

УДК 004.315.2

ДВОИЧНЫЙ СУММАТОР НА ПЛАТФОРМЕ ARDUINO*Бондарев К.Д. и Зиновенко В.И., студенты гр.250702**Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь**Сидорович А.С. – старший преподаватель кафедры ЭВМ*

Аннотация. В данной работе реализуется сумматор двоичных чисел в дополнительных кодах на платформе Arduino путем проектирования схемы для обработки входных и выходных данных и написания программы для их преобразования в конечное представление.

Ключевые слова. Параллельный двоичный сумматор, двоичная арифметика, логические «ИЛИ», «И», «XOR», обратный и дополнительный код, аппаратная платформа Arduino, симуляция, эмулятор, соединения, резистор, порты, жидкокристаллический дисплей, потенциометр, обработка нажатий кнопок, ввод и вывод.

В этом проекте используется двоичный параллельный сумматор дополнительных кодов, который обычно отображается в двоичном формате с использованием светодиодов, но вместо этого используются Arduino, чтобы принимать и отображать данные на жидкокристаллическом дисплее.

Двоичный сумматор дополнительных кодов – это сумматор, оперирующий представлениями двоичных чисел в дополнительном коде.

В сумматорах дополнительных кодов может производиться как сложение чисел с одинаковыми знаками, так и сложение чисел с разными знаками (вычитание). Сумма дополнительных кодов двоичных чисел представляет собой дополнительный код результата.

Существует несколько причин, почему выбор обратного или модифицированного кода может быть предпочтительнее, чем выбор обратного или модифицированного кода:

1. Простота операций. Операции сложения и вычитания в дополнительном коде требуют всего лишь простого суммирования или вычитания двух чисел в двоичном коде. В обратном коде для выполнения этих операций требуется дополнительный шаг инвертирования битов, что усложняет аппаратную реализацию. В случае модифицированных кодов требуется выполнить дополнительные операции, чтобы определить знак числа, что также усложняет процесс.

2. Уникальность представления. Каждое отрицательное число имеет единственное представление в дополнительном коде, что делает его более предсказуемым и удобным для работы в программном коде.

3. Меньший объем памяти. Представление чисел в дополнительном коде позволяет использовать меньшее количество битов для хранения чисел, по сравнению с обратным кодом и знаковым модулем. Например, для представления числа “-8” в дополнительном коде требуется всего 4 бита, тогда как в обратном коде или знаковом модуле требуется 5 битов.

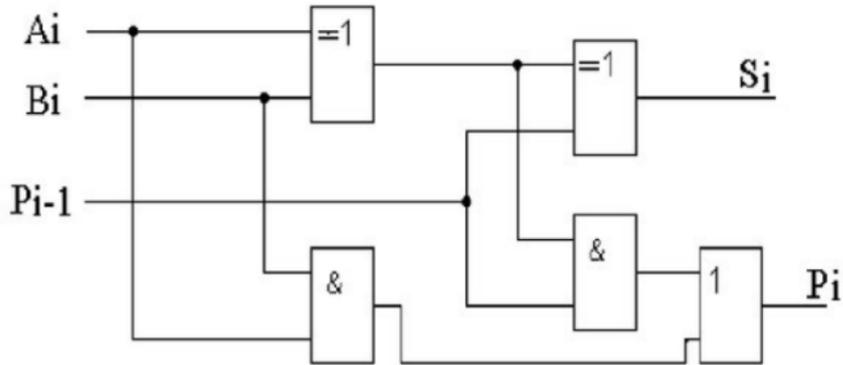
Таблица истинности полного одноразрядного сумматора двоичных кодов, где A_i , B_i – i -е разряды чисел A и B, P_{i-1} – перенос, формируемый в прошлом одноразрядном сумматоре, S_i – i -й разряд результата суммы, из которых формируется число результата, P_i – перенос в следующий одноразрядный сумматор, представлена в таблице 1.

Таблица 1 – Таблица истинности выходов полного одноразрядного сумматора двоичных кодов.

