



**Институт Интеллектуальных Кибернетических  
Систем**

**Кафедра «Компьютерные системы и  
технологии»**

**Пояснительная записка  
к курсовому проекту  
на тему «Проектирование процессора ЭВМ»**

Студент гр. \_\_\_\_\_ / \_\_\_\_\_ /  
Руководитель \_\_\_\_\_ / \_\_\_\_\_ /

Москва 2019

## ТЕХНИЧЕСКОЕ ЗАДАНИЕ № 19-15

Выполнить проектирование процессора с заданной системой команд.

### 1. Система памяти

- Оперативная память (ОП) имеет организацию  $16 \times 8$  ячеек. В каждой ячейке ОП хранится два четырехразрядных слова, каждое из которых имеет свой адрес. Таким образом, емкость ОП равна 32 словам.
- Регистровая память (РП) имеет организацию  $8 \times 4$  ячеек.

### 2. Формат и представление чисел

Дробные числа с фиксированной запятой представлены в дополнительном коде. Под запись числа отводится четыре разряда. Самый левый разряд используется для представления знака.

### 3. Формат команд

Команды двухадресные. Первый операнд хранится в РП и указывается в команде с помощью прямого адреса. Второй операнд хранится в ОП и в зависимости от признака адресации П2 указывается в команде с помощью прямой адресации ( $P2=0$ ) или **постиндексной косвенной вар. 2** адресации ( $P2=1$ ).

Результат длинной операции записывается по адресу 2-го операнда, короткой - по адресу 1-го операнда. Размещение командной информации в формате выполнить самостоятельно.

### 4. Операции

а) **УМНОЖЕНИЕ**. Операция выполняется по алгоритму умножения чисел в дополнительном коде с младших разрядов множителя и сдвигом суммы частичных произведений вправо путем последовательного преобразования множителя. При нулевом значении преобразованного разряда множителя такт суммирования **пропускается**. Первый операнд - множимое, второй - множитель. Результат формируется в 4-х разрядной сетке.

б) **ПЕРЕСЫЛКА ОТРИЦАТЕЛЬНАЯ**. Дополнительный код абсолютного значения второго операнда помещается по адресу первого операнда. Нуль, имеющий положительный знак, остается без изменения. Результат отличается от второго операнда только для положительных чисел; отрицательные числа остаются без изменения. Устанавливается признак результата: 0 – результат равен нулю, 1 – результат меньше нуля.

### 5. Команды переходов

а) **ПЕРЕХОД, ЕСЛИ 1**. Продвинутый адрес в счетчике команд (СК) замещается адресом перехода, если значение признака результата равно 1. Для задания адреса перехода используется относительная адресация, т.е. в команде указывается смещение со знаком в дополнительном коде.

б) **БЕЗУСЛОВНЫЙ ПЕРЕХОД**. Для задания адреса перехода используется относительная адресация, т.е. в команде указывается смещение со знаком в дополнительном коде.

# Оглавление

1	Проектирование блока операций .....	5
1.1	Примеры и алгоритмы выполнения операций.....	5
1.1.1	Алгоритм операции УМНОЖЕНИЕ .....	6
1.1.2	Алгоритм операции ПЕРЕСЫЛКА ОТРИЦАТЕЛЬНАЯ .....	10
1.2	Функциональная схема блока операций.....	12
1.3	Проектирование логических элементов блока операций .....	14
1.3.1	Регистр первого операнда RA .....	14
1.3.2	Регистр второго операнда RB .....	15
1.3.3	Регистр результата RR .....	16
1.3.4	Сумматор CM .....	17
1.3.5	Комбинационная схема KC1.....	18
1.3.6	Комбинационная схема KC2.....	19
1.3.7	Триггер TPR.....	20
1.4	Логическая схема блока операций .....	21
1.5	Микропрограммы выполнения операций .....	23
1.5.1	Алгоритмы выполнения микропрограмм.....	23
1.5.2	Примеры выполнения микропрограмм.....	25
2	Проектирование местного устройства управления .....	28
2.1	Алгоритмы выполнения операций.....	28
2.2	Проектирование логической схемы местного устройства управления.....	28
3	Отработка арифметико-логического устройства.....	28
3.1	Распечатки результатов моделирования .....	28
3.2	Размещение арифметико-логического устройства на кристалле .....	28
3.3	Определение параметров системы синхронизации и времени выполнения операций .....	28
4	Проектирование блока управления командами .....	29
4.1	Форматы команд и способы адресации.....	29
4.1.1	Способы адресации.....	29
4.1.2	Форматы команд .....	30
4.2	Схема алгоритма выполнения команд.....	31
4.3	Функциональная схема блока управления командами.....	33
4.4	Логическая схема блока управления командами .....	36
4.5	Схема алгоритма микропрограммы выполнения команд .....	38
4.6	Алгоритм тестирования блока управления командами .....	39
4.6.1	Пример выполнения алгоритма тестирования .....	40

4.6.2	Таблицы загрузки оперативной и регистровой памяти.....	42
4.7	Распечатки результатов моделирования блока управления командами.....	43
5	Проектирование блока выработки микрокоманд.....	47
5.1	Схема алгоритма микропрограммы выполнения команд .....	47
5.2	Формат микрокоманды.....	48
5.3	Функциональная схема микропрограммного устройства управления .....	49
5.4	Логическая схема микропрограммного устройства управления.....	50
5.4.1	Схема пуска останова микропрограммного устройства управления.....	50
5.4.2	Комбинационные схемы микропрограммного устройства управления .....	50
5.4.3	Полная логическая схема .....	50
5.5	Карта загрузки управляющей памяти.....	51

## 1 Проектирование блока операций

### 1.1 Примеры и алгоритмы выполнения операций

Формат представления чисел

0 разряд	1 разряд	2 разряд	3 разряд
Зн <b>A</b>	<b>A2</b> ст	<b>A1</b>	<b>A0</b> мл
Зн <b>B</b>	<b>B2</b> ст	<b>B1</b>	<b>B0</b> мл
Зн <b>RB</b>	<b>RB2</b> ст	<b>RB1</b>	<b>RB0</b> мл
Зн <b>RA</b>	<b>RA2</b> ст	<b>RA1</b>	<b>RA0</b> мл
Зн <b>RR</b>	<b>RR2</b> ст	<b>RR1</b>	<b>RR0</b> мл

В 0 разряде хранится знак числа. Это позволяет выполнять операцию сдвига таким образом, чтобы в мнимом разряде b4 оказалось значение b3 и так далее. Таким образом при выполнении операции умножения достаточно брать старшие разряды регистра RB.

### 1.1.1 Алгоритм операции УМНОЖЕНИЕ

Умножение в соответствии с заданием выполняется по алгоритму умножения чисел в дополнительном коде с младших разрядов множителя и сдвигом суммы частичных произведений вправо путем последовательного преобразования множителя. Данный алгоритм можно выразить следующей формулой:

$$[A]_д \times [B]_д = \left( \left( \dots \left( (0 + [A]_д \widetilde{b_n}) 2^{-1} + [A]_д \widetilde{b_{n-1}} \right) 2^{-1} + \dots + [A]_д \widetilde{b_2} \right) 2^{-1} + [A]_д \widetilde{b_1} \right) 2^{-1} + [A]_д \widetilde{b_0}$$

Где  $\widetilde{b_k} = b_{k+1} - b_k$ ,  $k = 0..n$   
 $\widetilde{b_{n+1}} = 0$

Для четырехразрядных чисел эта формула выглядит следующим образом:

$$[A]_д \times [B]_д = \left( \left( (0 + [A]_д \widetilde{b_3}) 2^{-1} + [A]_д \widetilde{b_2} \right) 2^{-1} + [A]_д \widetilde{b_1} \right) 2^{-1} + [A]_д \widetilde{b_0}$$

Поясним представленный алгоритм примерами:

Пример 1:

$$[A]_{дк} = 1.010 = -6/8$$

$$[B]_{дк} = 0.110 = 6/8$$

$$A \times B = -36/64 = -0,5625 \approx -5/8 = 1.011_{дк}$$

$k = 3 \quad b_3 = 0 - 0 = 0$		0	0	0	0	RR = 0
		пропуск				
	+	0	0	0	0	RR = RR * 2 <sup>-1</sup>
$k = 2 \quad b_2 = 0 - 1 = -1$		0	1	1	0	RA = RA * b <sub>2</sub>
		0	1	1	0	RR = RR + RA
		0	0	1	1	RR = RR * 2 <sup>-1</sup>
$k = 1 \quad b_1 = 1 - 1 = 0$		пропуск				
	+	0	0	0	1	RR = RR * 2 <sup>-1</sup>
$k = 0 \quad b_0 = 1 - 0 = 1$		1	0	1	0	RA = RA * b <sub>0</sub>
		1	0	1	1	RR = RR + RA

Пример 2:

$$[A]_{\text{ДК}} = 0.101 = 5/8$$

$$[B]_{\text{ДК}} = 1.011 = -5/8$$

$$A \times B = -25/64 = -0,390625 \approx -4/8 = 1.100_{\text{ДК}}$$

		0	0	0	0	RR = 0
$k = 3 \ b_3 = 0 - 1 = -1$	+	1	0	1	1	RA = RA * $b_3$
		1	0	1	1	RR = RR + RA
		1	1	0	1	RR = RR * $2^{-1}$
$k = 2 \ b_2 = 1 - 1 = 0$		пропуск				
		1	1	1	0	RR = RR * $2^{-1}$
$k = 1 \ b_1 = 1 - 0 = 1$	+	0	1	0	1	RA = RA * $b_1$
		0	0	1	1	RR = RR + RA
		0	0	0	1	RR = RR * $2^{-1}$
$k = 0 \ b_0 = 0 - 1 = -1$	+	1	0	1	1	RA = RA * $b_0$
		1	1	0	0	RR = RR + RA

Пример 3:

$$[A]_{\text{ДК}} = 0.111 = 7/8$$

$$[B]_{\text{ДК}} = 1.111 = -1/8$$

$$A \times B = -7/64 = -0,109375 \approx -1/8 = 1.111_{\text{ДК}}$$

		0	0	0	0	RR = 0
$k = 3 \ b_3 = 0 - 1 = -1$	+	1	0	0	1	RA = RA * $b_3$
		1	0	0	1	RR = RR + RA
		1	1	0	0	RR = RR * $2^{-1}$
$k = 2 \ b_2 = 1 - 1 = 0$		пропуск				
		1	1	1	0	RR = RR * $2^{-1}$
$k = 1 \ b_1 = 1 - 1 = 0$		пропуск				
		1	1	1	1	RR = RR * $2^{-1}$
$k = 0 \ b_0 = 1 - 1 = 0$		пропуск				
		1	1	1	1	RR

Пример 3:

$$[A]_{\text{ДК}} = 0.111 = 7/8$$

$$[B]_{\text{ДК}} = 0.101 = 5/8$$

$$A \times B = 35/64 = 0,546875 \approx 4/8 = 0.100_{\text{ДК}}$$

		0	0	0	0	RR = 0
$k = 3 \ b_3 = 0 - 1 = -1$	+	1	0	0	1	RA = RA * $b_3$
		1	0	0	1	RR = RR + RA
		1	1	0	0	RR = RR * $2^{-1}$
$k = 2 \ b_2 = 1 - 0 = 1$	+	0	1	1	1	RA = RA * $b_2$
		0	0	1	1	RR = RR + RA
		0	0	0	1	RR = RR * $2^{-1}$
$k = 1 \ b_1 = 0 - 1 = -1$	+	1	0	0	1	RA = RA * $b_1$
		1	0	1	0	RR = RR + RA
		1	1	0	1	RR = RR * $2^{-1}$
$k = 0 \ b_0 = 1 - 0 = 1$	+	0	1	1	1	RA = RA * $b_0$
		0	1	0	0	RR = RR + RA



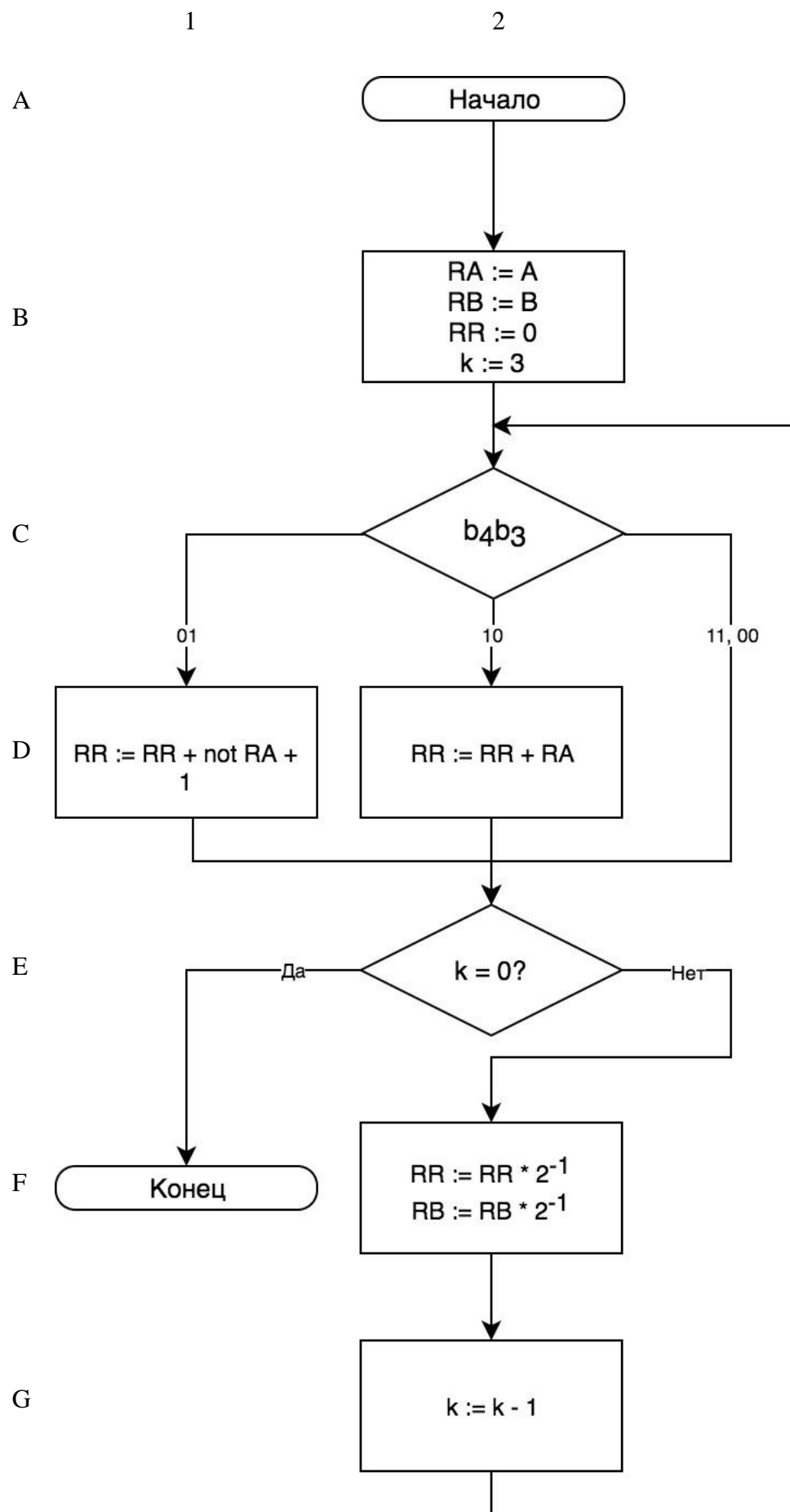


Рис. 1.1 Алгоритм операции УМНОЖЕНИЕ

### 1.1.2 Алгоритм операции ПЕРЕСЫЛКА ОТРИЦАТЕЛЬНАЯ

Дополнительный код абсолютного значения второго операнда помещается по адресу первого операнда. Устанавливается признак результата: 0 – результат равен нулю, 1 – результат меньше нуля. Поясним алгоритм примерами:

