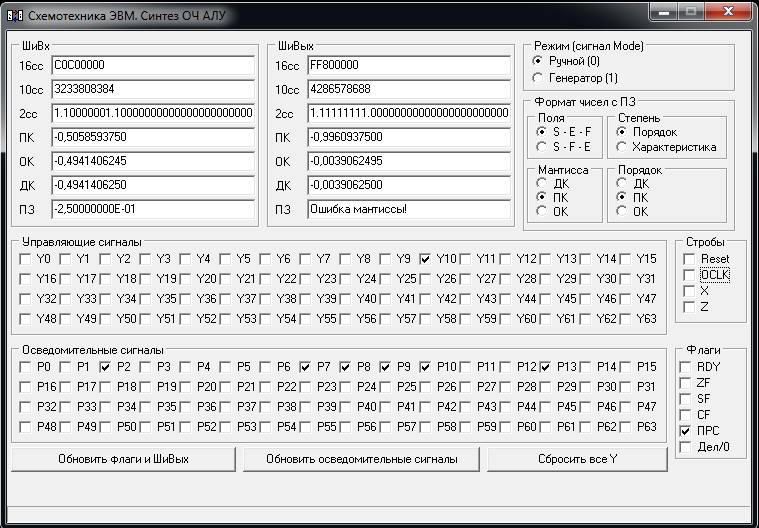


Рисунок 6 – Второй операнд и результат операции ПРС

* 1. Операция вычитание модулей
     1. Словесное описание алгоритма:

1. Принять первый операнд
2. Принять второй операнд
3. Проверить операнды на равенство нулю:

* если первый операнд равен нулю, то выдать второй операнд в качестве результата
* если второй операнд равен нулю, то выдать первый операнд в качестве результата
* иначе, перейти к п.4

1. Выполнить сравнение порядок, путем их вычитания
2. Сравнить разность порядков с числами «-23» и «23»:

* если разность порядков >= 23, то сформировать результат = |A|, перейти к п. 9
* если разность порядков <= -23, то сформировать результат = -|B|, перейти к п.9.
* иначе, перейти к п.6

1. Проанализировать разность порядков:
   1. Если разность порядков не равна нулю, то сдвинуть мантиссу меньшего из чисел вправо, увеличить его порядок на 1, перейти к п.6;
   2. Если разность порядков равна нулю, перейти к п. 7.
2. Вычесть модули мантисс чисел в ОК
3. Проанализировать результат:
   1. Если необходимо, нормализовать результат путем сдвига мантиссы результата влево и уменьшением порядка результата на 1, при этом:
      * Если в знаковом разряде порядка результата ноль и присутствует единица переноса – признак ПМР, сформировать результат равный нулю, перейти к п.9
      * Иначе перейти к п. 8)
   2. Если результат нормализован, перейти к п. 9.
4. Сформировать результат операции
5. Выдать результат операции
6. Завершить операцию вычитание модулей.
   * 1. Управляющие и осведомительные сигналы

y0 – сброс RG2, CT2, T1, T2, T3, T4, T5, T6, T7

y1 – запись в регистр RG1, счетчик CT1

y2 – запись в CT2 и триггер T1

y3 – запись в регистр RG2

y4 – сигнал о необходимости вычитания порядков

y5 – увеличение счетчика CT1 на единицу

y6 – увеличение счетчика CT2 на единицу

y8 – уменьшение счетчика CT2 на единицу

y9 – сдвиг вправо регистра RG1

y10 – сдвиг вправо регистра RG2

y11 – сигнал о необходимости вычитания мантисс

y12 – сдвиг влево регистра RG2

y13 – выдача результата на шину

y14 – сброс CT1

y15 – уменьшение CT1 на единицу

y16 – установка триггера T2 в единицу

y17 – установка триггера T3 в единицу

y18 – установка триггера T4 в единицу

y19 – установка триггера T5 в единицу

y20 – запись в T6, T7

X – проверка наличия операндов на входной шине

P0 – результат сравнения порядка

P1 – проверка разности порядков на ноль

P2 – знак CT2

P3 – проверка на необходимость нормализации, проверка первого операнда на ноль

P4 – проверка на ПМР

P5 – проверка второго операнда на ноль

P6 – проверка CT1 на 0

P7 – знаковый разряд КС2

P8 – знак CT1

P9 – проверка CT1 на 0

Z – проверка возможности выдачи результата на выходную шину

* + 1. Функциональная схема

Функциональная схема представлена на рисунке 7.