A_i	B_i	P_{i-1}	S_i	P_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1

1	1	1	1	1
---	---	---	---	---

Путем минимизации функции сложения была найдена минимальная дизъюнктивная функция и на ее основе спроектирован одноразрядный сумматор двоичных кодов. Правила сложения в одноразрядном сумматоре двоичных кодов двух чисел A_i и B_i отражены в схеме одноразрядного полного сумматора рисунке 1, выполненного на логических элементах.



$$S_i = A_i \oplus B_i \oplus P_{i-1};$$

$$P_i = A_i B_i + P_{i-1} A_i \oplus B_i.$$

Рисунок 1 – Структурная схема одноразрядного полного сумматора двоичных кодов

При параллельном способе сложения необходимо иметь отдельные одноразрядные сумматоры для каждого разряда чисел. Параллельный сумматор может быть составлен из одноразрядных сумматоров путем соединения выхода, на котором присутствует сигнал переноса данного разряда, со входом для сигнала переноса соседнего, более старшего разряда. За основу была взята данная схема параллельного сумматора двоичных кодов со сквозным переносом, представленная на рисунке 2.

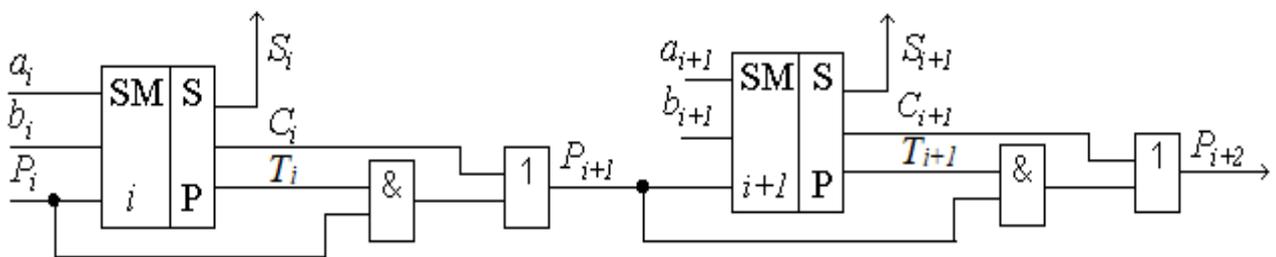


Рисунок 2 - Структурная схема параллельного сумматора со сквозным переносом

Для обеспечения функции вычитания, используется дополнительный код. Правило получения дополнительного кода отрицательных чисел можно сформулировать следующим образом: цифровая часть исходного числа инвертируется и совместно со знаком превращается в обратный код числа, к которому прибавляется код двоичного числа, соответствующий единице, в младшем разряде. Полученная сумма является числом в дополнительном коде к коду модуля этого числа. Дополнительный код положительного числа является его прямым кодом.

В работе используются 7-ми разрядные двоичные числа, самый старший разряд которых является знаковым. «1» или «0» являются «-» или «+» соответственно. Остальные разряды содержат значащую часть числа.

Для реализации аппаратной части Arduino используется:

- Arduino Uno R3
- 2 малые макетные платы
- Резисторы
- LCD 16 x 2 (жидко-кристаллический дисплей)
- 2 кнопки
- Потенциометр

Макет изображен на рисунке 3.

