

УДК 004.31:378.147.88

## МИКРОПРОГРАММНАЯ РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ УЧЕБНОГО ПРОЦЕССОРА В MICROSOFT EXCEL

Страбыкин Д.А.

ФГБОУ ВО «Вятский государственный университет», Киров, e-mail: Strabykin@mail.ru

В работе предлагается подход к микропрограммной реализации и экспериментальному исследованию учебного процессора ЭВМ, основанный на применении ранее разработанной функциональной модели базового процессора с микропрограммируемой архитектурой в Microsoft Excel. Приводятся структура и возможности функциональной модели базового процессора. Структура базового процессора с блоком памяти программ и данных отображается на экранной форме, используемой для отладки микропрограмм и программ при проведении экспериментальных исследований и позволяющей наблюдать изменения состояний внутренних регистров процессора. На примере простейшей вычислительной задачи (деление чисел нацело) рассматривается решение задач, связанных с микропрограммной реализацией и экспериментальным исследованием учебного процессора: разработка (выбор) алгоритма решения вычислительной задачи; определение архитектуры процессора и программирование; кодирование программы и распределение памяти программ и данных; разработка микропрограммы командного цикла и составление таблиц прошивок блока памяти микропрограмм и преобразователя кода операции в начальный адрес соответствующей микропрограммы; ввод и отладка микропрограммы командного цикла и программы решения вычислительной задачи. Отмечаются дополнительные возможности функциональной модели базового процессора, упрощающие и ускоряющие отладку микропрограмм и программ: наличие на экранных формах специальных полей, в которых отображаются мнемонические обозначения и пояснения, облегчающие понимание изменений состояний процессора при проведении экспериментальных исследований; использование пультовых дублеров регистров и ячеек памяти, позволяющих заносить данные в регистры и ячейки памяти перед выполнением любой микрокоманды (команды) с использованием клавиатуры; реализация различных режимов работы учебного процессора, обеспечивающих выполнение отдельных микрокоманд, команд и всей программы. Рассмотренный подход способствует более глубокому изучению архитектуры, структуры и рабочего цикла процессора, позволяя обучающемуся разработать простейший учебный процессор, создать и экспериментально исследовать его действующую модель.

**Ключевые слова:** микропрограммируемые вычислительные структуры, архитектура и структура процессора, действующие функциональные модели, разработка и отладка микропрограмм, компьютерные практикумы по ЭВМ, применение Microsoft Excel

## MICROPROGRAM IMPLEMENTATION AND EXPERIMENTAL TEST OF THE TRAINING MICROPROCESSOR IN MICROSOFT EXCEL

Strabykin D.A.

Federal State Budgetary Educational Institution of Higher Education «Vyatka State University»,  
Kirov, e-mail: Strabykin@mail.ru

This paper suggests a microprogram implementation and experimental tests of the training computer processor based on a previously developed functional model of a basic microprogrammable processor architecture in Microsoft Excel. The structure and capabilities of the functional model of the basic processor are described. The structure of the basic processor with a program and data memory blocks is displayed as a screen form and used to debug microprograms and programs during experimental tests. The screen form allows observing the changes in the states of processor's internal registers. Integer division is used as an example of a simple computational task to demonstrate how a set of problems related to microprogram implementation and experimental test of a training processor is solved. This set of problems includes: development (selection) of an algorithm to solve the computational task, deciding of the processor architecture and programming, coding the program and distributing the memory between programs and data, development of the command cycle microprogram and creation of tables for the microprogram memory block and the converter of operation codes to the initial address of the corresponding microprograms, input and debugging of the command cycle microprogram and the program solving the computational task. The functional model of the basic processor has the following additional features to simplify and accelerate microprogram and program debugging: special fields in the screen form to display mnemonic designations and explanations, which help in understanding processor's state changes during experimental tests; the use of input panels to control register and memory cells, which allow data to be entered with keyboard into registries and memory cells before execution of any microcommand (command); the implementation of various modes of operation of the training processor allowing execution of individual microcommands, commands, and the entire program. The suggested approach helps deeper studies of processor's architecture, structure and operating cycle by allowing the students to design the simplest training processor, implement and experimentally test its functional model.

**Keywords:** microprogrammable computing devices, processor architecture, structure of processors, functional models, development and debugging of microprograms, practical studies of computers, Microsoft Excel applications

Важную роль при изучении процессоров ЭВМ играют практические занятия и лабораторные работы, позволяющие обучающимся получить более глубокие знания.