Рисунок 7 – Функциональная схема ОА операции вычитание модулей

* + 1. Содержательная граф-схема алгоритма

ГСА представлена на рисунках 8-9.



Рисунок 8 – ГСА алгоритма вычитание модулей



Рисунок 9 – ГСА алгоритма вычитание модулей

* + 1. Экранные формы
       1. Штатная ситуация

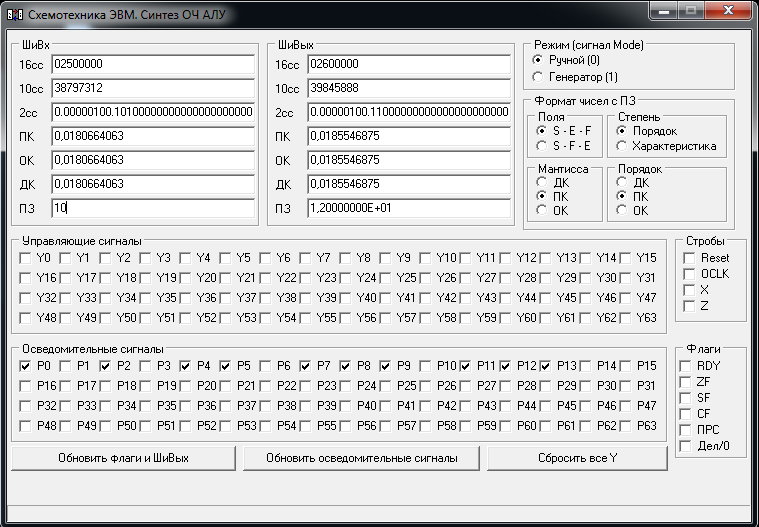


Рисунок 10 – Первый операнд штатной ситуации

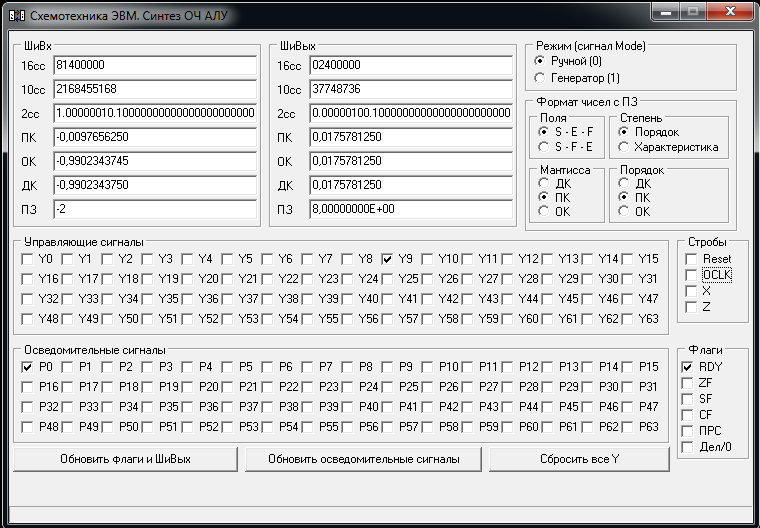


Рисунок 11 – Второй операнд и результат штатной операции

* 1. Логическая операция XOR-НЕ
     1. Словесное описание алгоритма

1. Принять операнды
2. Выполнить операцию XOR над знаковыми разрядами и мантиссами операндов
3. Проинвертировать результат
4. Присвоить результату порядок первого операнда
5. Выдать результат
6. Завершить операцию XOR-НЕ.
   * 1. Управляющие и осведомительные сигналы

y0 – сброс триггеров T1, T2, T3, T4, T5, T6, T7, T8 запись в регистр RG1

y1 – запись в регистр RG2

y2 – запись в регистр RG3

y3 – запись в триггеры T2, T4

y4 – выдача результата на шину

y5 – запись в T1

y6 – запись в T3

y7 – запись в T5

y8 – запись в T6

X – проверка наличия операндов на входной шине

Z – проверка возможности выдачи результата на выходную шину

* + 1. Функциональная схема

Функциональная схема представлена на рисунке 12.



Рисунок 12 – функциональная схема ОА операции XOR-НЕ

* + 1. Содержательная граф-схема алгоритма операции XOR-НЕ

ГСА представлена на рисунке 13.