Пример 1

$$[B]_{\text{ДК}} = 1.101$$

---

$$\begin{array}{l} RR = 1.101 \\ PR = 1 \end{array}$$

Пример 2

$$[B]_{\text{ДК}} = 0.101$$

---

$$\begin{array}{l} RR = 1.011 \\ PR = 1 \end{array}$$

Пример 3

$$[B]_{\text{ДК}} = 0.000$$

---

$$\begin{array}{l} RR = 0.000 \\ PR = 0 \end{array}$$

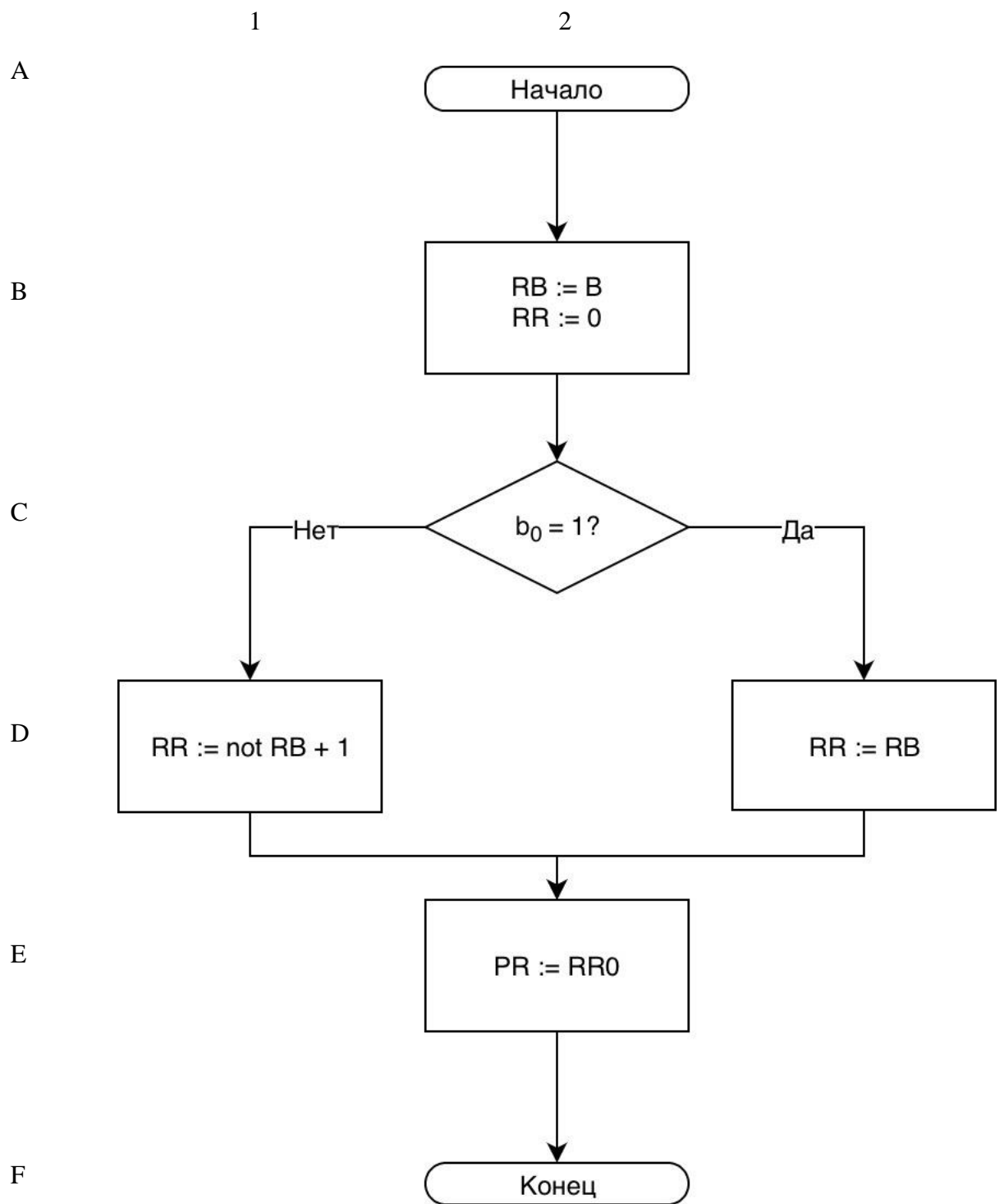


Рис. 1.2 Алгоритм операции ПЕРЕСЫЛКА ОТРИЦАТЕЛЬНАЯ

## 1.2 Функциональная схема блока операций

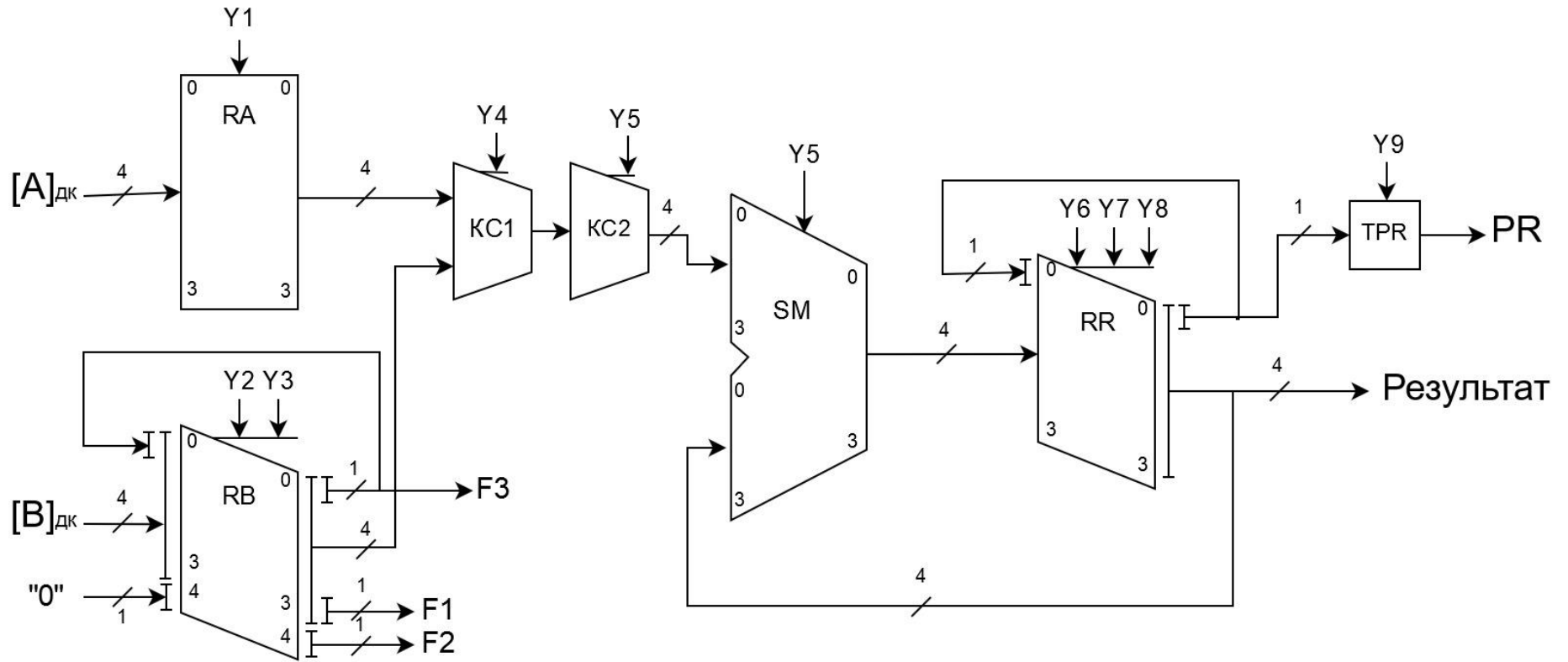


Рис. 1.3. Функциональная схема блока операций

На рис. 1.3 представлена функциональная схема БО, в следующей таблице 1.1 приведены все элементы данной схемы блока операций с их назначением и функциями.

Таблица 1.1

Функциональное назначение блоков

Обозначение блока	Назначение
RA	Регистр первого операнда; 4 разряда Параллельная загрузка [A]
RB	Регистр второго операнда; 4 разряда Параллельная загрузка [B], арифметический сдвиг вправо на 1 разряд
RR	Регистр результата; 4 разрядов; Установка в 0, параллельная загрузка, арифметический сдвиг вправо на 1 разряд
TPR	Триггер признака результата; 1 разряд Установка в 0, загрузка
KC1	Выбор операнда (RA или RB)
KC2	Инверсия операнда
SM	Сумматор; 4 разряда

## 1.3 Проектирование логических элементов блока операций

### 1.3.1 Регистр первого операнда RA

Данный регистр является четырехразрядным регистром хранения. Наиболее подходящим для реализации функций регистра RA является регистр FD4CE (рис. 1.4).

Таблица 1.2

Таблица управляющих сигналов RA

Обозначение	Назначение
У1	
0	Хранение
1	Разрешение загрузки

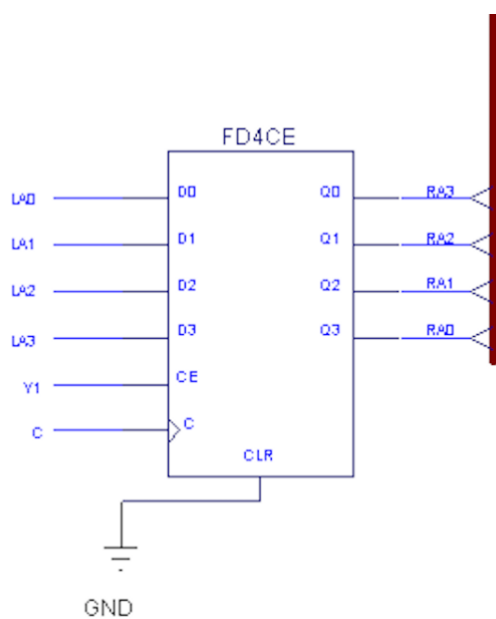


Рис. 1.4. Логическая схема RA

### 1.3.2 Регистр второго операнда RB

Данный четырехразрядный регистр RB, в отличие от регистра RA, должен также осуществлять микрооперацию арифметического сдвига вправо на 1 разряд. Для этой цели был выбран универсальный регистр сдвига SR4CLE (рис. 1.5).

Таблица 1.3  
Таблица управляющих сигналов RB

Обозначение		Назначение
Y2	Y3	
0	0	Хранение
1	0	Разрешение загрузки
0	1	Сдвиг вправо 1 разряд

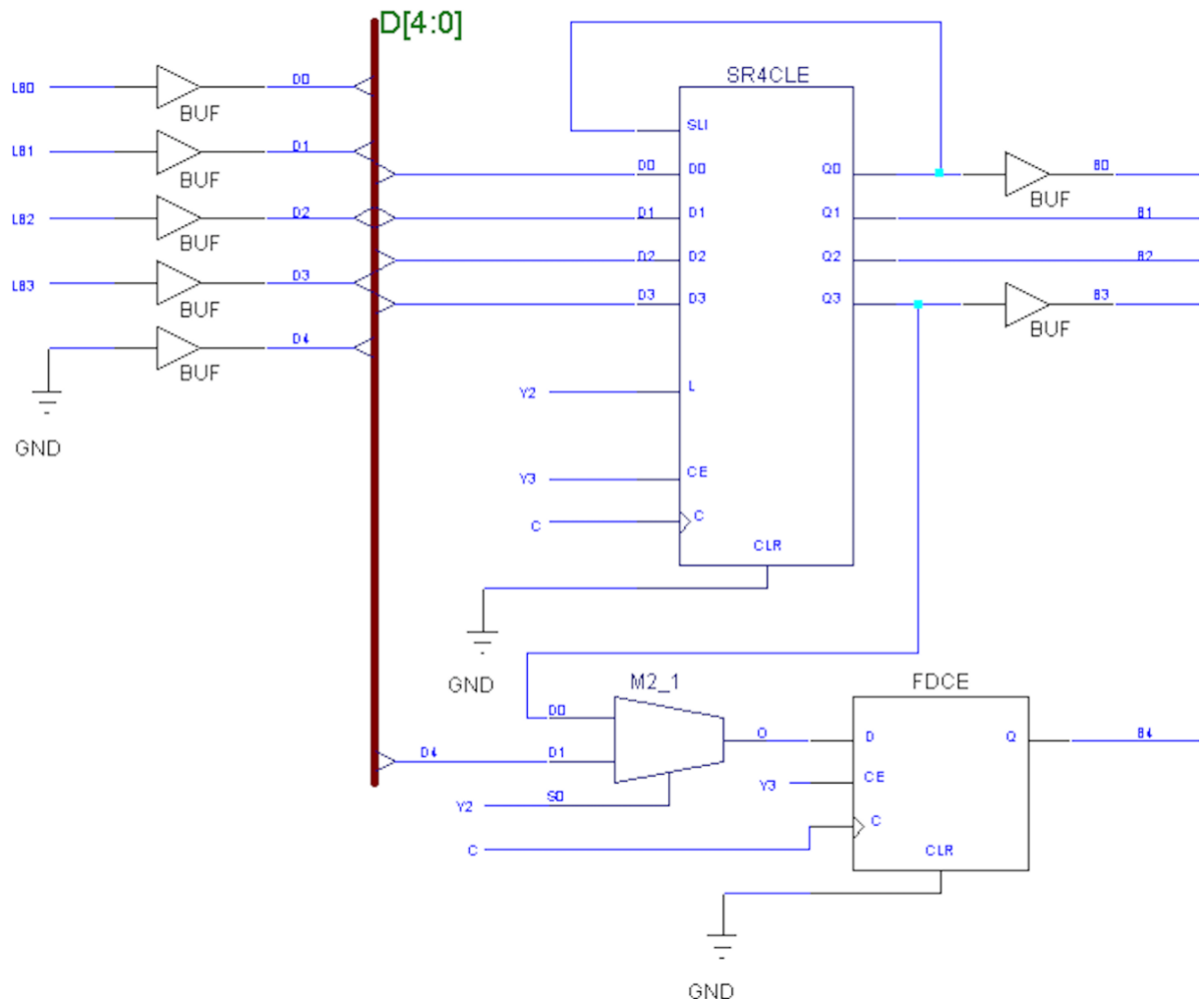


Рис. 1.5. Логическая схема RB

### 1.3.3 Регистр результата RR

Регистр RR должен выполнять микрооперации параллельной загрузки, хранения, арифметического сдвига вправо на 1 разряд и установки в нуль. С учетом данных требований был выбран универсальный регистр сдвига SR8CLE (рис. 1.6). В таблице 1.4 представлены обозначения управляющих сигналов регистра и их назначение.

Причем результат умножения хранится в четырех старших разрядах регистра. Таким образом происходит обрезание результата микрооперации без округления.

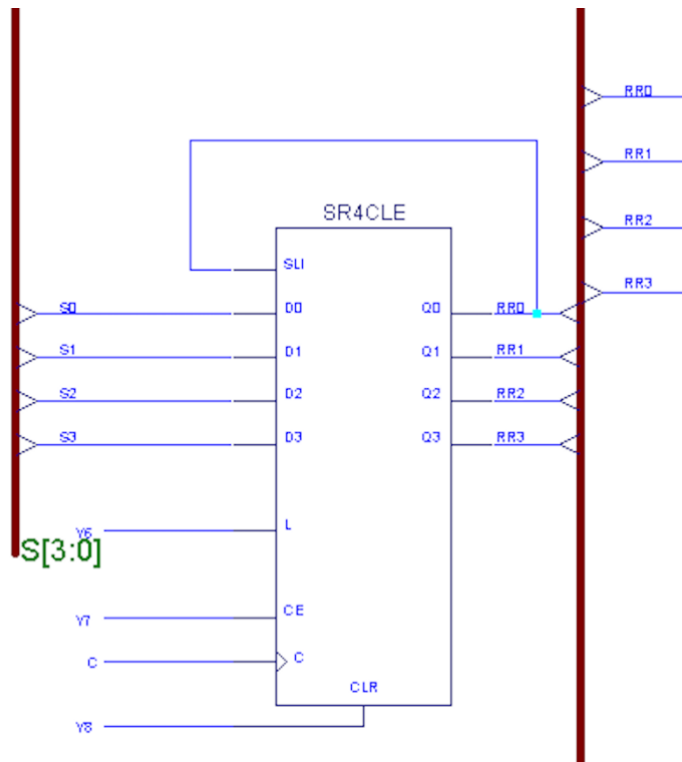


Рис 1.6 Логическая схема RR

Таблица 1.4

Таблица управляющих сигналов RR

Обозначение			Назначение
Y6	Y7	Y8	
0	0	1	Асинхронный сброс в 0
0	0	0	Хранение
0	1	0	Сдвиг вправо 1 разряд
1	0	0	Разрешение загрузки



### 1.3.4 Сумматор СМ

Для построения сумматора возьмем восьмиразрядный сумматор ADD8, подавая сигнал Y5 во время корректирующего шала для получения инверсного значения А или В в дополнительном коде (рис. 1.7).

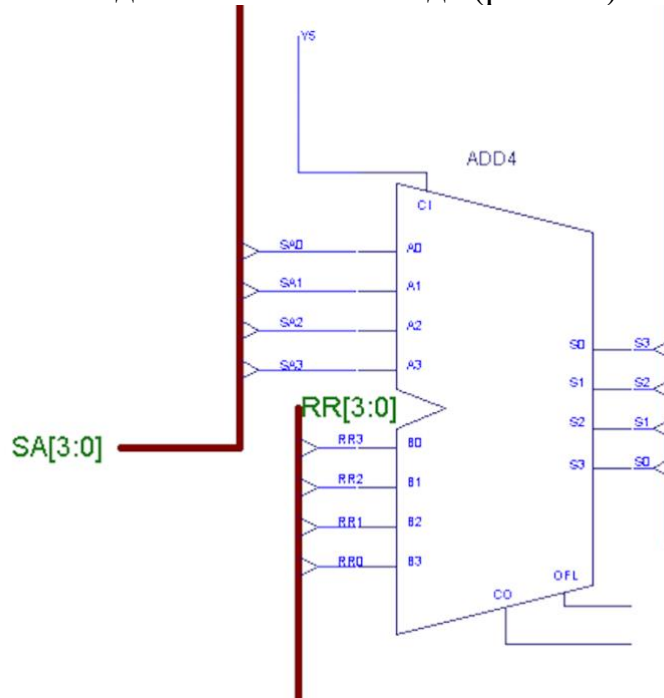


Рис 1.7 Сумматор с корректирующим входом

### 1.3.5 Комбинационная схема КС1

С помощью данной комбинационной схемы происходит выбор операнда А или В. За выбор операнда отвечает У4 (рис. 1.8).

В таблице 1.5 представлены назначения управляющего сигнала комбинационной схемы.