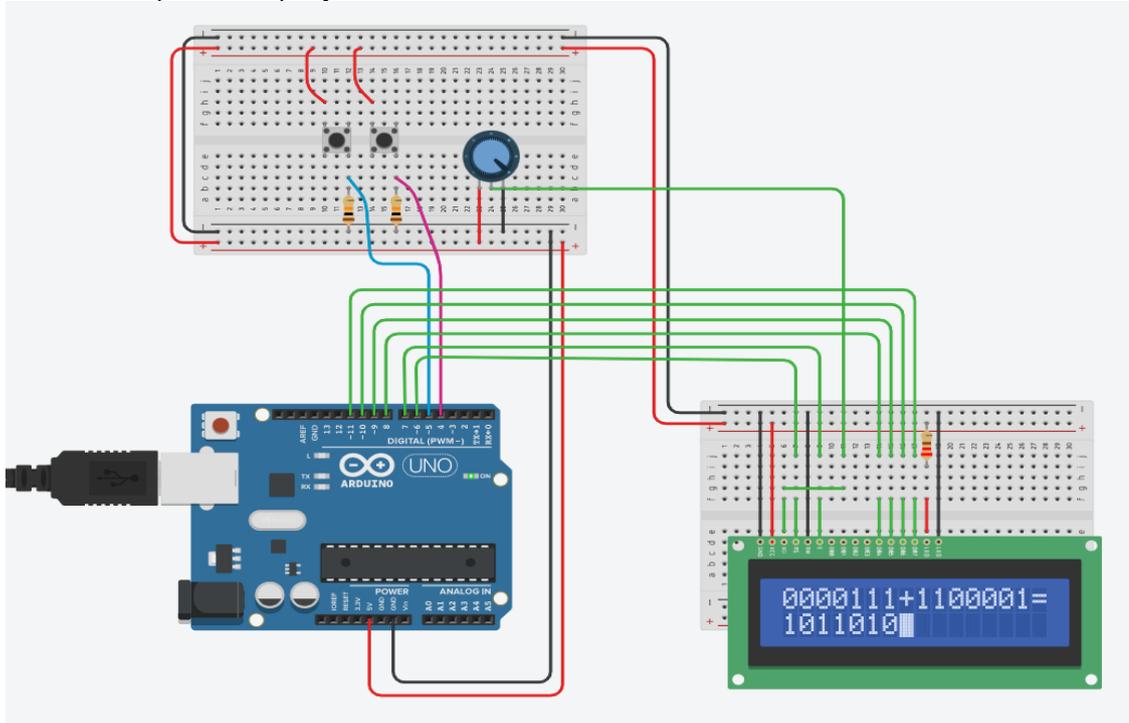


Рисунок 3 - Вид схемы двоичного сумматора на Arduino

Для разработки прототипов электронных устройств обычно используется безопасная макетная плата[1]. На верхней плате присутствуют две кнопки, отвечающие за ввод данных. К кнопкам ведут голубой и розовый провода, отвечающие за ввод «0» и «1» соответственно. Контакты 5V и GND подключены к макетным платам на плюс и минус соответственно, чтобы управлять электричеством на них. Также подключен ЖК-дисплей, яркость которого регулируется потенциометром. Потенциометр подключается к ЖК-дисплею по порту V0. Выбор ЖК-дисплея в качестве способа отображения информации обусловлен тем, что жидкокристаллические индикаторы, отображающие информацию в символах (текст и числа), — самые недорогие, простые и удобные в использовании[2].

Подключения портов представлены в таблице 2.

Таблица 2 – Таблица подключения портов.

Из Arduino uno r3	4	5	6	7	8	9	10	11
К чему	Кнопка "1"	Кнопка "0"	На ЖК-дисплее порт RS	На ЖК-дисплее порт E	На ЖК-дисплее порт DB4	На ЖК-дисплее порт DB5	На ЖК-дисплее порт DB6	На ЖК-дисплее порт DB7

Вся оставшаяся логическая структура сумматоров и обработка данных реализована в программной части Arduino при помощи логических функций XOR, OR, AND. Пример реализации одноразрядного полного сумматора на данных функциях изображен на рисунке 4.

```

110 char HSM(char a1, char a2){
111     char lastPrev = prev;
112     if(OR_1(AND_1(prev, XOR_1(a1, a2)), AND_1(a1, a2))=='1')
113         prev = '1';
114     else
115         prev = '0';
116     if(XOR_1(XOR_1(a1, a2), lastPrev)=='1')
117         return '1';
118     else return '0';
119 }