Для проведения лабораторных работ часто используются программные модели реальных [1, 2] или учебных [3, 4] процессоров. Программные модели могут быть кросс-

сплатформенными, а также представлять собой веб-приложения. В программных моделях реальных процессоров обучаемым доступна архитектура процессора на уровне системы команд, поэтому экспериментальные исследования в основном связаны с разработкой и покомандным выполнением ассемблерных программ. Программные модели учебных процессоров могут быть ориентированы на микропрограммируемые процессоры и позволять обучающимся проводить самостоятельную разработку микропрограмм интерпретаторов команд и экспериментальное исследование процессоров с различной архитектурой [4, 5]. Такие модели дают возможность исследовать работу процессора не только на уровне архитектуры, но и на уровне «микроархитектуры», отражающей структуру процессора и демонстрирующей потактовое выполнение команд.

Цель исследования: развитие технологии применения функциональных моделей микропрограммируемых вычислительных структур в учебном процессе, использующей в качестве средства моделирования табличный процессор Microsoft Excel и позволяющей осуществлять микропрограммную реализацию и экспериментальное исследование учебного процессора ЭВМ.

Предлагается подход к микропрограммной реализации и экспериментальному исследованию учебного процессора ЭВМ, основанный на применении функциональной модели процессора с микропрограммируемой архитектурой в Microsoft Excel [5]. Моделируемый процессор с микропрограммируемой архитектурой называется далее базовым процессором (БПР), в отличие от учебного процессора (УПР), микропрограммная реализация которого осуществляется с помощью БПР.

*Структура и возможности  
функциональной модели базового  
процессора*

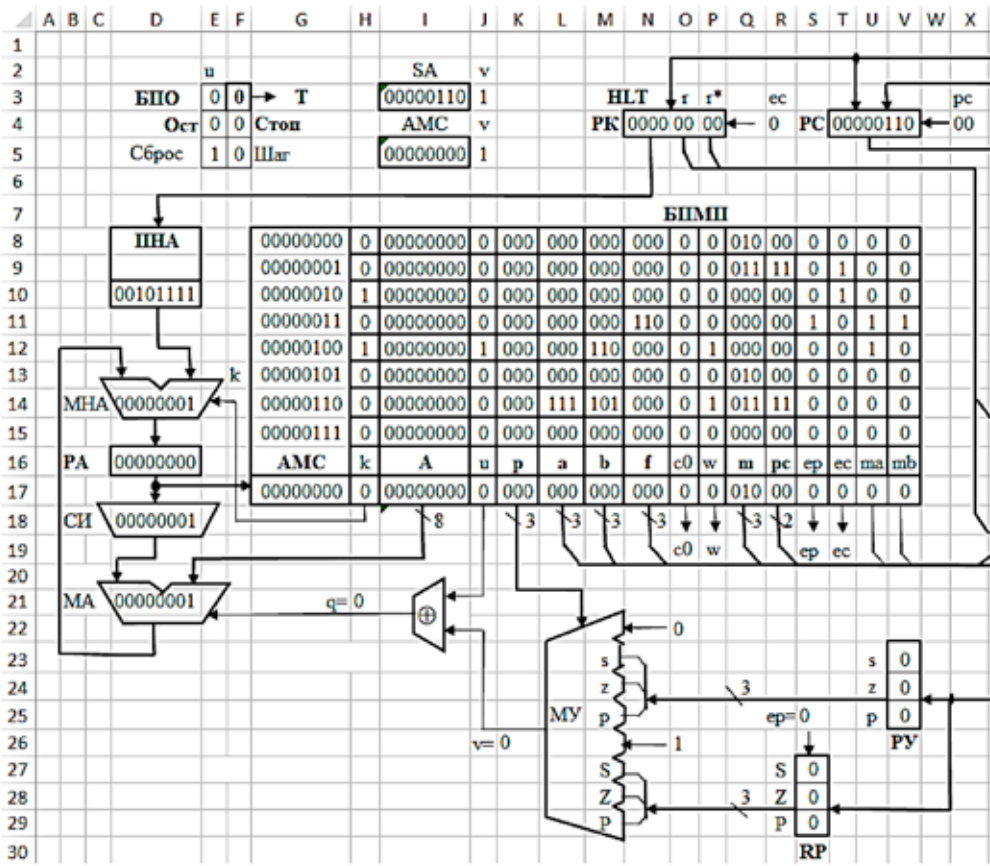
Используемая функциональная модель, кроме модели восьмиразрядного микропрограммируемого БПР, содержит модель оперативного запоминающего устройства. Структура БПР с запоминающим устройством отображается на экранной форме ПР-Ф при проведении экспериментальных исследований. На экранной форме ПР-Ф присутствуют следующие основные блоки (рис. 1) [5]: БПО – блок пуска-останова; БППД – блок памяти программ и данных; ПНА – преобразователь начального адреса, БПМП – блок памяти микропрограмм; БР – блок регистров; БАЛ – блок арифметико-логический.