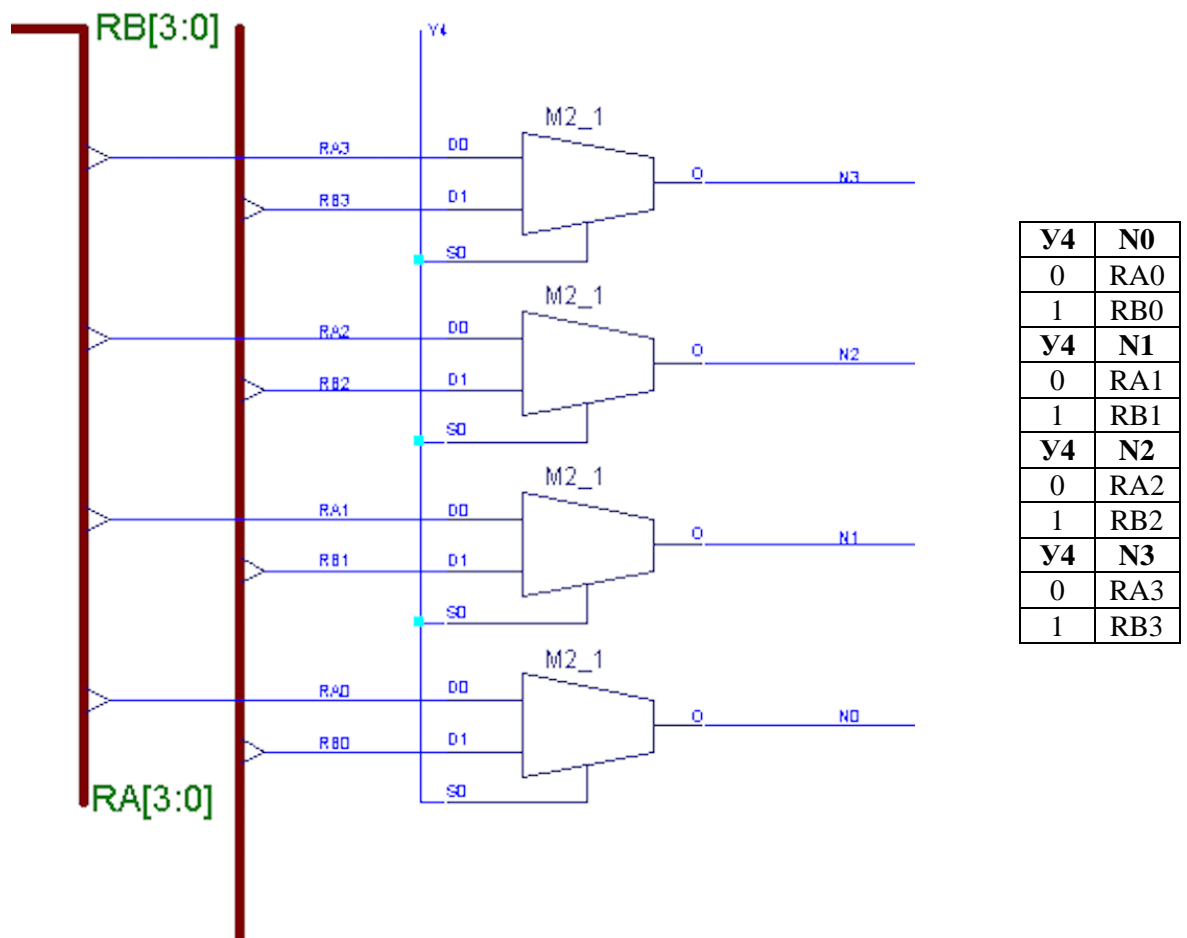


Рис. 1.8 Комбинационная схема КС1

Таблица 1.5

Таблица управляющих сигналов КС1

Обозначение	Назначение
У4	
0	Подача А на сумматор
1	Подача В на сумматор

1.3.6 Комбинационная схема КС2

Комбинационная схема инвертирует входное значение (рис. 1.9). В качестве управляющего сигнала поступает У5. Таблица назначений представлена ниже (Таблица 1.6)

У5	Ni	XOR
0	0	0
0	1	1
1	0	1
1	1	0

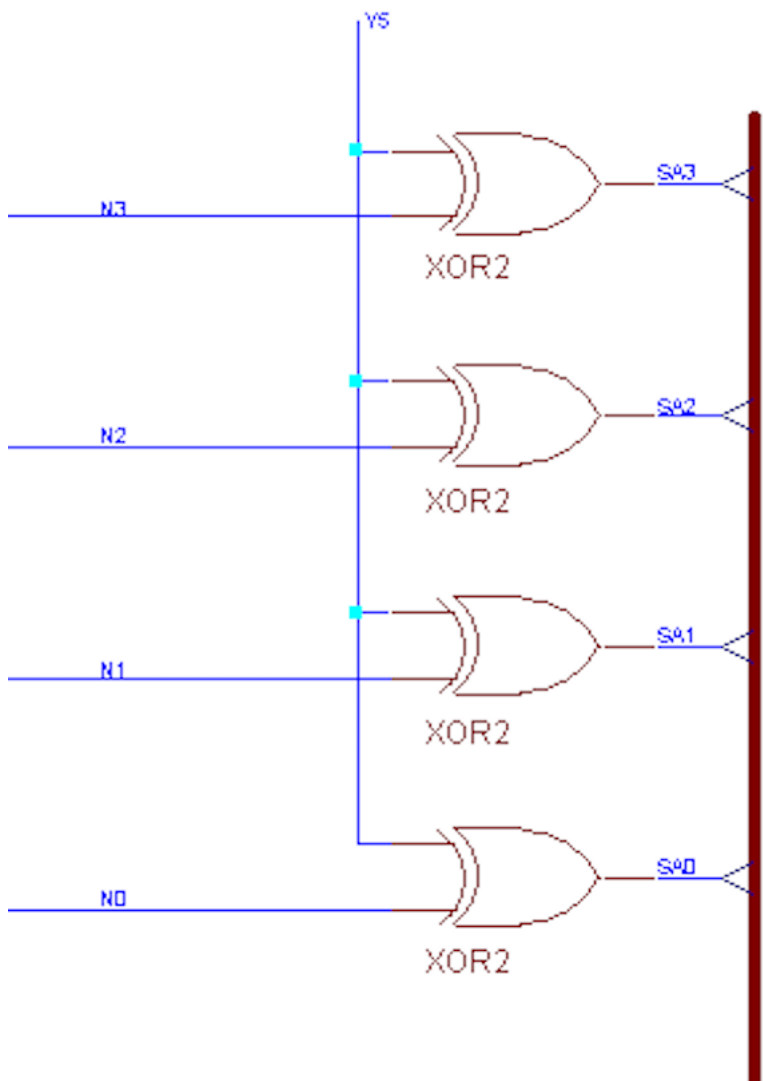


Рис. 1.9 Комбинационная схема КС2

Таблица 1.6

Таблица управляющих сигналов КС2

Обозначение	Назначение
У5	
0	
	Подача N на сумматор

1	Подача $\bar{N}$ на сумматор
---	------------------------------

### 1.3.7 Триггер TPR

Триггер TPR хранит знак результата – признак результата (рис. 1.11). Может принимать значения 0 – результат равен 0, 1 – результат меньше 0. Управляется сигналом У9. Его назначение представлено в таблице 1.8

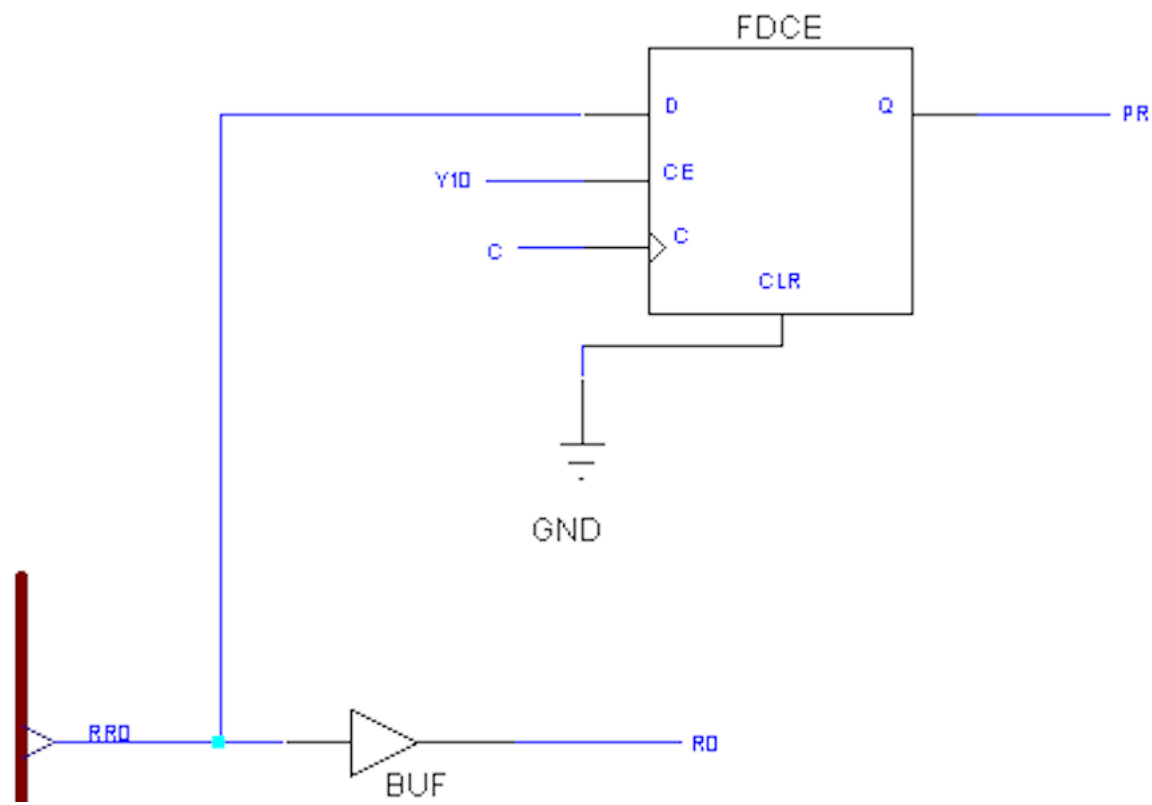


Рис. 1.11 Триггер TPR

Таблица 1.8  
Таблица управляющих сигналов TPR

Обозначение	Назначение
У9	
0	Хранение
1	Разрешение загрузки

## 1.4 Логическая схема блока операций

На рис. 1.12 приведена полная логическая схема блока операций

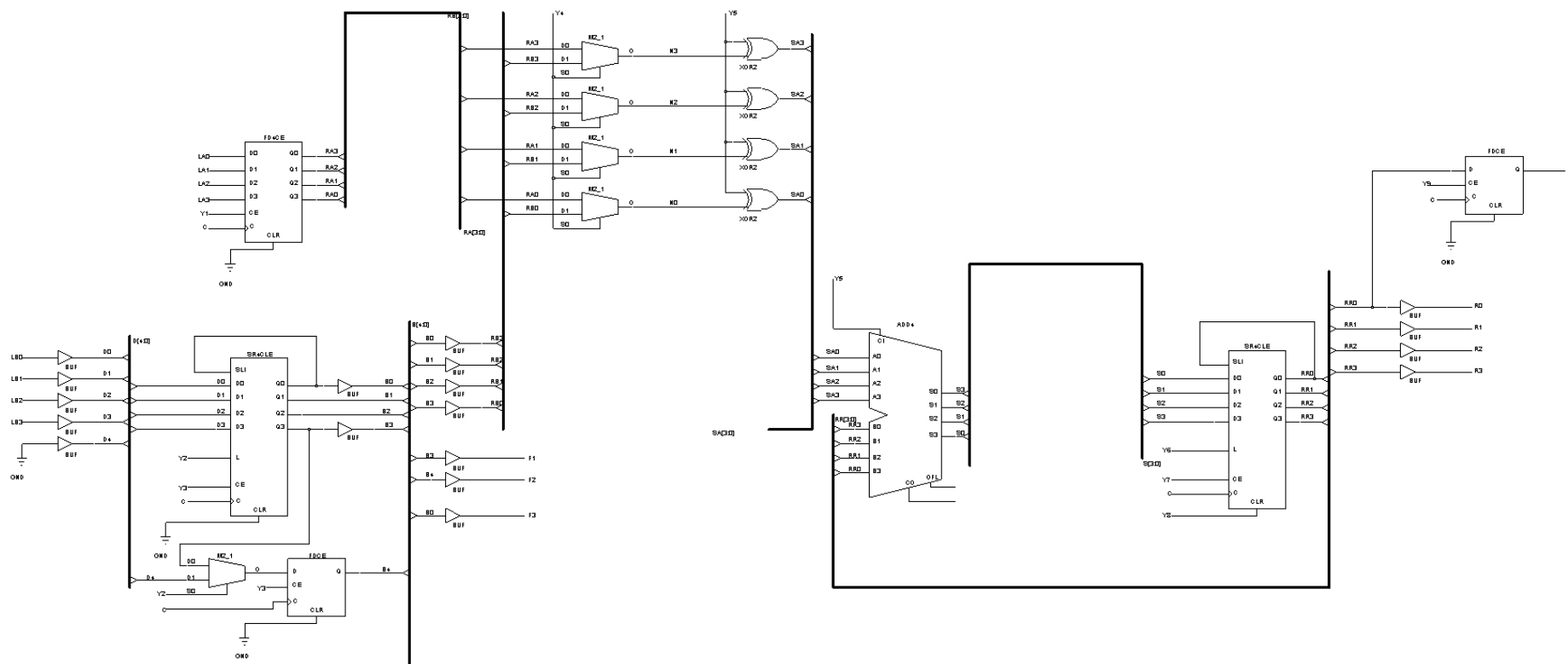


Рис. 1.12. Логическая схема блока операций

В таблицах 1.9 и 1.10 сведены описания управляющих сигналов и сигналов-признаков блока операций.

### Определение управляющих сигналов БО

Таблица 1.9

Таблица управляющих сигналов

Сигнал	Операция
У1	Загрузка А в RA
У2	Загрузка В в RB
У3	Арифметический сдвиг RB вправо на 1 разряд
У4	Прием значения из RA или из RB
У5	Подача инверсного входа на сумматор
У6	Загрузка RR
У7	Арифметический сдвиг RR вправо на 1 разряд
У8	Сброс RR
У9	Загрузка TPR

### Определение сигналов-признаков БО

Таблица 1.10

Таблица сигналов признаков

Сигнал	Признак
F1	3 разряд В
F2	4 разряд В
F3	Знак числа В

## 1.5 Микропрограммы выполнения операций

### 1.5.1 Алгоритмы выполнения микропрограмм

На рис. 1.13 и 1.14 представлены блок-схемы алгоритмов микропрограмм выполнения операций Умножение и Пересылка отрицательная.