Рисунок 13 – ГСА алгоритма XOR-НЕ

* + 1. Экранные формы

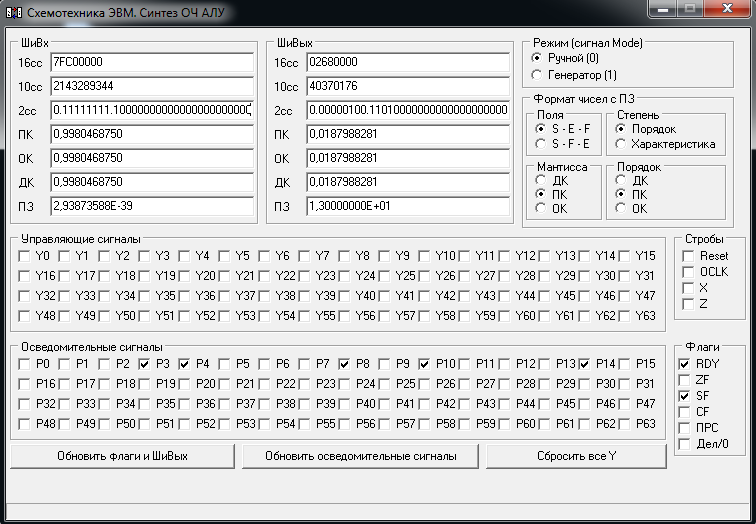


Рисунок 14 – Первый операнд логической операции

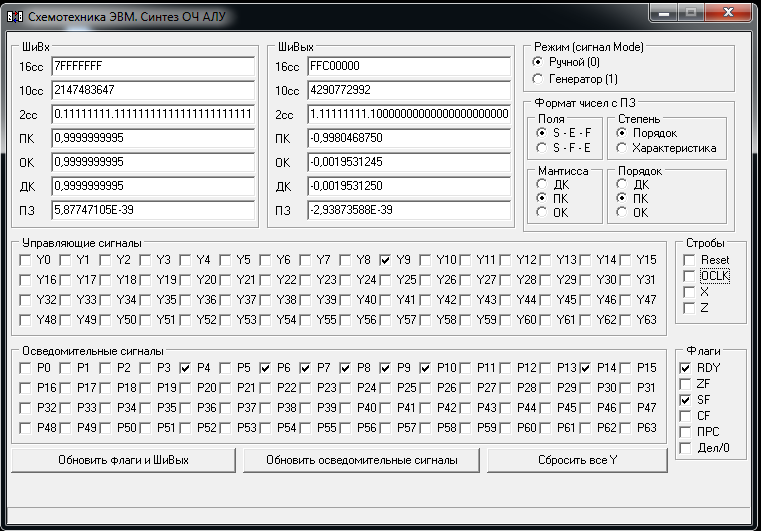


Рисунок 15 – Второй операнд и результат логической операции

* 1. Объединенная функциональная схема
     1. Управляющие и осведомительные сигналы

y0 – сброс триггеров T3, T4, T5, T6, T7, T8, T9, T10, T11, запись в триггер T1

y1 – запись в RG1, CT1, в триггер T2

y2 – сброс RG3, CT2

y3 – запись в регистр RG2

y4 – запись в счетчик CT2 и триггер T3

y5 – сигнал о необходимости вычитания порядков

y6 – сдвиг влево регистра RG3, переключение триггера T4

y7 - увеличение счетчика CT2 на единицу

y8 – уменьшение счетчика CT2 на единицу

y9 – выдача результата на шину

y10 – установка триггера T5 в единицу

y11 – установка триггера T6 в единицу

y12 - сброс триггеров T1, T2

y13 – сдвиг вправо регистра RG2

y14 – увеличение счетчика CT1 на единицу

y15 – сдвиг вправо регистра RG1

y16 – сброс регистра RG2

y17 – управление КС1

y18 – сдвиг влево регистра RG2

y19 – запись в RG3, T8

y20 – запись в триггер T7, T11

y21 – запись в триггер T9

y22 – запись в триггер T10

y23 – сброс CT1

y24 – уменьшение счетчика CT1 на единицу

X – проверка наличия операндов на входной шине

P0 – проверка второго операнда на 0

P1 – проверка первого операнда на 0, проверка на необходимость нормализации

P2 – проверка на ПРС

P3 – проверка на истинное ПМР

P4 – проверка на окончание цикла деления

P5 – проверка на корректировку частного

P6 – проверка на временное ПМР

P7 – знак CT2

P8 – результат сравнения порядков

P9 – проверка разности порядков на 0

P10 – знак CT1

P11 – проверка CT1 на 0

P12 – знаковый разряд КС5

cop0 - младший бит кода операции

cop1 - старший бит кода операции

* + 1. Функциональная схема

Функциональная схема представлена на рисунке 16.