Кроме того, в состав базового процессора входят (рис. 1): РК – регистр команд; РС – программный счетчик; РА – регистр адреса БППД, МНА – мультиплексор начального адреса, РА – регистр адреса, СИ – схема инкремента, МА – мультиплексор адреса, «⊕» – инвертор кода условия, МУ – мультиплексор условий и признаков, РУ – регистр условий, РР – регистр признаков, МД – мультиплексор данных, Мс, Мд – мультиплексоры номеров регистров БР; РВ – буферный регистр. На экранной форме ПР-Ф, используемой для проведения экспериментальных исследований (рис. 1), также отображаются: SA – пультовый регистр ввода в РС адреса команды; АМК – пультовый регистр ввода в РА – адреса микрокоманды; ДБР – пультовый дублер блока регистров.

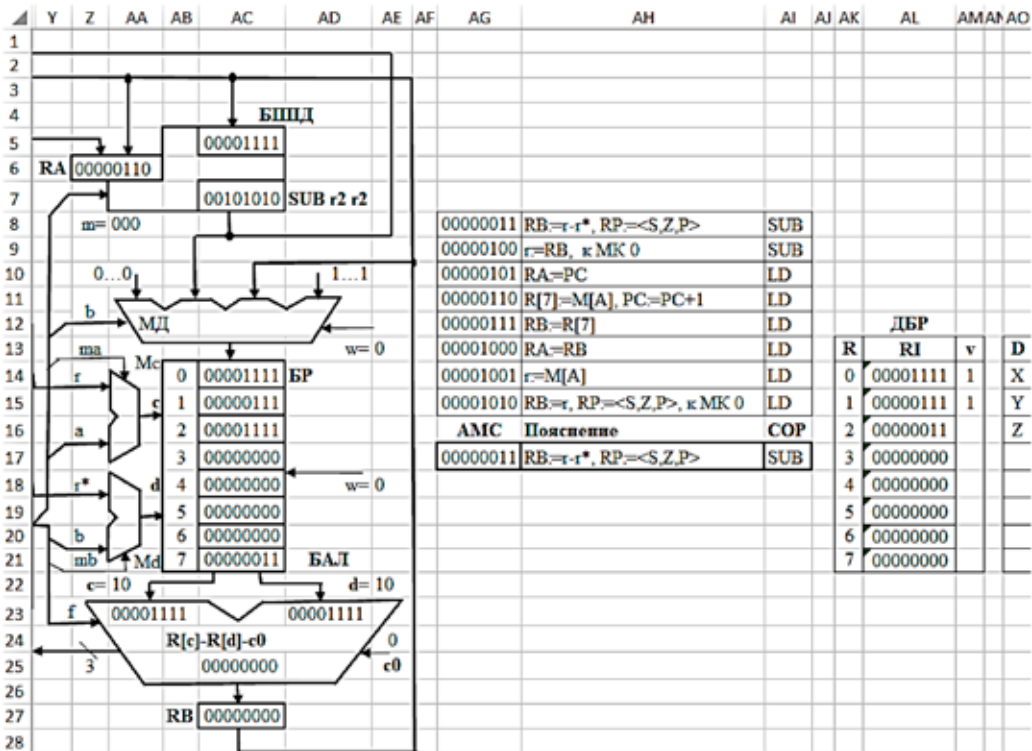
Основу операционной части базового процессора составляют БР и БАЛ. БР содержит восемь восьмиразрядных регистров. Содержимое любой пары регистров с помощью двух трехразрядных входов номеров регистров может быть выбрано и передано на входы БАЛ, в котором выполняется одна из восьми микроопераций и формируются значения трех логических условий. Управляющая часть БПР представляет собой устройство управления с программируемой логикой, в памяти которого хранятся двоичные коды микрокоманд. На рисунке 2 показаны назначение и значения кодов многоразрядных полей микрокоманды (где  $V$  – наименование,  $n$  – разрядность поля, а  $M[A]$  – содержимое ячейки памяти с адресом  $A$ ).