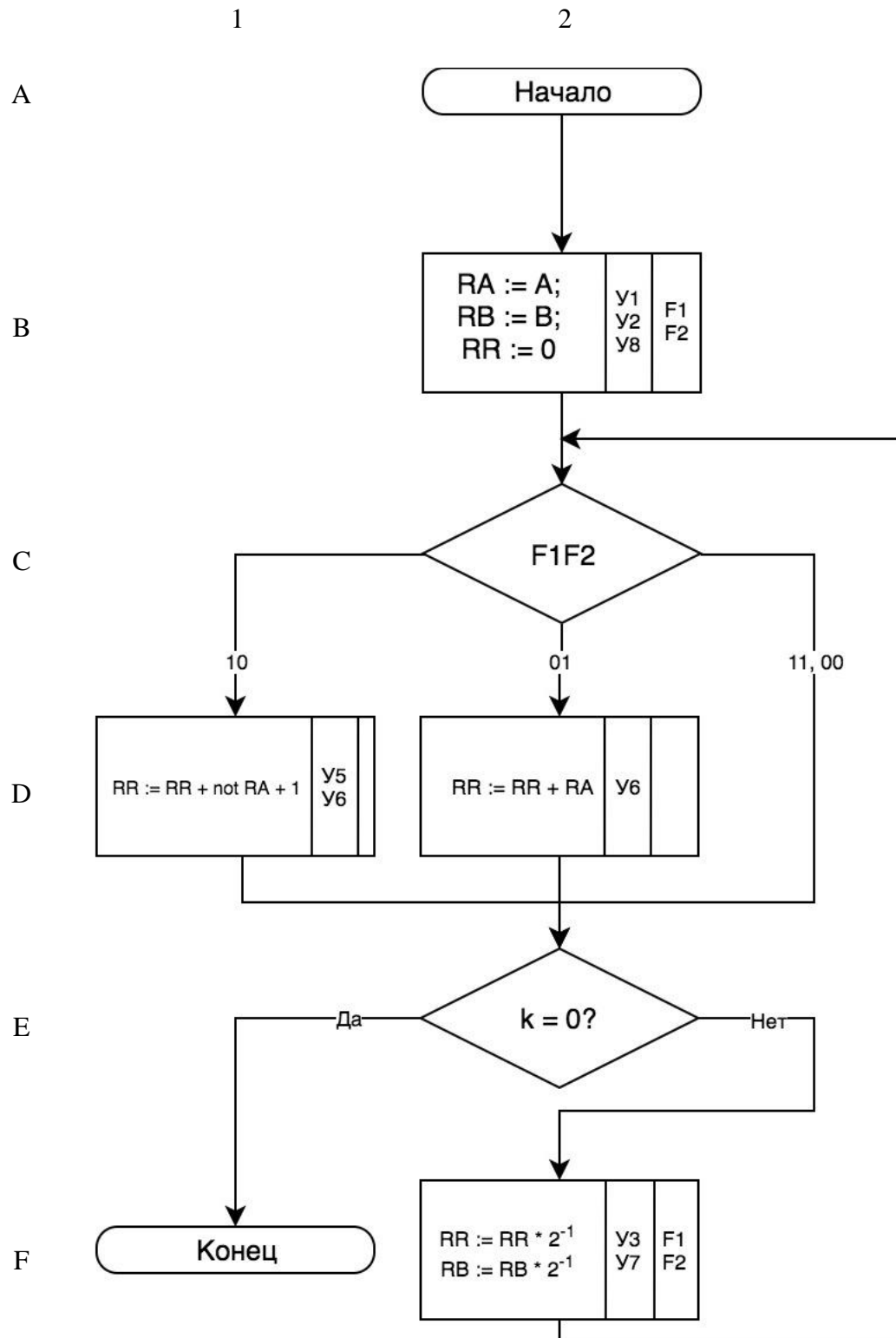


Рис. 1.13 Микропрограмма операции УМНОЖЕНИЕ

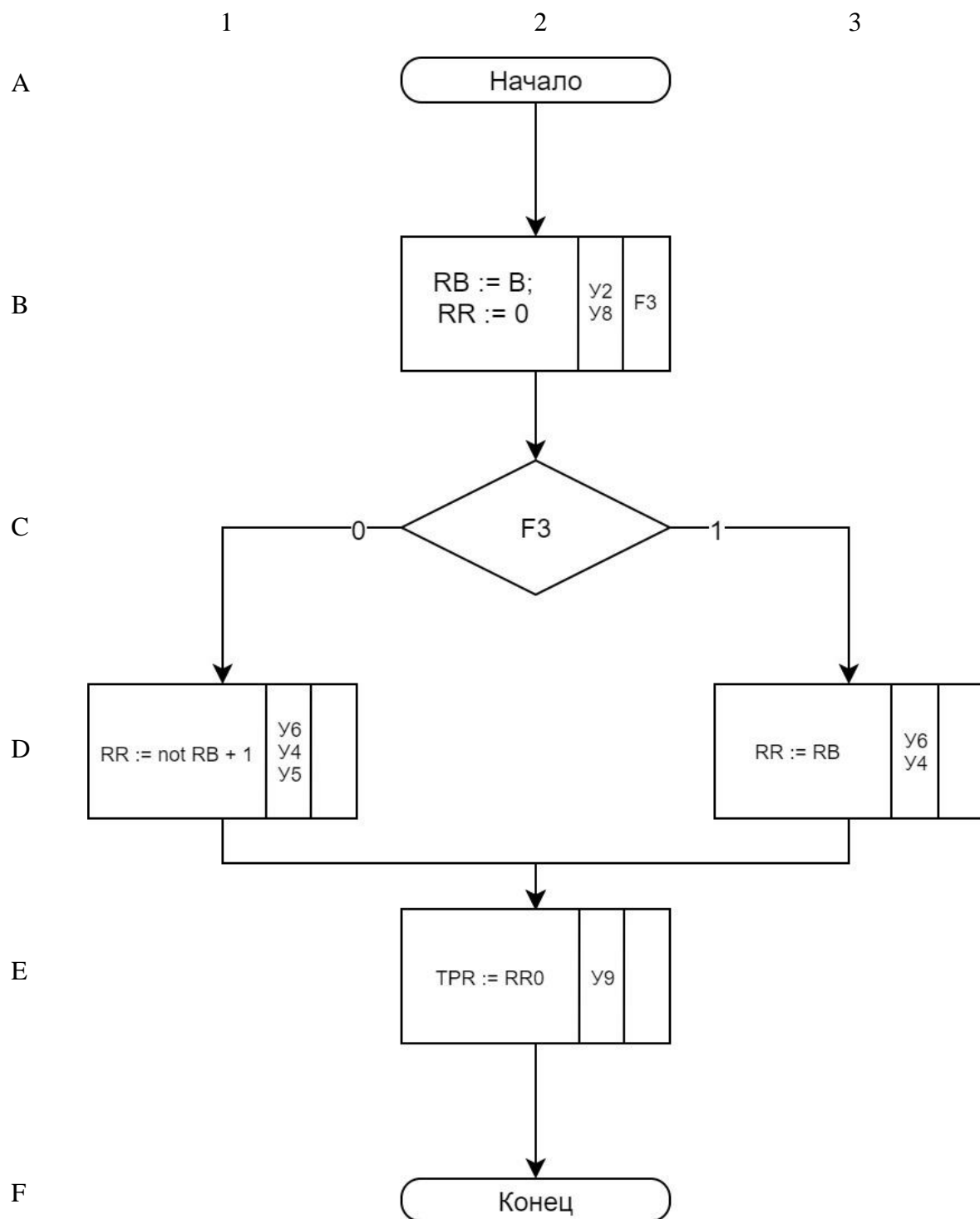
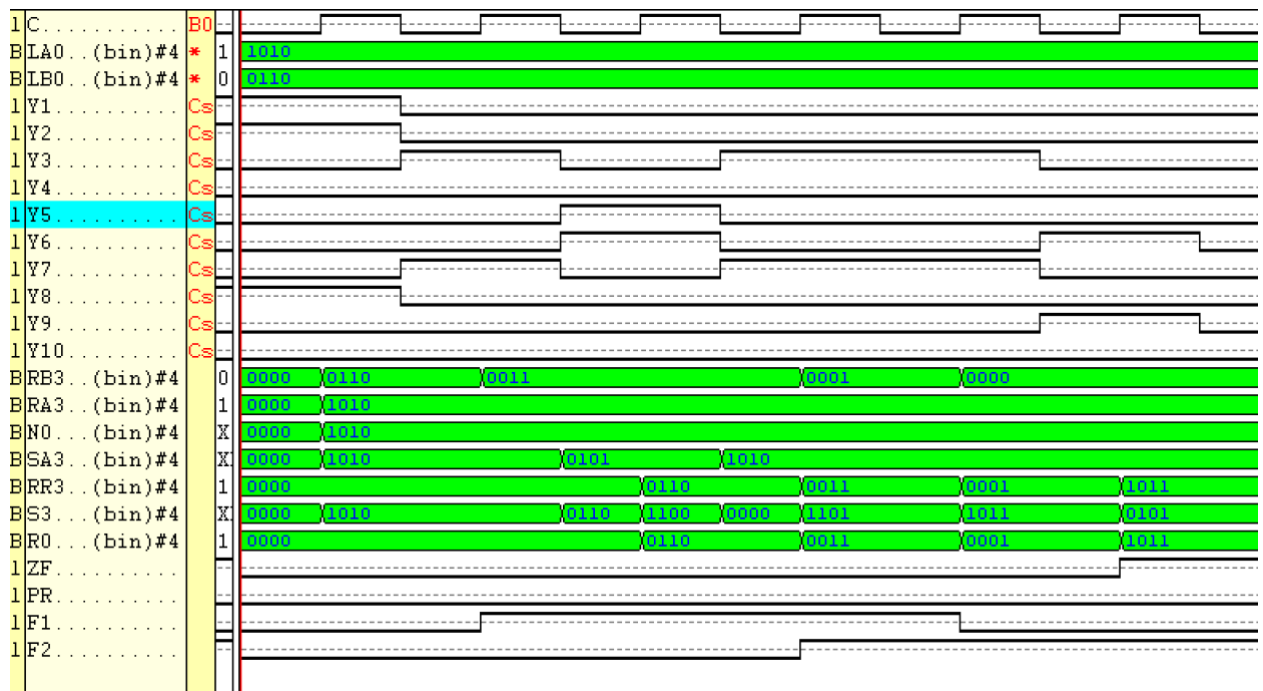


Рис. 1.14 Микропрограмма операции ПЕРЕСЫЛКА ОТРИЦАТЕЛЬНАЯ

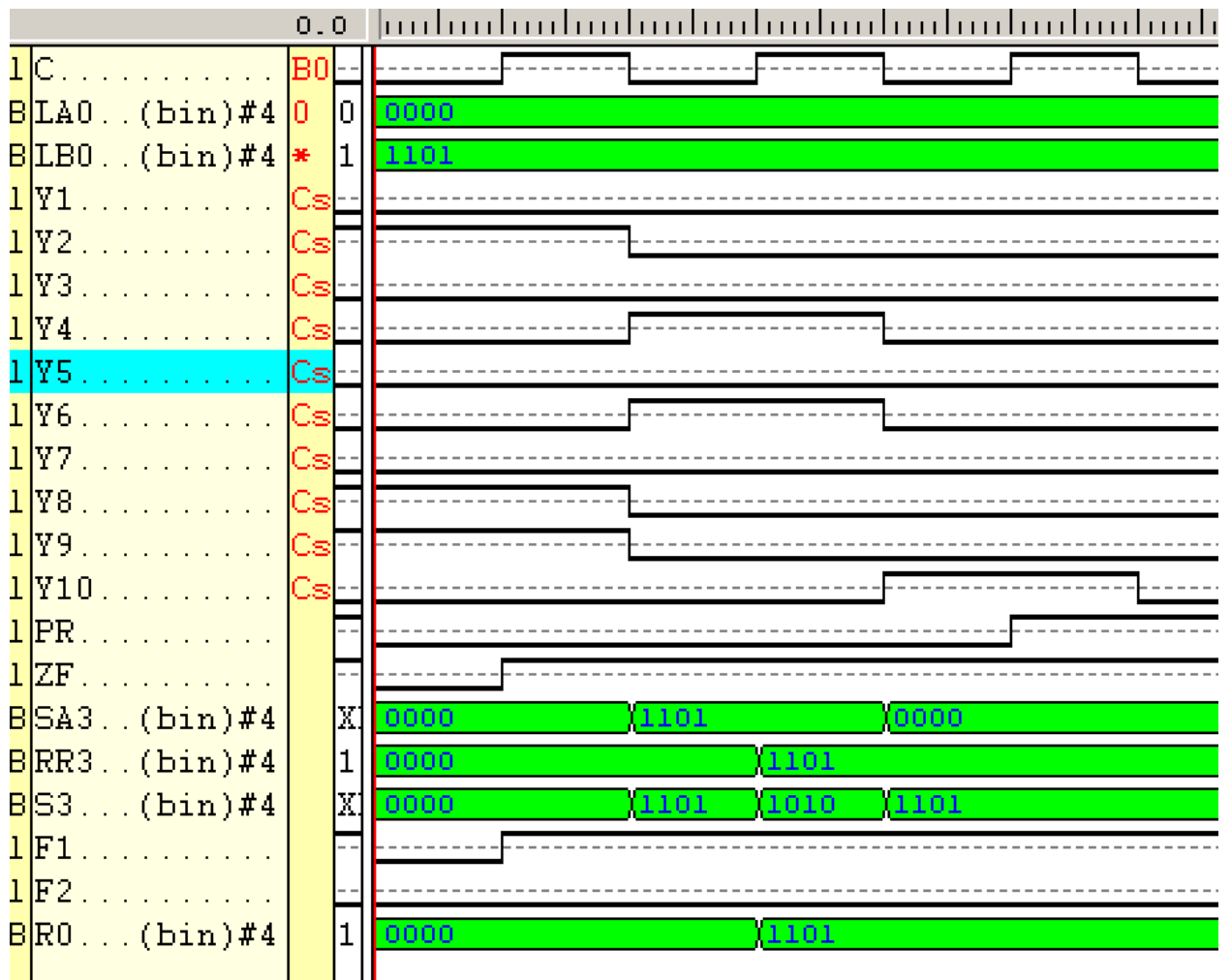
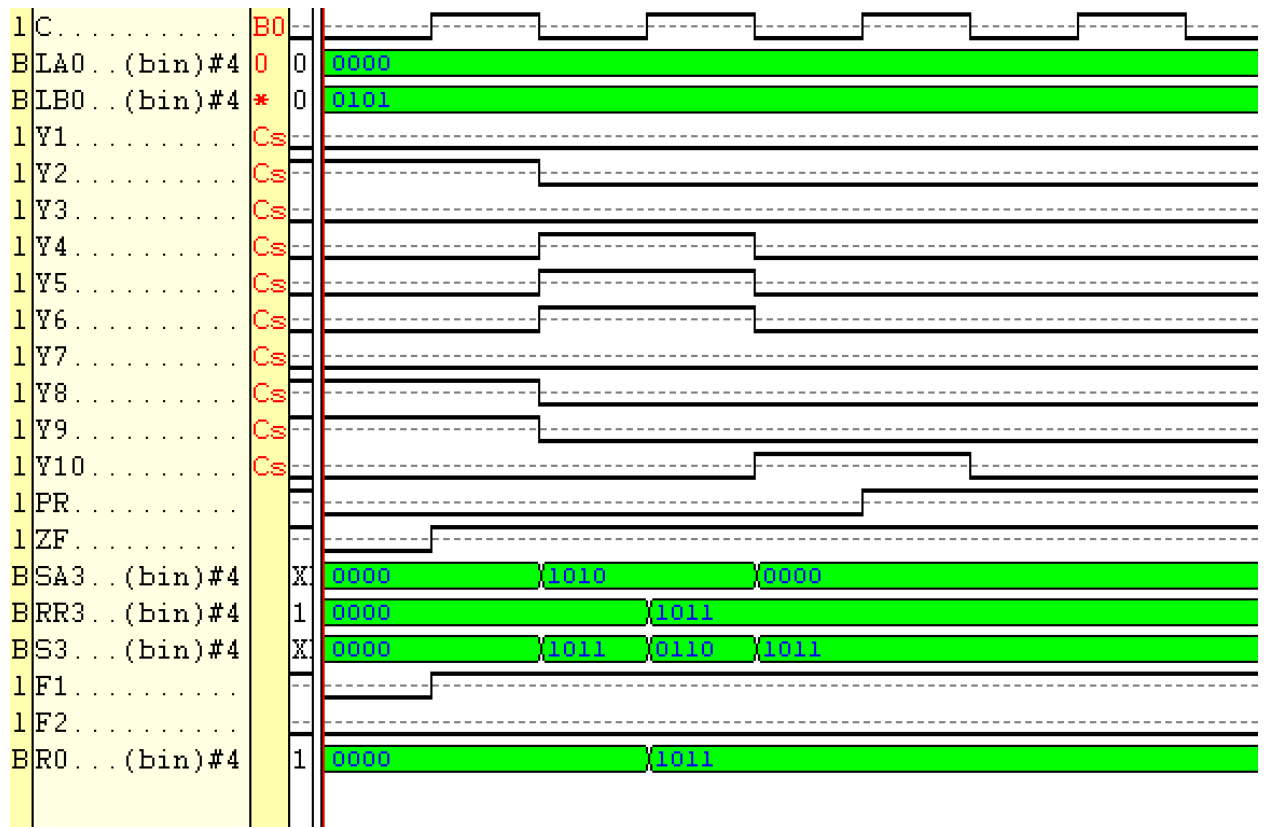


### 1.5.2 Примеры выполнения микропрограмм

На рис. 1.15-1.17 приведены результаты моделирования схемы блока операций при выполнении операции умножения и пересылки отрицательной для примеров, приведенных в пункте 1.1.1.



$k = 3 \quad b_3 = 0 - 0 = 0$	+	0	0	0	0	RR = 0
		пропуск				
$k = 2 \quad b_2 = 0 - 1 = -1$	+	0	0	0	0	RR = RR * 2 <sup>-1</sup>
		0	1	1	0	RA = RA * b <sub>2</sub>
$k = 1 \quad b_1 = 1 - 1 = 0$	+	0	1	1	0	RR = RR + RA
		0	0	1	1	RR = RR * 2 <sup>-1</sup>
$k = 0 \quad b_0 = 1 - 0 = 1$	+	пропуск				
		0	0	0	1	RR = RR * 2 <sup>-1</sup>
		1	0	1	0	RA = RA * b <sub>0</sub>
		1	0	1	1	RR = RR + RA





## **2 Проектирование местного устройства управления**

### **2.1 Алгоритмы выполнения операций**

### **2.2 Проектирование логической схемы местного устройства управления**

## **3 Отработка арифметико-логического устройства**

### **3.1 Распечатки результатов моделирования**

### **3.2 Размещение арифметико-логического устройства на кристалле**

### **3.3 Определение параметров системы синхронизации и времени выполнения операций**

## 4 Проектирование блока управления командами

### 4.1 Форматы команд и способы адресации

Блок управления командами обрабатывает четыре типа команд: умножение, пересылка отрицательная, переход, если 1, безусловный переход. Линейные команды (умножение и пересылка), в зависимости от признака адресации (PA2), могут использовать два разных способа адресации второго операнда. Команды переходов используют относительную адресацию.

#### 4.1.1 Способы адресации

Используемые способы адресации указаны в таблицах 4.1 и 4.2.