Рисунок 16 – функциональная схема объединенного ОА.

* + 1. Содержательная граф-схема алгоритма

ГСА представлена на рисунках 17-20.



Рисунок 17 – ГСА объединенного алгоритма



Рисунок 18 – ГСА объединенного алгоритма



Рисунок 19 – ГСА объединенного алгоритма



Рисунок 20 – ГСА объединенного алгоритма

* + 1. Схема в Quartus

Схема в Quartus представлена на рисунке 21.

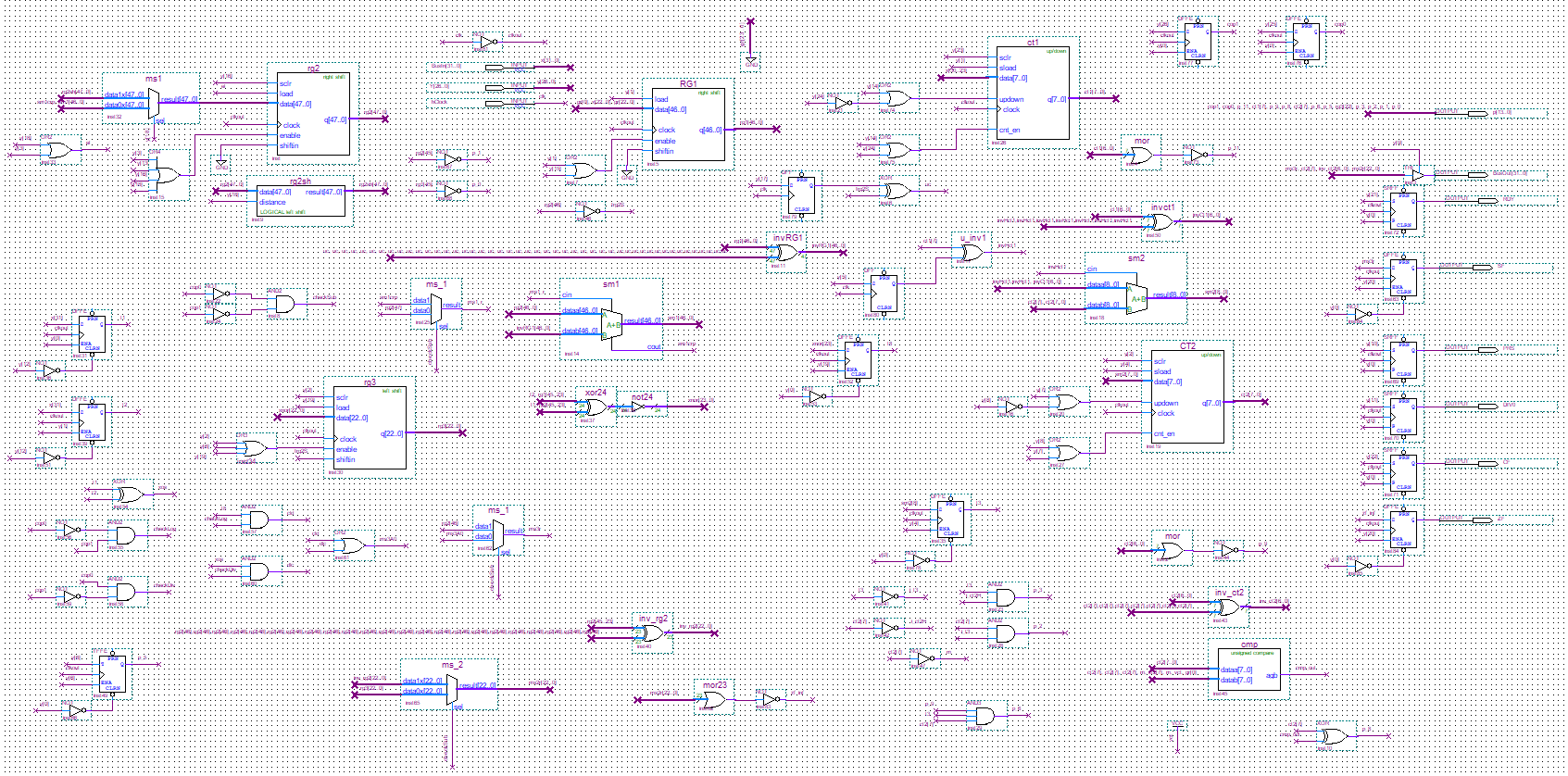


Рисунок 21 – Схема в Quartus

* + 1. Экранные формы