БПМП отображается на экранной форме проведения исследований ПР-Ф в виде блока памяти, в котором выделено окно микрокоманд, содержащее восемь последовательно расположенных 33-разрядных ячеек памяти (рис. 1а). Содержимое окна обновляется после выполнения каждой микрокоманды. Занесение данных в ПНА и БПМП производится на листе ПР-В путем ввода данных с клавиатуры в ячейки таблиц, моделирующих ПНА и БПМП, а также ячейки, предусмотренные для ввода дополнительной информации (пояснения, коды операций).

Работа БППД моделируется с помощью таблицы, расположенной на листе ПР-В. Ввод данных в БППД производится на экранной форме ПР-В аналогично записи в БР с помощью ячеек пультового дублера БППД. Данные вводятся с клавиатуры в выбранные дублирующие регистры БР (ячейки БППД), а соответствующие им управляющие сигналы  $v$  ( $vm$ ) устанавливаются в единичное. При нажатии клавиши F9 данные из дублирующих регистров (ячеек), у которых  $v=1$  ( $vm=1$ ), переносятся в основные регистры БР (БППД).



a)



b)

Рис. 1. Экранная форма ПР-Ф: левая часть (а) и правая часть (б)

V	n	Назначение
k	1	Занесение в РА микрокоманды адреса из ПНА (при k=1)
A	8	Адрес следующей микрокоманды
u	1	Инверсия значения выбранного условия (признака) (при u=1)
p	3	Код выбираемого условия (признака)
a	3	Номер регистра в БР (при чтении подается на левый вход БАЛ)
b	3	Номер регистра в БР (при чтении подается на правый вход БАЛ)
f	3	Код микрооперации, выполняемой в БАЛ
c0	1	Значение входа переноса в младший разряд БАЛ
w	1	Запись в БР (при w=1)
m	3	Код микрооперации, выполняемой в БППД и РА
pc	2	Код микрооперации, выполняемой в РС
ep	1	Запись в регистр признаков RP (при ep=1)
ec	1	Запись в регистр команд РК (при ec=1)
ma	1	Выбор источника номера регистра (c=a при ma=0 и c=0r при ma=1)
mb	1	Выбор источника номера регистра (d=b при mb=0 и d=0r* при mb=1)

а)

f	Микрооперация
000	RB=R[c]
001	RB=R[c]&R[d]
010	RB=R[c]∨R[d]
011	RB=R[c]⊕R[d]
100	RB=2R[d].0
101	RB=0.R[d]/2
110	RB=R[c]-R[d]-c0
111	RB=R[c]+R[d]+c0

б)

p	Признак
000	0
001	s
010	z
011	p
100	1
101	S
110	Z
111	P

в)

w	b	БР
0	xxx	Хранение
1	100	R[c]:=00000000
1	101	R[c]:=M[A]
1	110	R[c]:=RB
1	111	R[c]:=11111111

г)

m	БППД	РА
000	Хранение	Хранение
001	Хранение	RA:=RB
010	Хранение	RA:=PC
011	Чтение	Хранение
100	Запись	Хранение

д)

pc	РС
00	Хранение
01	PC:=RB
10	PC:=M[A]
11	PC:=PC+1

е)

Рис. 2. Назначение полей микрокоманды (а) и значения кодов многоразрядных полей микрокоманды: БАЛ (б), БР (в), РС (г), БППД и РА (д), признаков (е)

Кроме того, по сигналу «Сброс» при единичном значении соответствующего управляющего сигнала (v) в РС (РА) может быть занесен предварительно набранный на клавиатуре стартовый адрес команды SA (адрес микрокоманды АМК).

Процессор функционирует под действием сигналов, вырабатываемых блоком пуска-останова БПО (рис. 1а), в состав которого входит генератор тактовых импульсов (ГТИ). В БПО с клавиатуры подаются

сигналы «Сброс» (E5) и «Шаг» (F5) (в соответствующую ячейку MS Excel вводится необходимое значение сигнала (0 или 1) и нажимается клавиша F9). В БПО вырабатываются два признака: «Ост» и «Стоп». Признак «Ост» (останов) принимает единичное значение, если выполняется микрокоманда, содержащая микрооперацию останова (p=100), и нулевое значение – по сигналу «Сброс». Признак «Стоп» принимает единичное значение, если единичное значение

имеет сигнал «Шаг» и выполняется микрокоманда, содержащая микрооперацию ( $k=1$ ) перехода по коду операции, и нулевое значение – по сигналу «Сброс».