Таблица 4.1 – Способы адресации операндов в линейных командах

Команда	Способ адресации операнда		
	Первый операнд	Второй операнд	
		PA2 = 0	PA2 = 1
Умножение	Прямая регистровая	Прямая	Постиндексная косвенная вар. 2
Пересылка отрицательная			

Таблица 4.2 – Способы адресации операндов в нелинейных командах

Команда	Способ адресации адреса перехода
Переход, если 1	Относительная
Безусловный переход	

Способы адресации, используемые в командах продемонстрированы на рис. 4.1 – 4.3

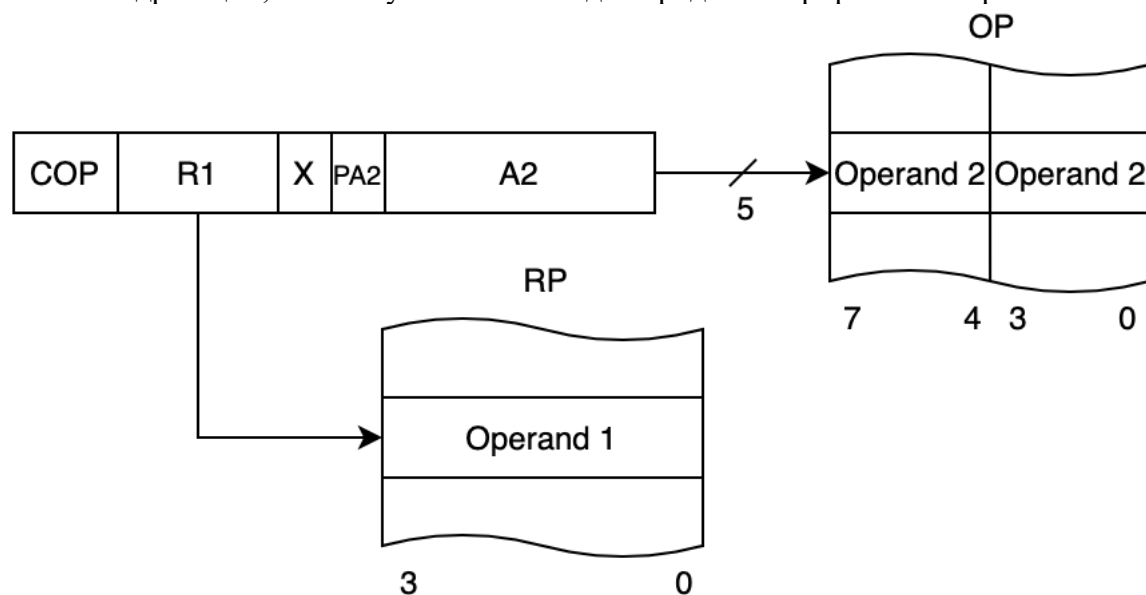


Рис. 4.1 – Адресация операндов в линейных командах, использующих прямую адресацию

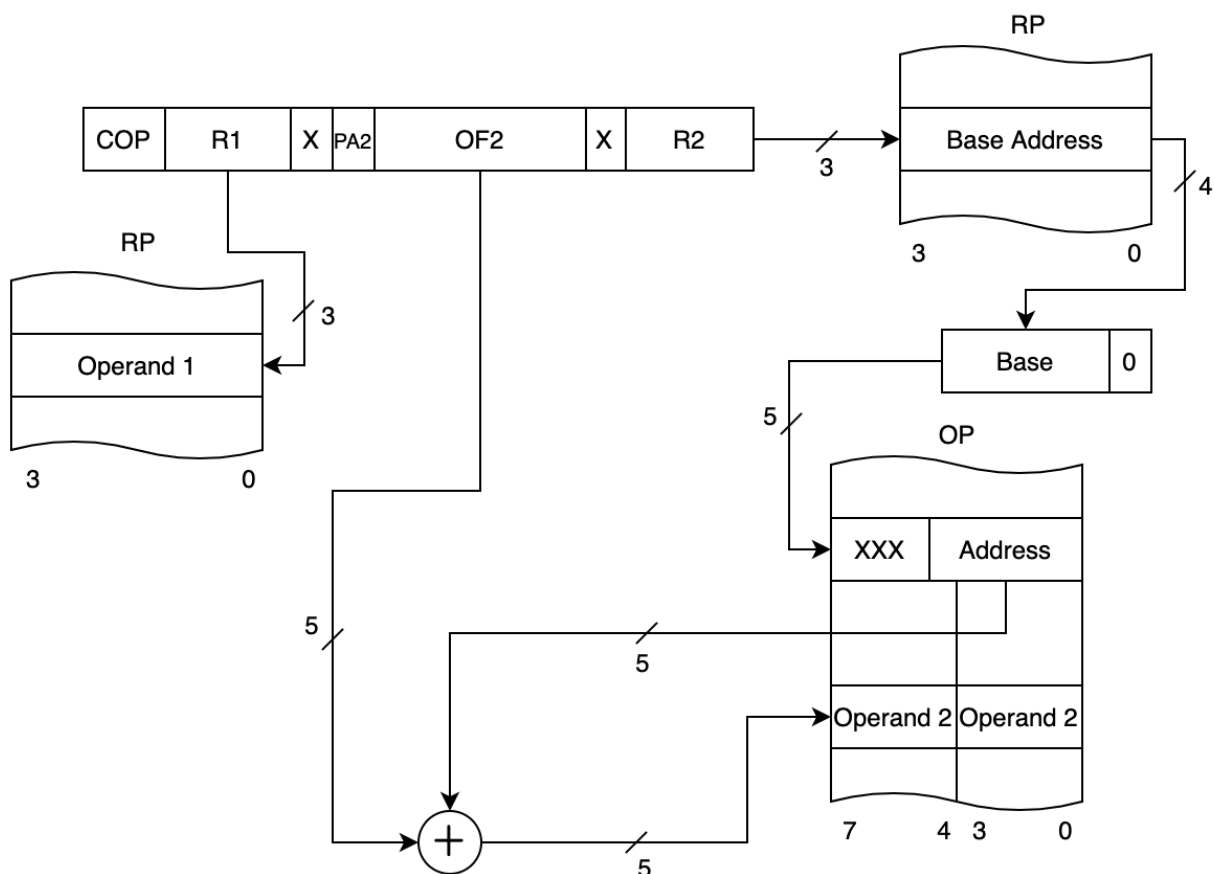


Рис. 4.2 – Адресация операндов в линейных командах, использующих постиндексную косвенную вар. 2 адресацию

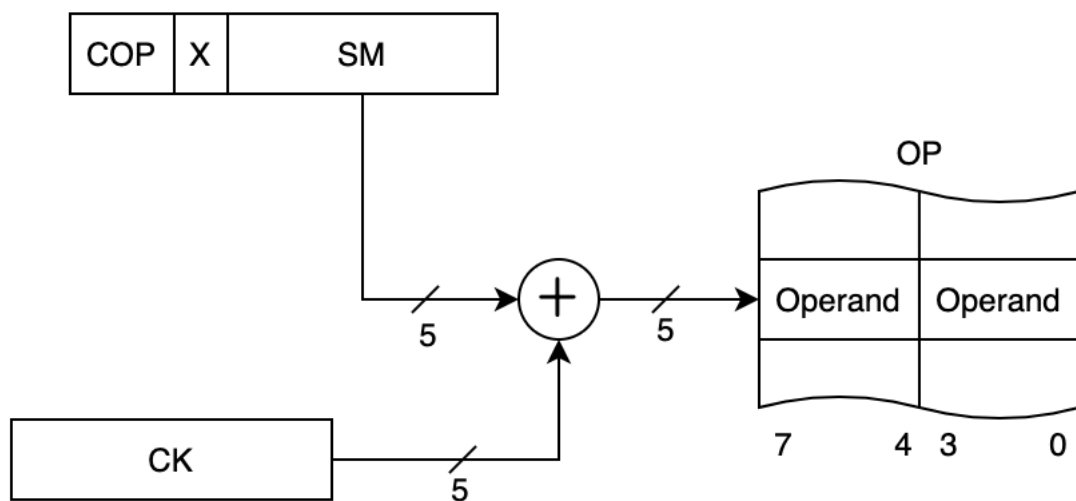


Рис. 4.3 – Адресация операндов в нелинейных командах, использующих относительную адресацию

#### 4.1.2 Форматы команд

При определении формата команд решающее значение влияла оптимизация расположения команд в оперативной памяти. Этим обусловлено различие размера команд и их кратность 4 битам.

Разработанные форматы команд представлены на рис. 4.4

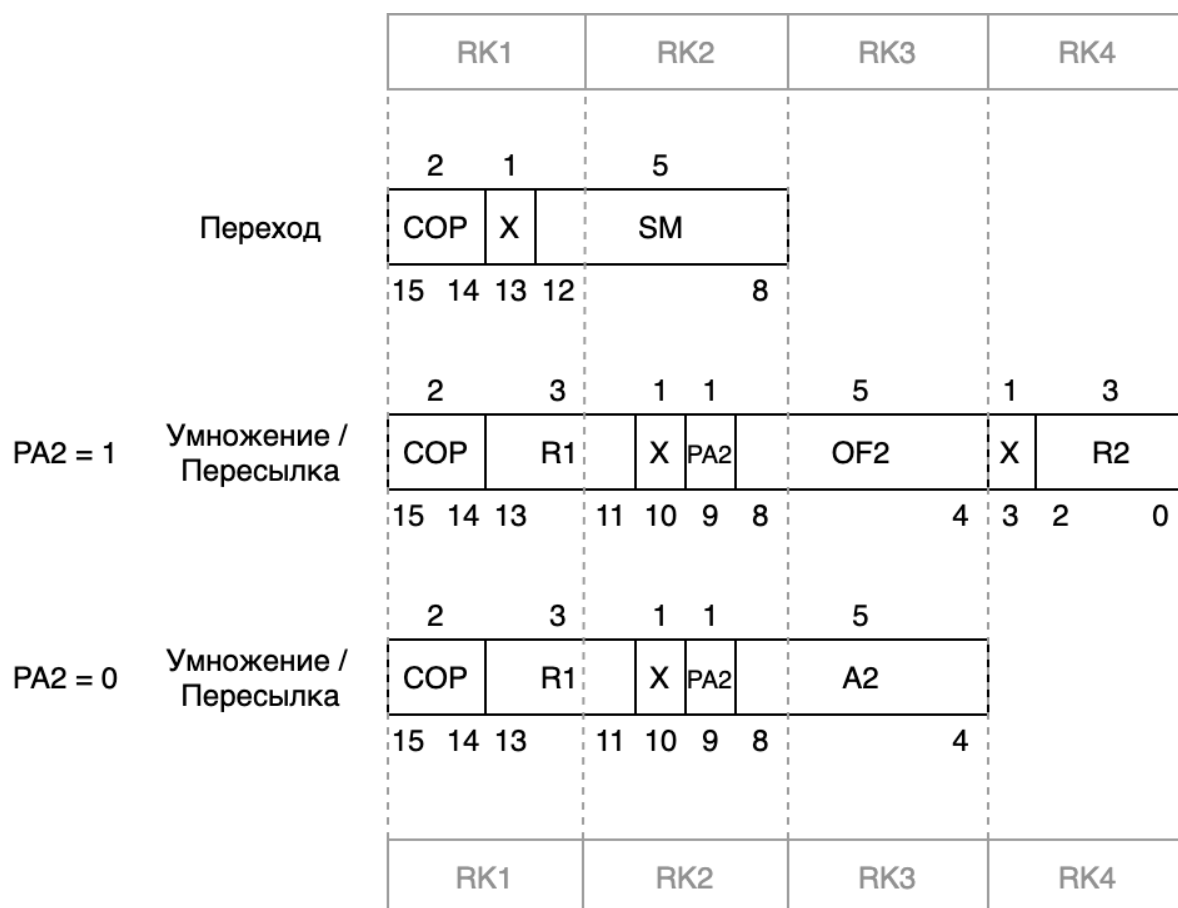


Рис. 4.4 – Форматы команд

Под кодирование выполняемой операции было выделено два разряда. Выбранный способ кодирования COP представлен в таблице 4.3

Таблица 4.3 – Кодирование операций

Разряды поля COP		Операция
15 разряд (COP1)	14 разряд (COP0)	
0	0	Умножение
0	1	Пересылка отрицательная
1	0	Переход, если 1
1	1	Безусловный переход

## 4.2 Схема алгоритма выполнения команд

Алгоритм выполнения команд блоком управления командами представлен в виде блок-схемы на рис. 4.5

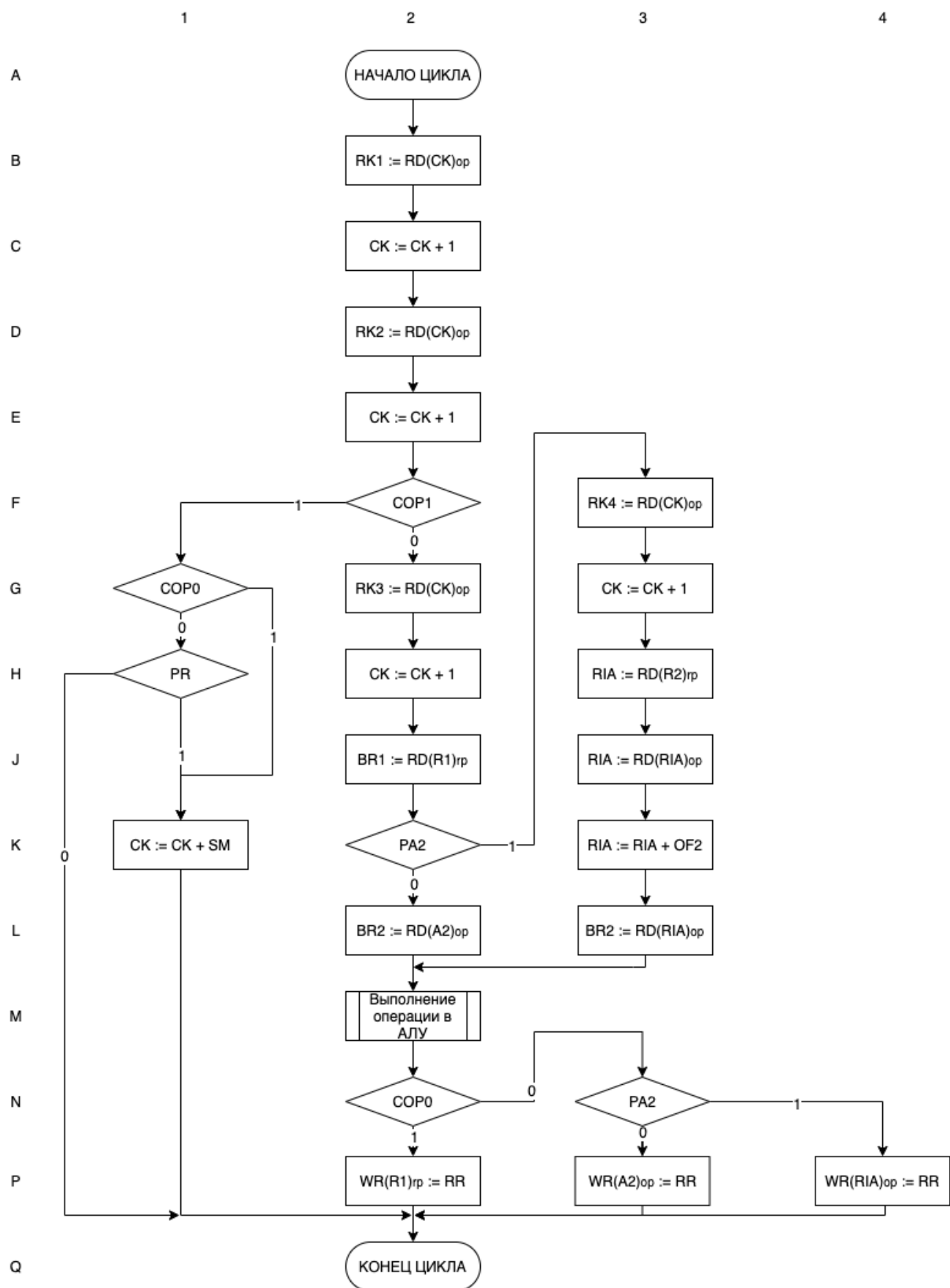


Рис. 4.5 – Блок-схема алгоритма выполнения команд



### 4.3 Функциональная схема блока управления командами

Функциональная схема блока управления командами, работающего с перечисленными форматами команд, представлена на рис. 4.6.

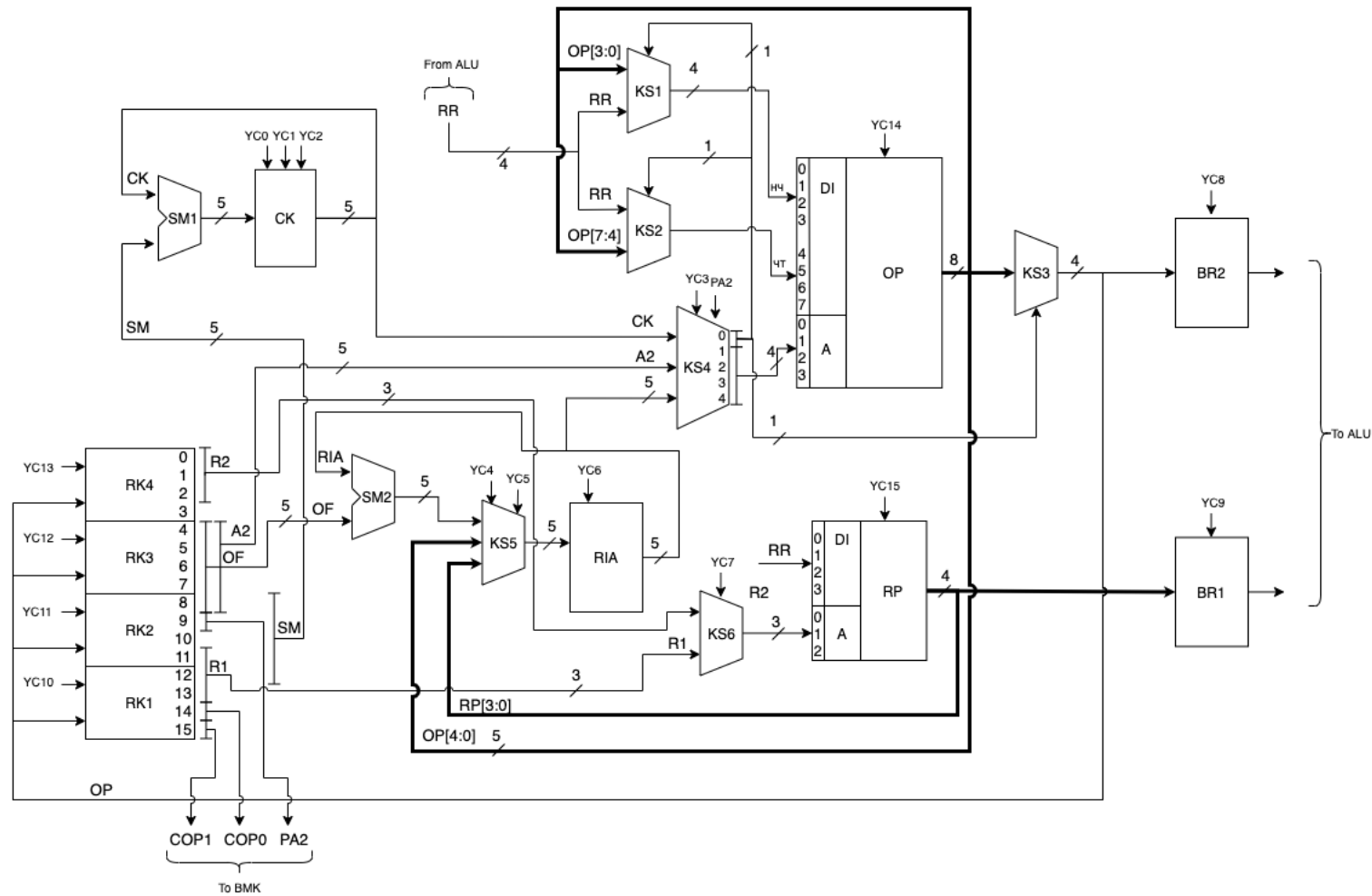


Рис. 4.6 – Функциональная схема БУК.