Экранные формы представлены на рисунках

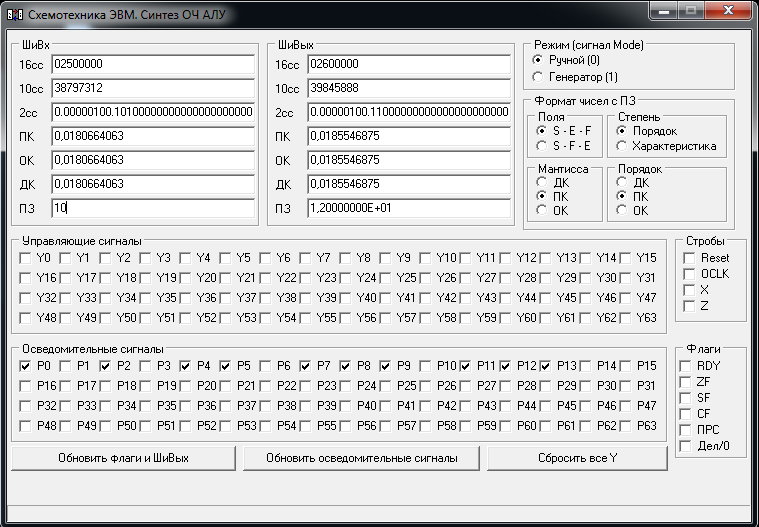


Рисунок 22 – Первый операнд операции деления

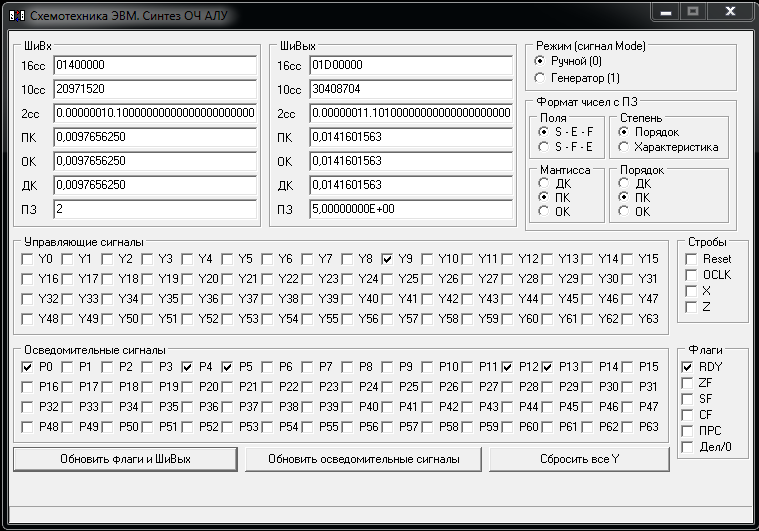


Рисунок 23 – второй операнд и результат операции деления

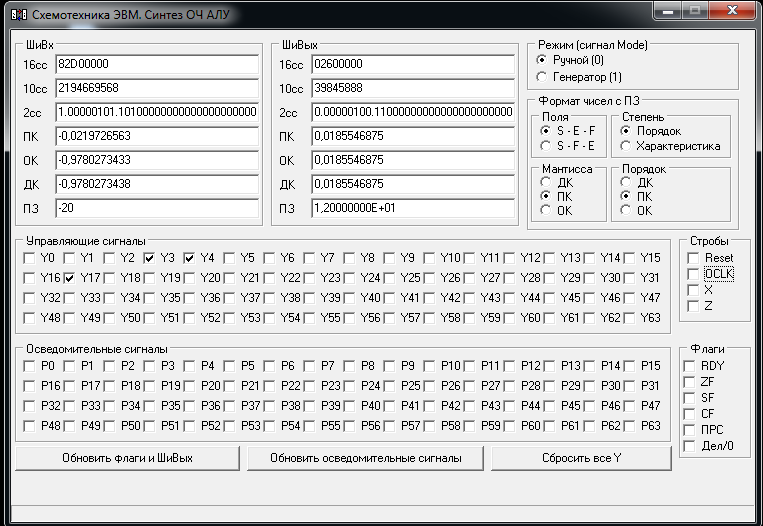


Рисунок 24 – первый операнд операции вычитание модулей

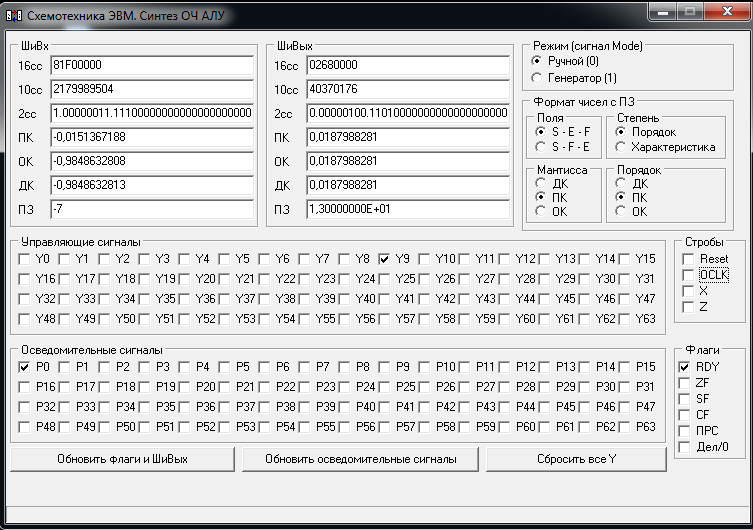


Рисунок 25 – Второй операнд и результат операции вычитание модулей

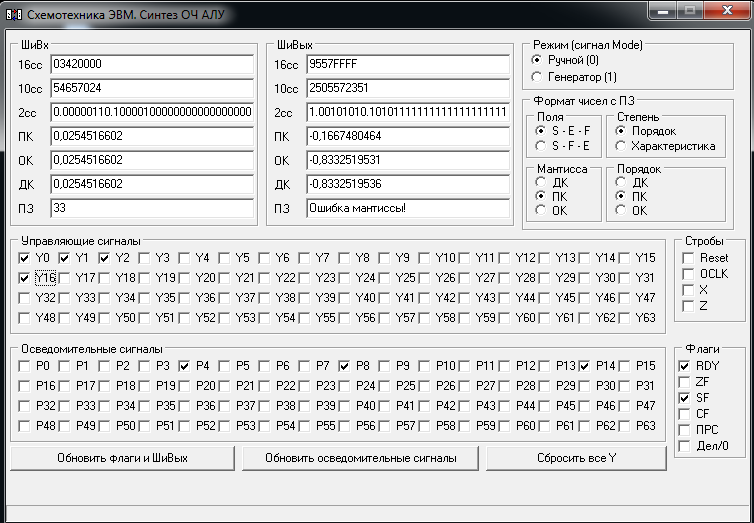