```

Рисунок 4 – Код одноразрядного полного сумматора двоичных кодов

Код базовых функций изображен на рисунке 5.

```

89 char XOR_1(char a1, char a2) {
90     if(a1 == a2)
91         return '0';
92     else
93         return '1';
94 }
95
96 char AND_1(char a1, char a2) {
97     if (a1 == '1' && a2 == '1')
98         return '1';
99     else
100         return '0';
101 }
102
103 char OR_1(char a1, char a2){
104     if(a1 == '1' || a2 == '1')
105         return '1';
106     else
107         return '0';
108 }

```

Рисунок 5 – Код логических функций XOR, OR, AND

Переменные a_1 и a_2 передают разряды чисел A_i и B_i соответственно, переменная $prev$ – значение переноса P_{i-1} , формируемого в прошлой итерации. При запуске программы значение переноса $prev$ стандартно задано '0'. Функция возвращает разряд S_i результата.

Также в коде реализованы функции для перевода чисел из прямого кода в дополнительный.

Таким образом в сумматор вводятся два семизрядных числа в прямом коде, которые при необходимости переводятся в дополнительный, складываются через цепь одноразрядных полных сумматоров, при необходимости результат переводится из дополнительного в прямой код и выводится на экран.

Иногда, при сложении чисел возникает переполнение разрядной сетки. Об этом свидетельствует перенос из знакового разряда при сложении отрицательных чисел или несовпадение знакового разряда результата со знаковыми разрядами складываемых положительных чисел. Такой случай обозначается на ЖК-экране единицей в правом нижнем углу.

В результате был получен макет семизрядного двоичного сумматора на Arduino. Работа помогает понять, каким образом проводятся арифметические операции в ЭВМ.

В дальнейшем можно сделать расширение до стандартной размерности чисел в ЭВМ. Это можно сделать на базе микрокомпьютера Raspberry Pi 3 Model B или STM32 Nucleo F401RE. В первом случае можно подключить многопоточность для сложения одновременно нескольких чисел. Во втором случае из-за большого количества портов и большей мощности удобно изменить способ отображения данных, что лучше для больших чисел. Однако для приведенной реализации достаточно и обычной платы на базе микроконтроллера Arduino Uno r3. Это обусловлено

следующими факторами: достаточные показатели производительности, низкая цена, поддержка онлайн эмуляторов, активное пользовательское сообщество, множество документации и информационных ресурсов в открытом доступе, а также возможность разработки на си-подобном языке программирования.

Список использованных источников:

1. *Arduino для изобретателей. Обучение электронике на 10 занимательных проектах: Пер. с англ. / Б. Хуанг, Д. Ранберг.* — СПб.: БХВ-Петербург, 2019. — 288 с.: ил.
2. Бокселл Джон. *Изучаем Arduino. 65 проектов своими руками.* 2-е изд. — СПб.: Питер, 2022. — 448 с.: ил.
3. Хан Т., Стивенсон Р. *Arduino: Подробное руководство.* - М.: ДМК Пресс, 2015. - 672 с.
4. Ашкрофт Д. *Arduino для начинающих.* - М.: ДМК Пресс, 2015. - 320 с.
5. *Арифметические и логические основы вычислительной техники :учеб. пособие / Ю. А. Луцик, И. В. Лукьянова.* – Минск : БГУИР, 2014. –174 с. : ил.
6. *Сервис для построения электронных схем [Электронный ресурс].* – Режим доступа: <https://www.tinkercad.com/circuits>

UDC 004.315.2

BINARY SUMMER ON THE ARDUINO PLATFORM

Bondarev K.D., Zinovenko V.I.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Sidorovich A.S. – senior lecturer of the computer department

Annotation. In this paper, the binary number adder is implemented in additional codes on the Arduino platform by designing a circuit for processing input and output data and writing a program to convert them into a final representation.

Keywords. Parallel Binary Adder, Binary Arithmetic, Logic OR, AND, XOR, Inverse and Complementary Code, Arduino Hardware Platform, Simulation, Emulator, Connections, Resistor, Pins, LCD Display, Potentiometer, Button Press Handling, input and output.