Список и назначения логических элементов, управляющих сигналах и сигналов-признаков блока управления командами представлены в таблицах 4.4 и 4.5.

Таблица 4.4 – Элементы БУК

Обозначение	Описание
KS1	Коммутатор младших разрядов входа оперативной памяти
KS2	Коммутатор старших разрядов входа оперативной памяти
KS3	Коммутатор выхода оперативной памяти
KS4	Коммутатор адресного входа оперативной памяти
KS5	Коммутатор входа регистра исполнительного адреса
KS6	Коммутатор адресного входа регистровой памяти
SM1	Сумматор адреса перехода
SM2	Сумматор исполнительного адреса
СК	Счетчик команд
RIA	Регистр исполнительного адреса
RK1	Регистр команд (15-12)
RK2	Регистр команд (11-8)
RK3	Регистр команд (7-4)
RK4	Регистр команд (3-0)
OP	Оперативная память
RP	Регистровая память
BR1	Буферный регистр первого операнда
BR2	Буферный регистр второго операнда

Таблица 4.5 – Сигналы-признаки и управляющие сигналы

Обозначение	Описание
YC1	Разрешение счета счетчика команд
YC2	Разрешение загрузки в счетчик команд
YC3	Управляющий вход коммутатора адреса оперативной памяти
YC4	Управляющий вход коммутатора данных регистра исполнительного адреса (1)
YC5	Управляющий вход коммутатора данных регистра исполнительного адреса (2)
YC6	Разрешение загрузки в регистр исполнительного адреса
YC7	Управляющий вход коммутатора адреса регистровой памяти
YC8	Разрешение загрузки в буферный регистр первого операнда
YC9	Разрешение загрузки в буферный регистр второго операнда
YC10	Разрешение загрузки в первый регистр команд
YC11	Разрешение загрузки в второй регистр команд
YC12	Разрешение загрузки в третий регистр команд
YC13	Разрешение загрузки в четвертый регистр команд
YC14	Разрешение загрузки в оперативную память
YC15	Разрешение загрузки в регистровую память
COP0	Младший разряд кода операции
COP1	Старший разряд кода операции
PA2	Признак способа адресации

#### 4.4 Логическая схема блока управления командами

Логические схемы компонентов функциональной схемы таких как сумматоры, регистры, счетчик команд следуют из их предназначения. Модули памяти были реализованы с помощью встроенного конструктора LogiBLOX. Коммутаторы шин, используемые в функциональной схеме, были реализованы с использованием внутренних шин и мультиплексоров. Одна из реализаций представлена на рис. 4.7.

Полная логическая схема представлена на рис. 4.8.

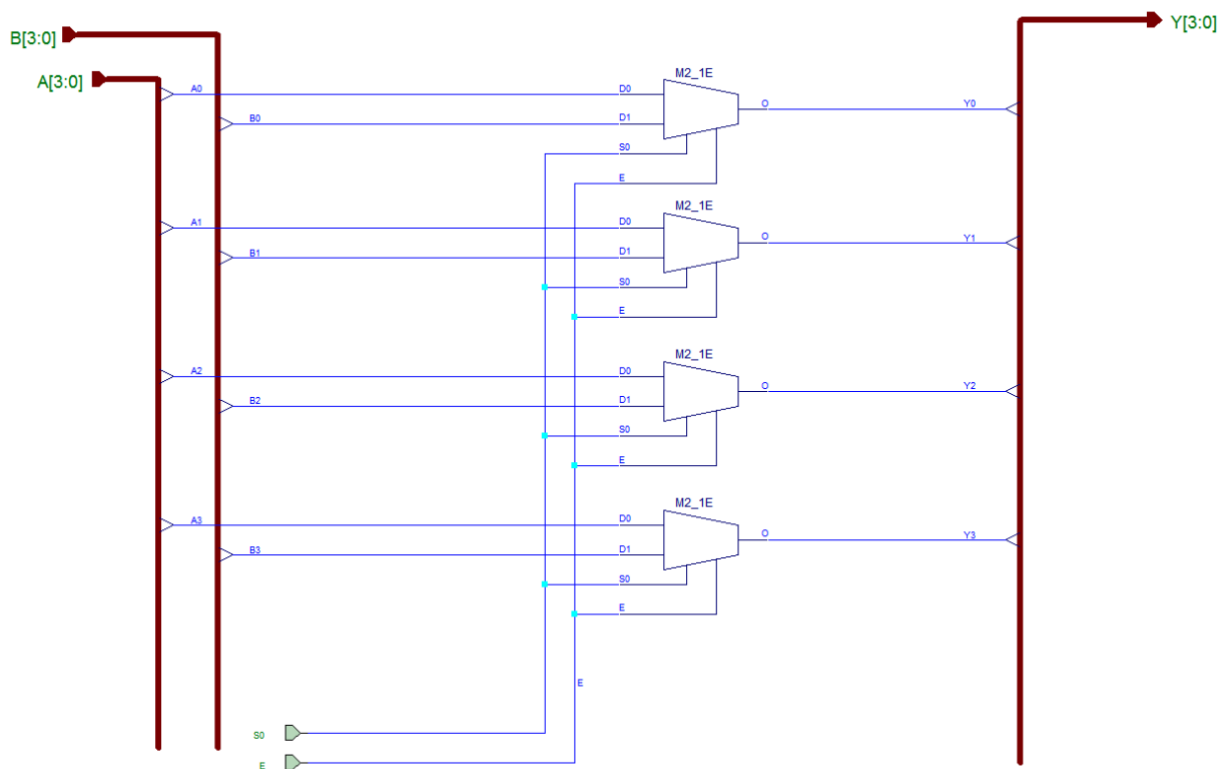


Рис. 4.7 – Реализация коммутатора шин.

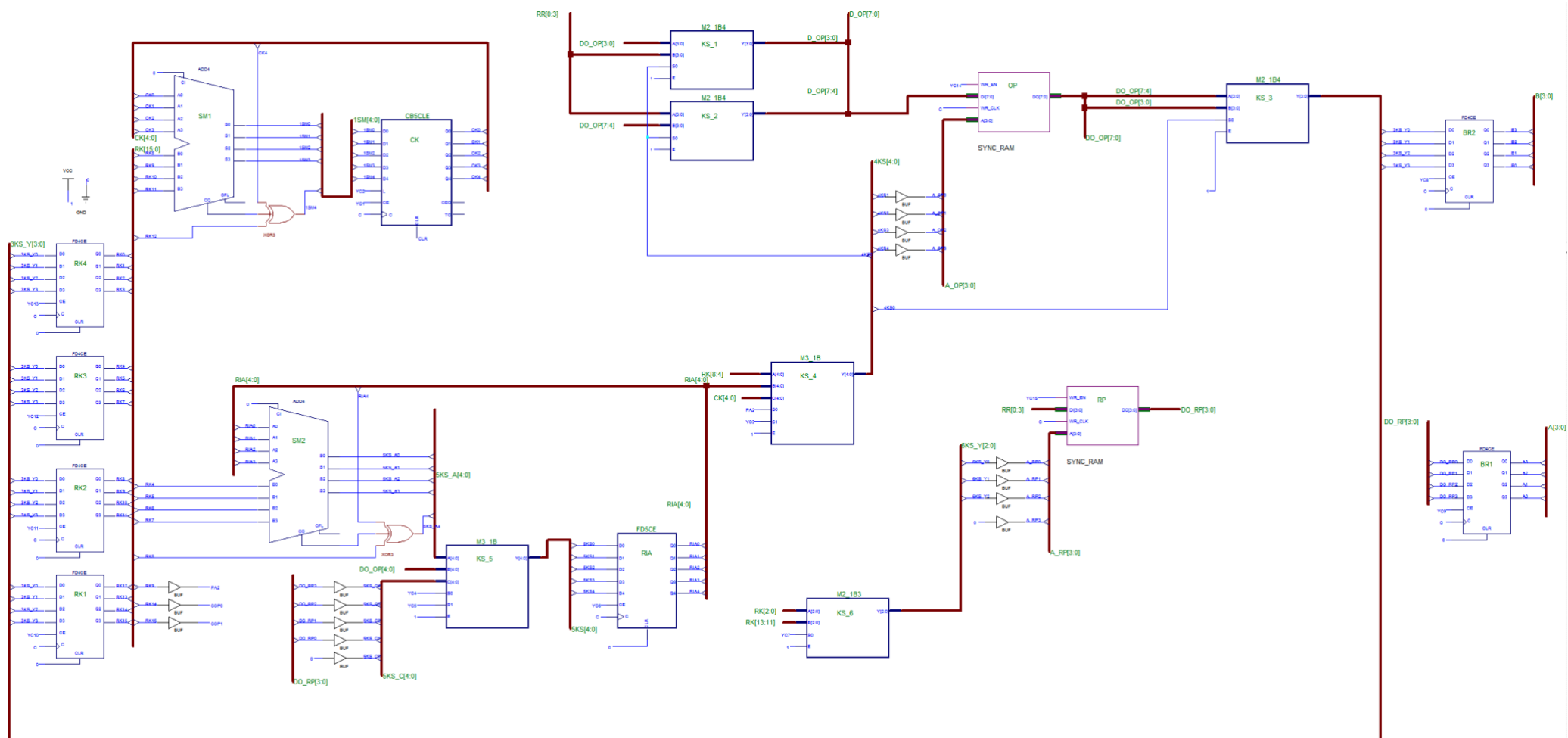


Рис. 4.8 – Логическая схема блока управления командами.

## 4.5 Схема алгоритма микропрограммы выполнения команд

Схема алгоритма, дополненная необходимыми для каждой операции управляющими сигналами представлена на рис. 4.9.

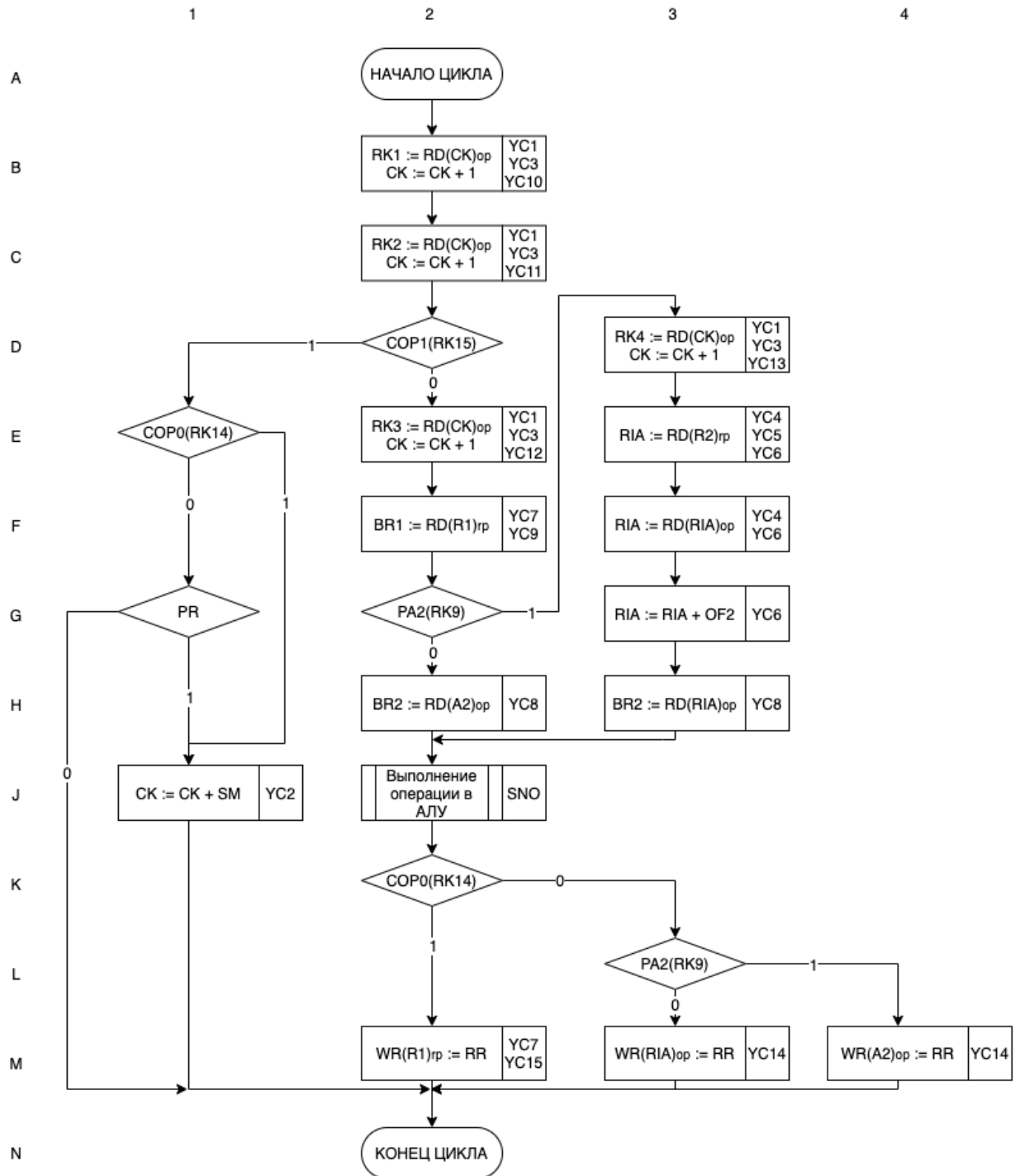


Рис. 4.9 – Схема алгоритма микропрограммы выполнения команд.

#### 4.6 Алгоритм тестирования блока управления командами

Для тестирования блока управления командами был разработан алгоритм тестирования и соответствующая ему программа. Блок-схема данного алгоритма представлена на рис. 4.10.

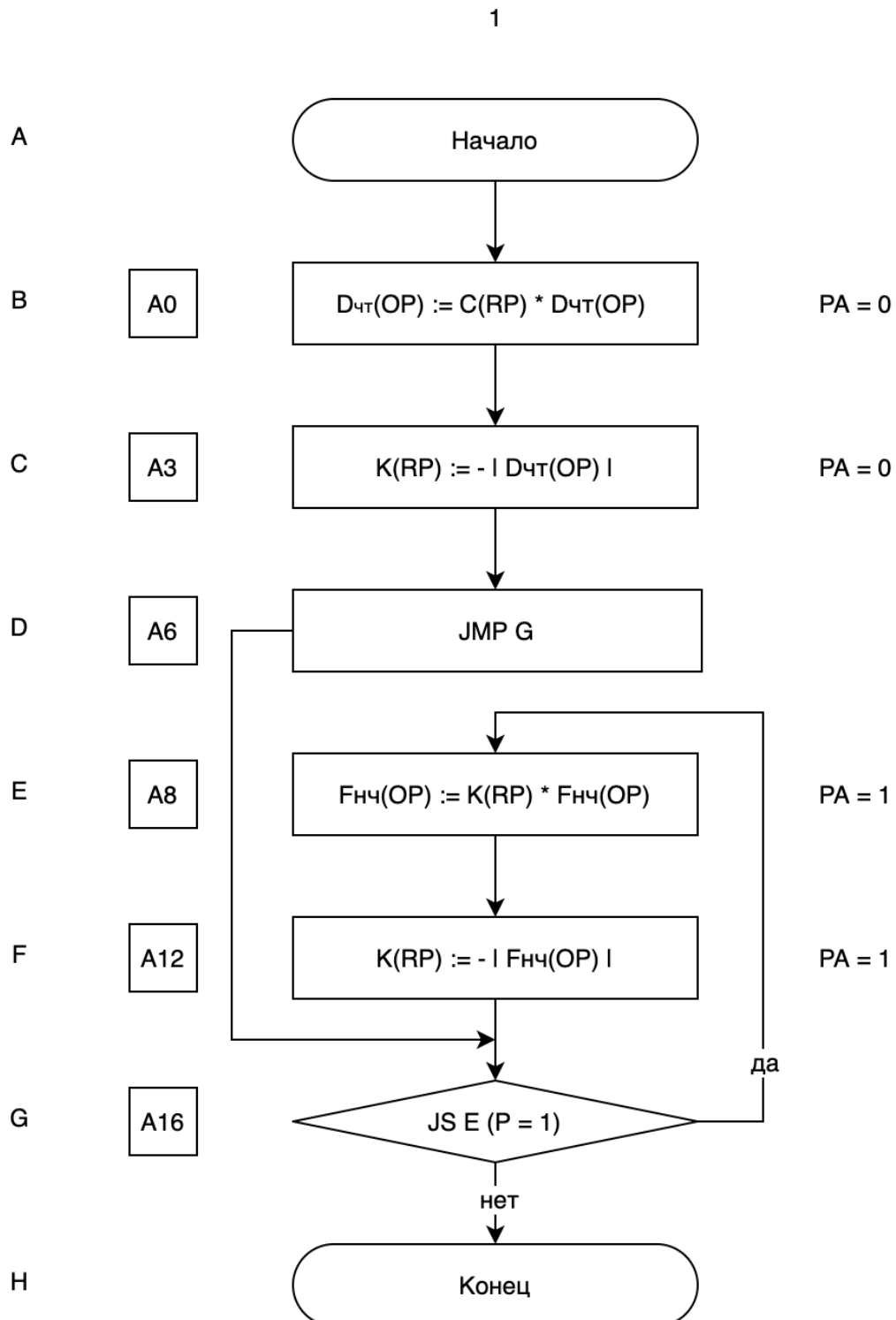


Рис. 4.10 – блок-схема алгоритма тестовой микропрограммы выполнения команд

JMP <address> - безусловный переход, <address> - смещение СК.  
 JS <address> - переход, если  $P = 1$ , где  $P$  – признак отрицательного результата пересылки (знак числа, 0 при 0, 1 при отр.), <address> - смещение СК.