Рисунок 26 – Первый операнд логической операции

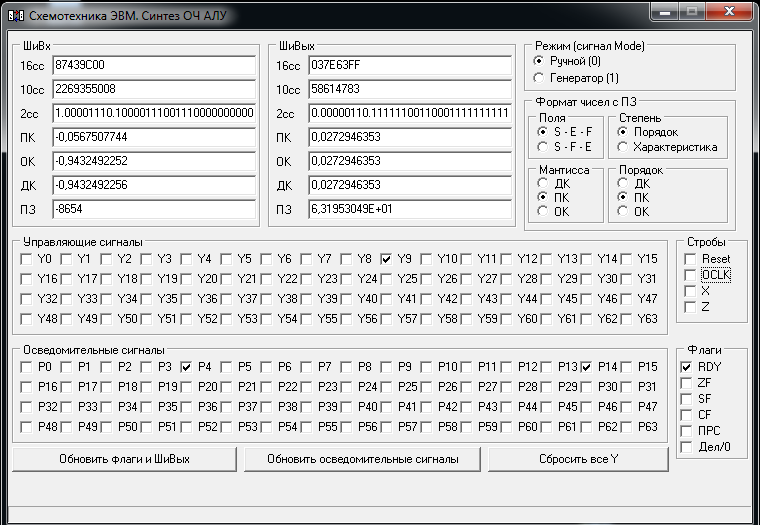


Рисунок 27 – Второй операнд и результат логической операции

Вывод: в ходе выполнения лабораторных работ было разработано арифметико-логическое устройство для выполнения операции деления, вычитания модулей и логической операции XOR-НЕ в системе автоматизированного проектирования (САПР) Quartus. Работоспособность АЛУ была протестирована на программируемой логической интегральной схеме (ПЛИС).