Представленный на рис. 4.10 алгоритм обеспечивает:

- Проверку всех микрокоманд алгоритма
- Проверку считывания команд разного формата, расположенных по четным и нечетным адресам в оперативной памяти
- Проверку считывания операндов из оперативной памяти, расположенных по четным и нечетным адресам
- Проверку записи результата в оперативную память по четным и нечетным адресам
- Проверку считывания операндов из регистровой памяти
- Проверку записи операндов в регистровую память
- Проверку всех используемых способов адресации
- Проверку правильности вычисления признака перехода

#### **4.6.1 Пример выполнения алгоритма тестирования**

Для проверки правильности выполнения микрокоманд алгоритма тестирования данный алгоритм был выполнен вручную. Результат выполнения продемонстрирован на рис. 4.11.



Начальные значения

$C(R0) = 0.100$  (4/8)  
 $Dчт(A28) = 0.100$  (4/8)  
 $Фнч(A29) = 1.111$  (-1/8)  
 $K(R0) = 0.101$  (5/8)

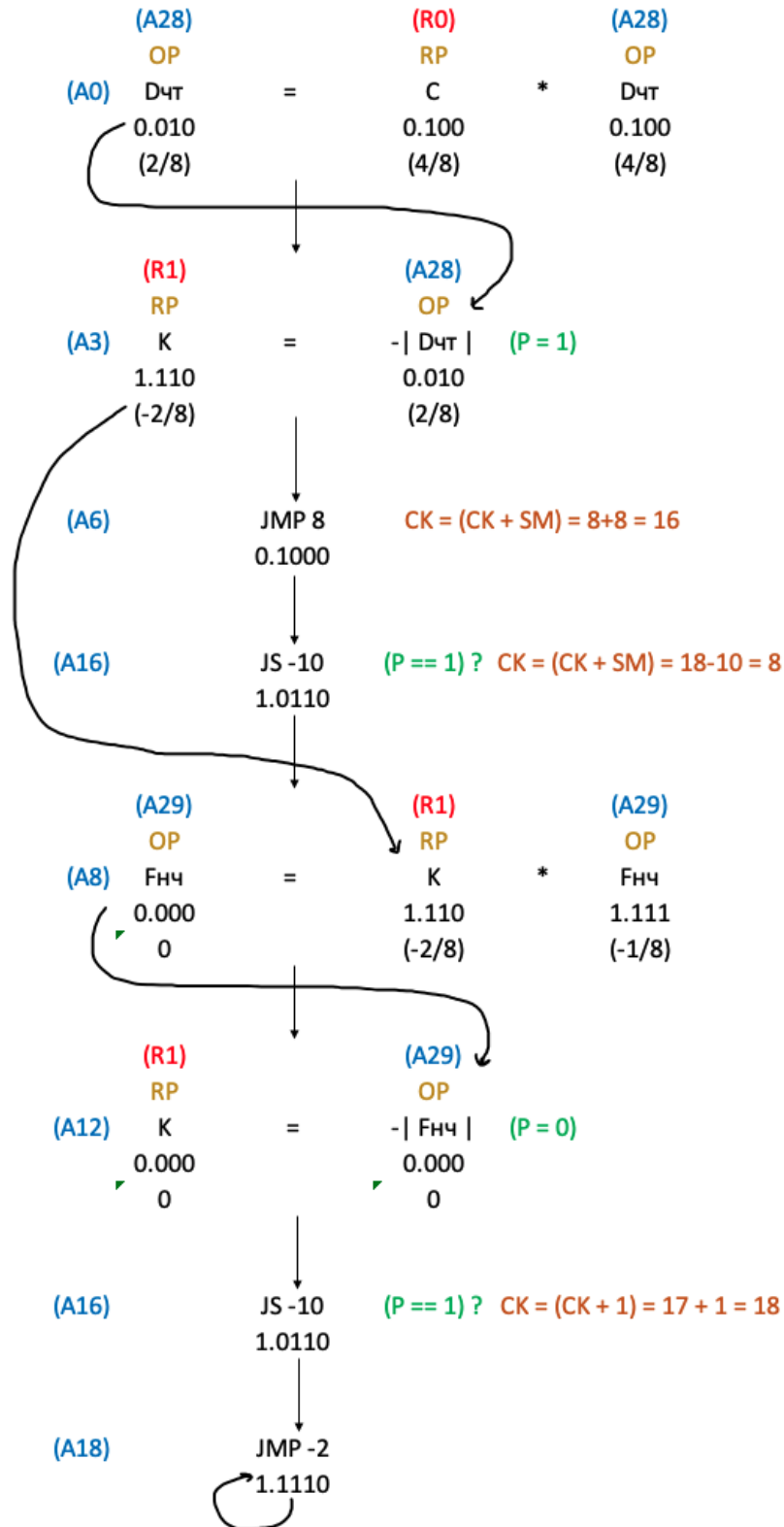


Рис. 4.11 – Пример выполнения алгоритма тестирования

#### 4.6.2 Таблицы загрузки оперативной и регистровой памяти

Для размещения команд и операндов в памяти были составлены карты загрузки оперативной (Таблица 4.6) и регистровой (Таблица 4.7) памяти.

Таблица 4.6 – загрузка оперативной памяти для выполнения программы тестирования

Адрес ячейки	Адрес слова	Команды и данные	Двоичный код		HEX	
0	A0; A1	MUL, R0, 0, PA2, A28	0000	0001	0	1
1	A2; A3	A28, SND, R1	1100	0100	C	4
2	A4; A5	R1, 0, PA2, A28	1001	1100	9	C
3	A6; A7	JMP, 0, 8	1100	1000	C	8
4	A8; A9	MUL, R1, 0, PA2, OF	0000	1010	0	A
5	A10; A11	OF, 0, R2	1010	0010	A	2
6	A12; A13	SND, R1, 0, PA2, OF	0100	1010	4	A
7	A14; A15	OF, 0, R2	1010	0010	A	2
8	A16; A17	JS, 0, -10	1001	0110	9	6
9	A18; A19	JMP, 0, -2	1101	1110	D	E
10	A20; A21	0000, 0000	0000	0000	0	0
11	A22; A23	0000, 0000	0000	0000	0	0
12	A24; A25	0000, 0000	0000	0000	0	0
13	A26; A27	000, 19	0001	0011	1	3
14	A28; A29	[D] = +4/8, [F] = -1/8	0100	1111	4	F
15	A30; A31	0000, 0000	0000	0000	0	0

Таблица 4.7 – загрузка регистровой памяти для выполнения алгоритма тестирования

Адрес ячейки	Адрес слова	Команды и данные	Двоичный код	HEX
0	R0	[C] = +4/8	0100	4
1	R1	[K] = +5/8	0101	5
2	R2	A13	1101	D
3	R3	0000	0000	0
4	R4	0000	0000	0
5	R5	0000	0000	0
6	R6	0000	0000	0
7	R7	0000	0000	0

## 4.7 Распечатки результатов моделирования блока управления командами

Результаты функционального моделирования блока управления командами в системе Xilinx представлены на рис. 4.12 - 4.15.

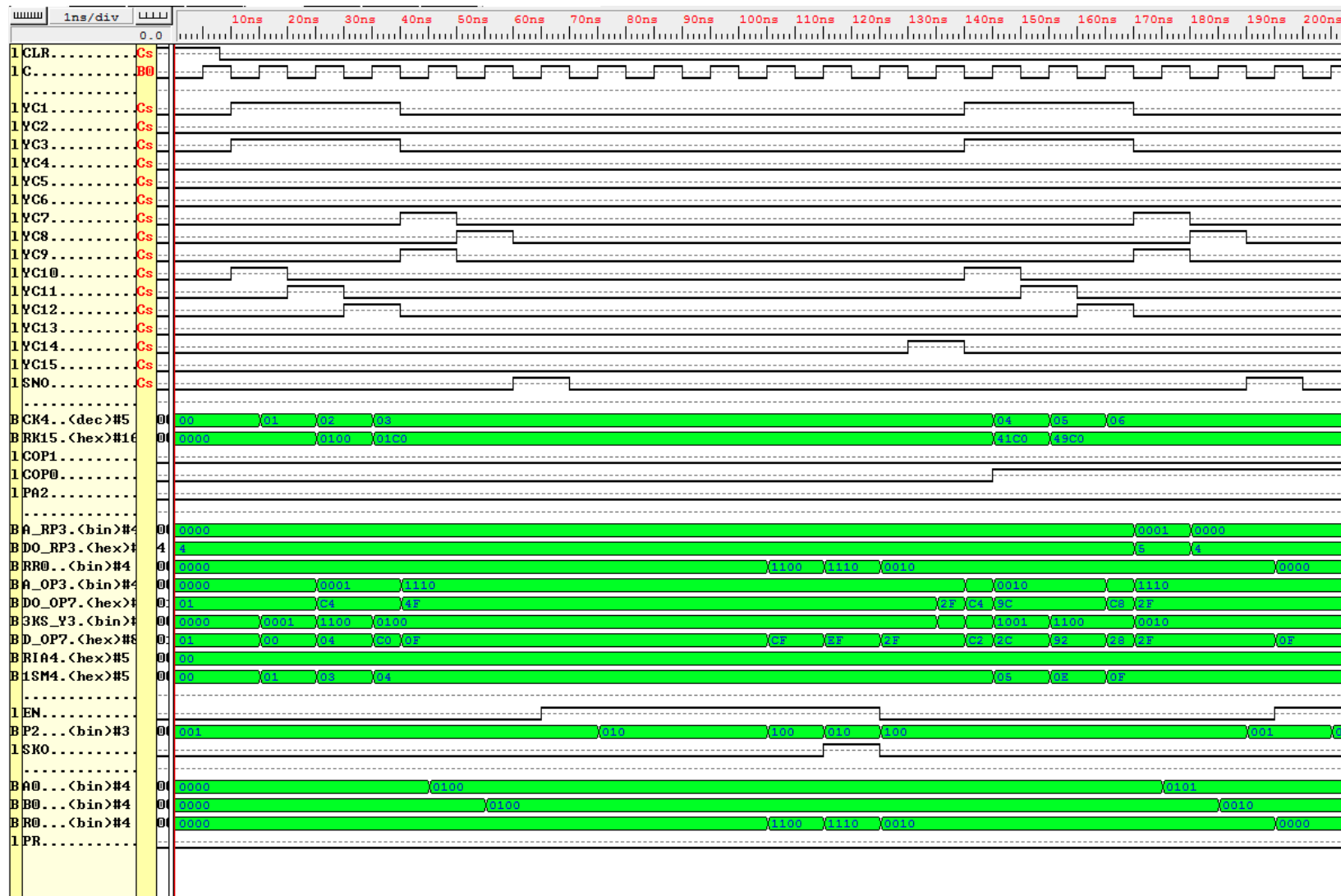


Рис. 4.12 – Лист функционального моделирования БУК №1

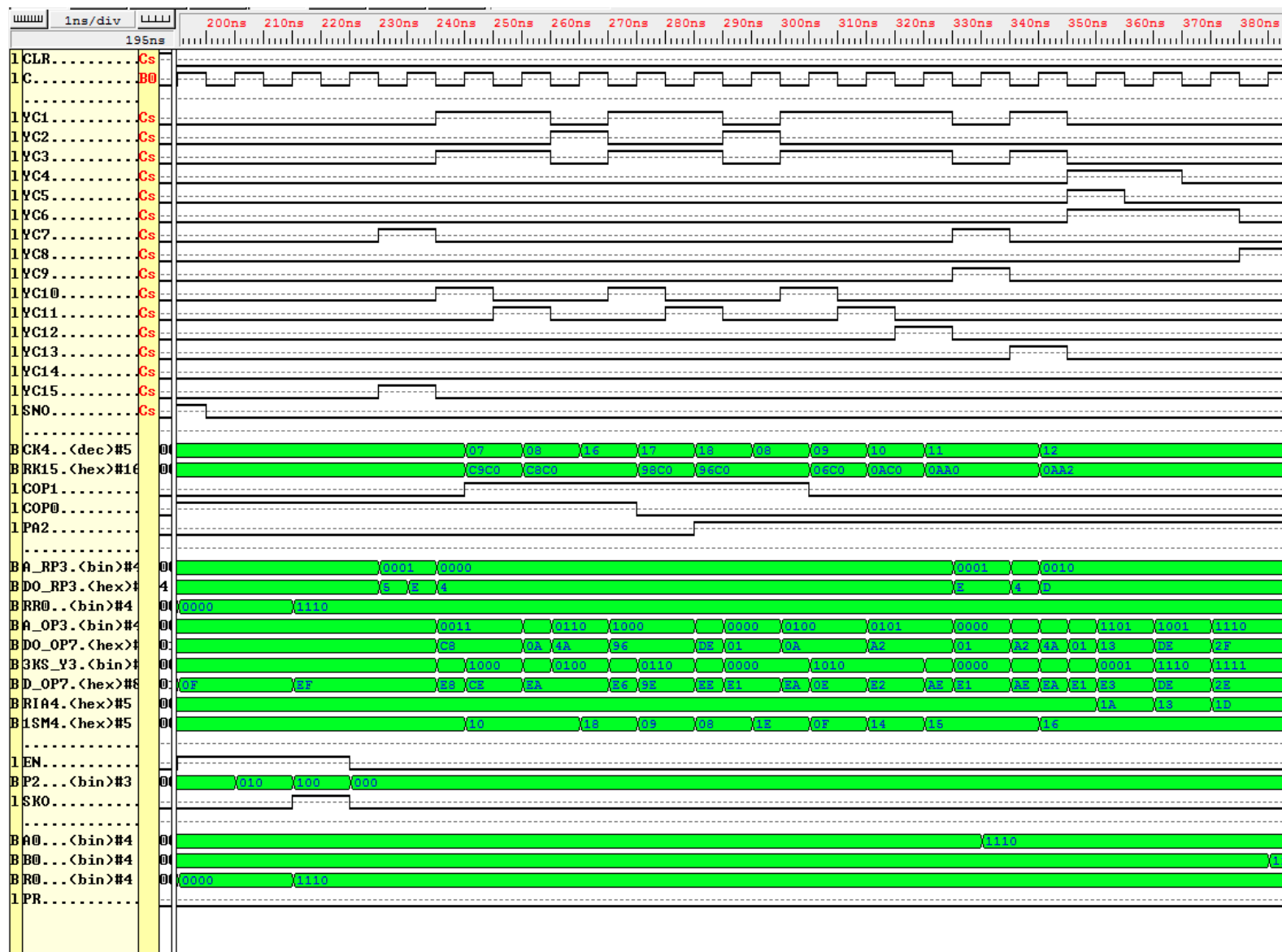


Рис. 4.13 – Лист функционального моделирования БУК №2

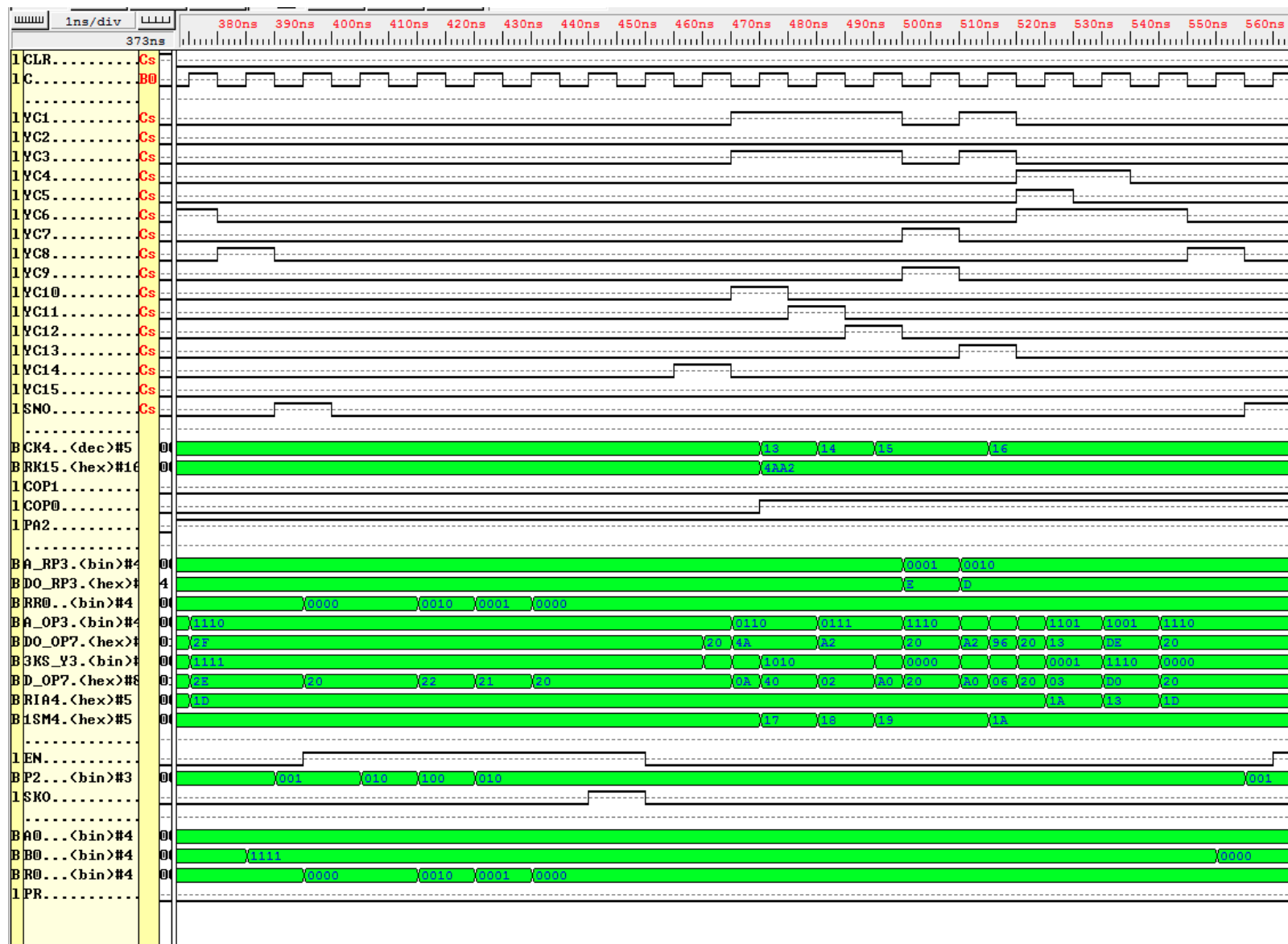


Рис. 4.14 – Лист функционального моделирования БУК №3

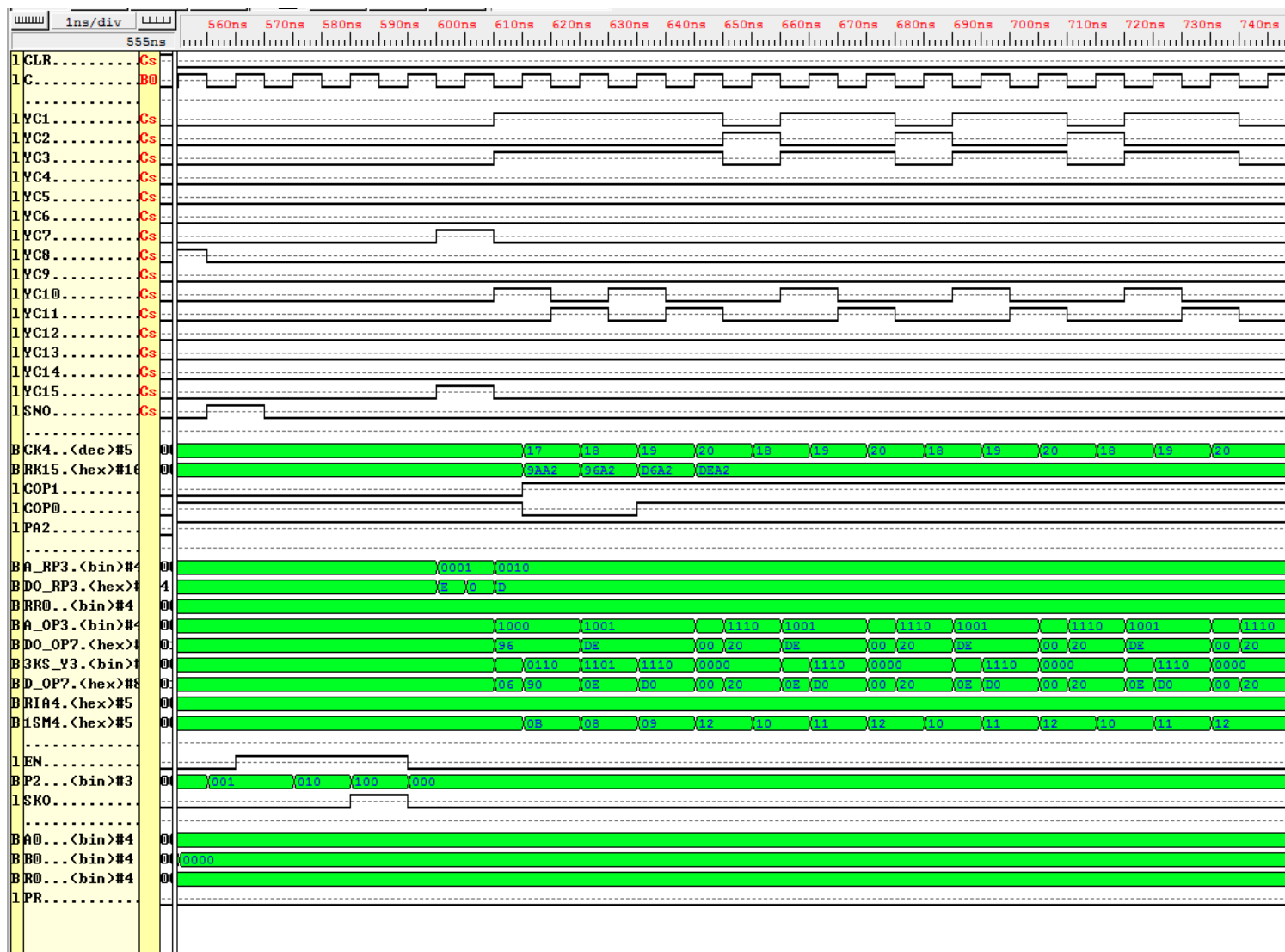


Рис. 4.15 – Лист функционального моделирования БУК №4

## 5 Проектирование блока выработки микрокоманд

В качестве способа адресации в блоке выработки микрокоманд была использована принудительная адресация. В случае, если все проверяемые признаки совпадают с текущими сигналами-признаками, в регистр адреса микрокоманд записывается адрес, идущий за адресом перехода. Иначе записывается адрес перехода. Для реализации такой адресации адрес перехода должен быть четным в случае ветвления. Если ветвление отсутствует адрес перехода может быть любым.

Способы выбора следующей микрокоманды представлены на рис. 5.1-2

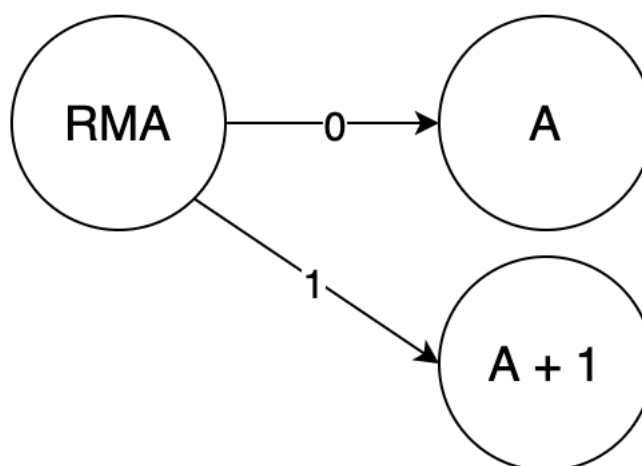


Рис. 5.1 – Выбор адреса следующей микрокоманды при ветвлении

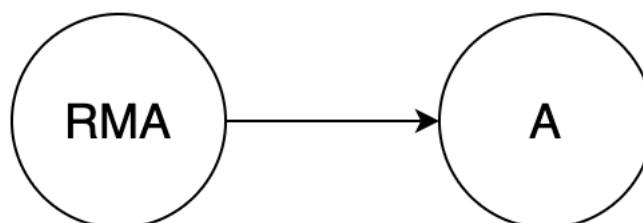


Рис. 5.2 – Выбор адреса следующей микрокоманды при отсутствии ветвления

### 5.1 Схема алгоритма микропрограммы выполнения команд

Для составления карты загрузки управляющей памяти микропрограммного устройства управления алгоритм микропрограммы выполнения команд был дополнен адресами перехода и текущими адресами микрокоманд в УП. Данный алгоритм изображен на рис. 5.3.

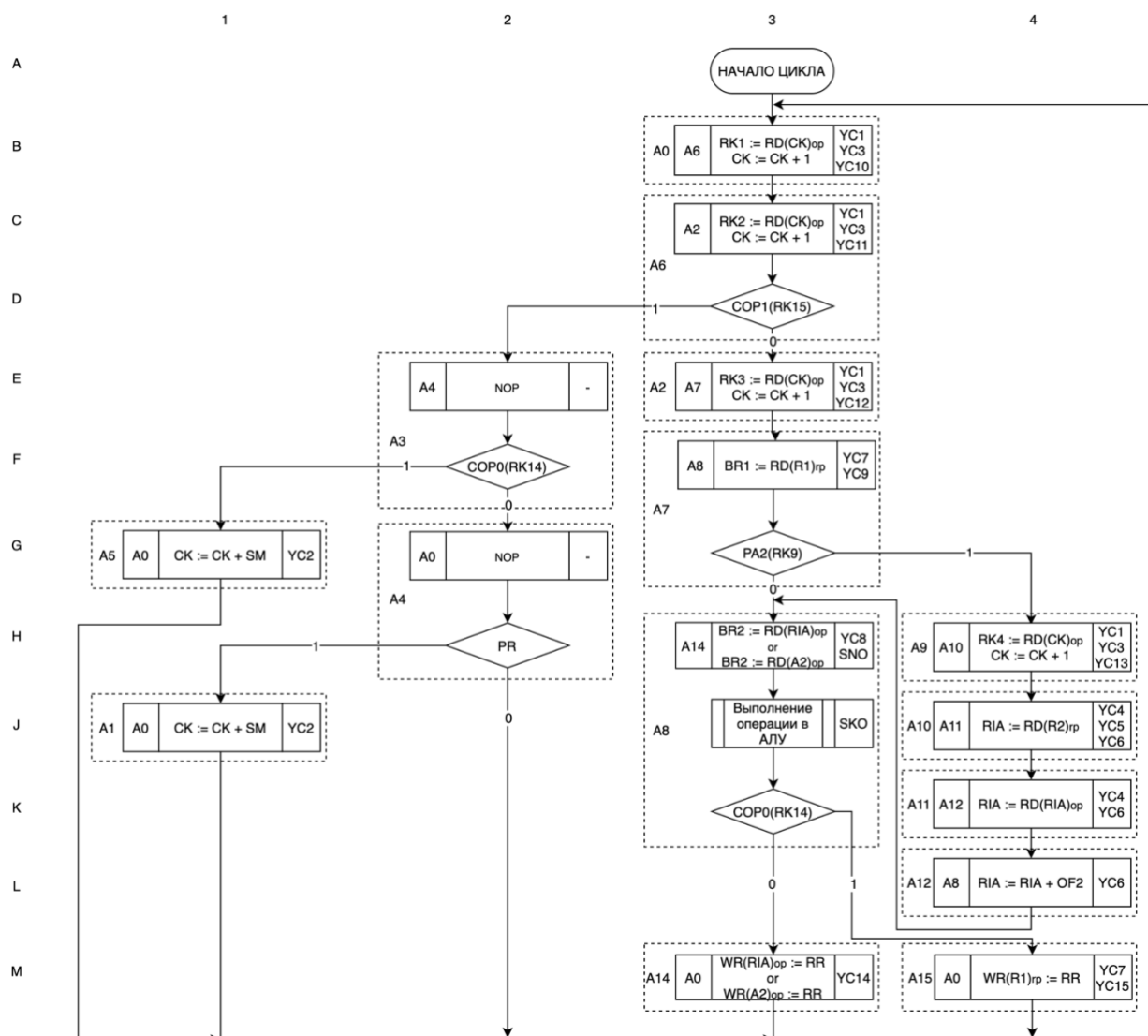


Рис. 5.3 – Блок-схема алгоритма микропрограммы выполнения команд.

## 5.2 Формат микрокоманды

В соответствии с выбранным способом адресации микрокоманд был разработан формат микрокоманды.

Для реализации принудительной адресации в микрокоманде кроме управляющих сигналов так же указаны адрес перехода и маска проверяемых признаков. В случае наличия ветвления в микрокоманде, маска признаков ненулевая. Если после применения маски к входным признакам результат оказывается ненулевой выполняется переход по адресу, следующему за адресом перехода, иначе выполняется переход по указанному адресу (аналогично микрокоманде без ветвления).

Разработанный формат микрокоманды приведен в таблице 5.1.

23	22	21	20	19	18	17	16	15	14	13	12
A3	A2	A1	A0	COP1	COP0	PA2	PR	YC15	YC14	YC13	YC12

11	10	9	8	7	6	5	4	3	2	1	0
YC11	YC10	YC9	YC8	YC7	YC6	YC5	YC4	YC3	YC2	YC1	SNO

Таблица 5.1 – Формат микрокоманды



### 5.3 Функциональная схема микропрограммного устройства управления

Функциональная схема микропрограммного устройства управления, вырабатывающего в соответствии с выбранным алгоритмом (рис. 5.3) управляющие сигналы, представлена на рис. 5.4

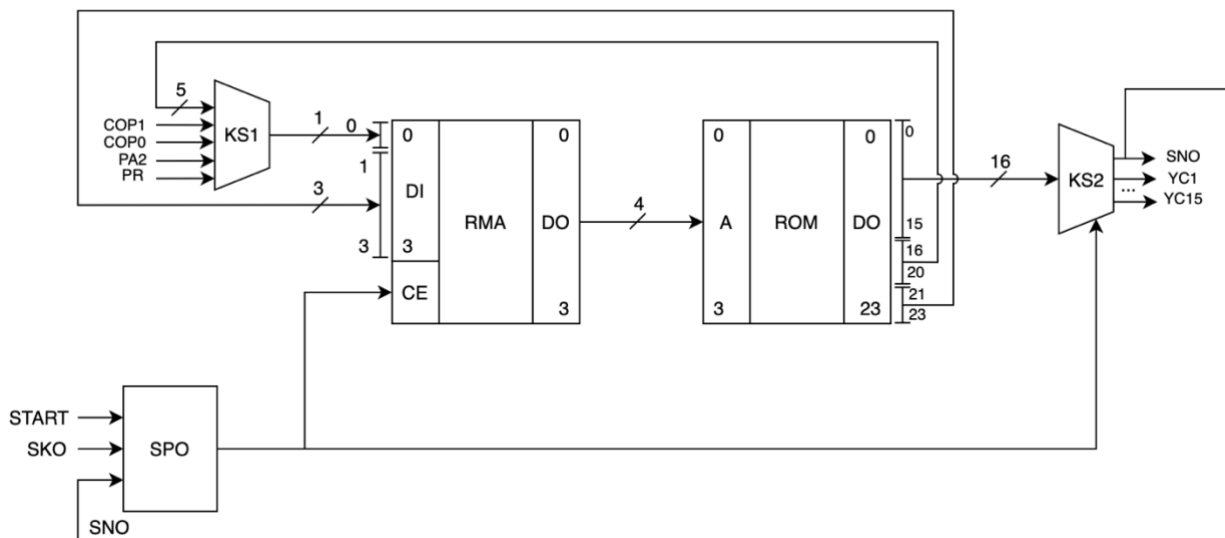


Рис. 5.4 – Функциональная схема микропрограммного устройства управления

Назначение комбинационных схем указано в таблице 5.2.

Таблица 5.2 – Комбинационные схемы БМК

Название	Назначение
KS1	Формирование нулевого разряда адреса перехода на основе сравнения маски и признаков
KS2	Стробирование управляющих сигналов блока управления командами с использованием сигнала разрешения работы микропрограммного устройства управления

<b>5.4</b>	<b>Логическая схема микропрограммного устройства управления</b>
NONE	
<b>5.4.1</b>	<b>Схема пуска останова микропрограммного устройства управления</b>
NONE	
<b>5.4.2</b>	<b>Комбинационные схемы микропрограммного устройства управления</b>
NONE	
<b>5.4.3</b>	<b>Полная логическая схема</b>
NONE	

## 5.5 Карта загрузки управляющей памяти

На основе алгоритма микропрограммы выполнения команд (рис. 5.3) и формата микрокоманды (таблица 5.1) была составлена карта загрузки управляющей памяти микропрограммного устройства управления (Таблица 5.3).

Адрес	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	HEX	Пояснение
	A3	A2	A1	A0	COP1	COP0	PA2	PR	YC15	YC14	YC13	YC12	YC11	YC10	YC9	YC8	YC7	YC6	YC5	YC4	YC3	YC2	YC1	SNO		
0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	60040A	Извлечение 15-12 разрядов команды и СК := СК + 1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	000004	Запись адреса перехода в СК и возврат в начало цикла
2	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	70100A	Извлечение 7-4 разрядов команды и СК := СК + 1
3	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	440000	Проверка кода операции (COP0)
4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	010000	Проверка признака результата (PR)
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	000004	Запись адреса перехода в СК и возврат в начало цикла
6	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	28080A	Извлечение 11-8 разрядов команды, СК := СК + 1 и проверка кода операции (COP1)
7	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	820280	Запись первого операнда из RP и проверка признака адресации
8	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	E00101	Запись второго операнда из OP, начало операции в АЛУ и проверка кода операции (COP0)
9	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	A0200A	Извлечение 3-0 разрядов команды и СК := СК + 1
10	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	B00070	Извлечение адреса базы исполнительного адреса
11	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	C00050	Извлечение базы исполнительного адреса
12	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	800040	Вычисление исполнительного адреса
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	000000	-
14	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	004000	Запись результата в OP и возврат в начало цикла
15	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	008080	Запись результата в RP и возврат в начало цикла

Таблица 5.3 – Карта загрузки управляющей памяти микропрограммного устройства управления