Проведению экспериментальных исследований предшествует изменение параметров Microsoft Excel, связанных с вычислением формул (выбираются вычисления в книге «вручную» и включаются итеративные вычисления с предельным числом итераций, превышающим удвоенное максимальное число микрокоманд, которые необходимо выполнить после нажатия клавиши F9). БПО обеспечивает работу БПР в следующих режимах: выполнение микрокоманды за два полутакта; выполнение микрокоманды за один такт; выполнение команды; выполнение программы.

*Применение функциональной модели базового процессора в учебном процессе*

Функциональная модель базового процессора позволяет производить микропрограммную реализацию и экспериментальное исследование учебных процессоров с различной архитектурой. В качестве исходных данных может выступать простая задача, на примере программного решения которой исследуется работа учебного процессора. Применение функциональной модели базового процессора позволяет обучающимся решать следующие проектные задачи для учебного процессора.

1. Разработка (выбор) алгоритма решения задачи.
2. Определение архитектуры процессора и программирование.
3. Кодирование программы и распределение памяти программ и данных.
4. Разработка микропрограммы командного цикла и составление таблиц прошивок ПНА и БПМП.
5. Ввод и отладка микропрограммы командного цикла и программы решения задачи.

Рассмотрим применение функциональной модели БПР в учебном процессе

на примере операции деления нацело 8-разрядного двоичного числа  $X$  на 8-разрядное двоичное число  $Y$ .

**1. Разработка (выбор) алгоритма решения задачи.** Для выполнения заданной операции предлагается использовать следующий алгоритм. Если число  $Y=0$ , то деление не выполняется, а формируется признак переполнения  $P$ . В противном случае из делимого  $X$  вычитается делитель  $Y$ , и если разность  $R$  положительное число, то частное  $Z$  увеличивается на единицу, а разность  $R$  выступает в качестве нового уменьшаемого на следующем шаге. Процесс вычитания продолжается до тех пор, пока не будет получена отрицательная разность. В этом случае  $Z$  будет представлять собой искомое частное от деления  $X$  на  $Y$  нацело [МПВУ].

**2. Определение архитектуры УПР и программирование.** Определение архитектуры включает выбор программистской структуры (состава программно-доступных регистров) и определение внутреннего языка УПР (системы данных и системы команд). Программистская структура содержит: четыре общих регистра ( $r0-r3$ ), программный счетчик ( $PC$ ) и регистр признаков ( $RP$ ), включающий три разряда: знака ( $S$ ), нуля ( $Z$ ) и переполнения ( $P$ ) (рис. 3а).

Разработка системы команд предполагает определение набора операций, способов адресации, модификаций и форматов команд. Для рассматриваемого примера система команд УПР приведена в таблице 1, а форматы команд – на рисунке 3б. В таблице 1 использованы следующие обозначения:  $r, r^* \in \{r0, r1, r2, r3\}$  – программно-доступные регистры: регистр  $r^*$  является источником данных, а регистр  $r$  – приемником результата, но может также служить источником второго операнда;  $M[A]$  – ячейка памяти с адресом  $A$ ; знак «+» в описании признаков означает, что устанавливается новое значение признака по результату выполнения команды, а знак «-» свидетельствует о сохранении старого значения признака.

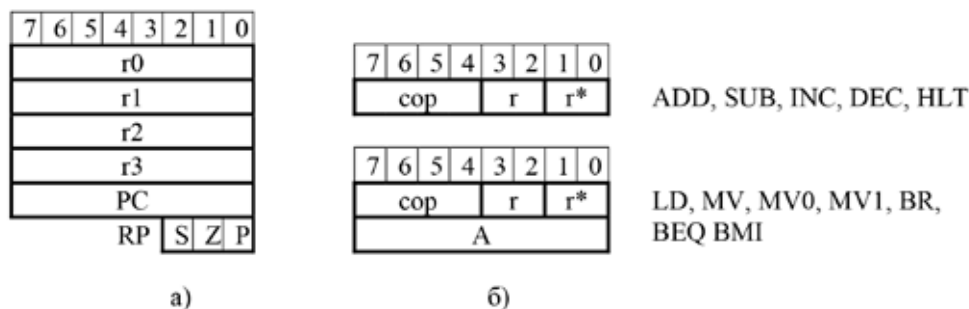


Рис. 3. Регистры программистской структуры (а) и форматы команд (б)

Таблица 1

## Система команд

Наименование	Мнемоника	Описание	Признаки		
			S	Z	P
СУММИРОВАНИЕ	ADD r r*	r:=r+r*, PC:=PC+1	+	+	+
ВЫЧИТАНИЕ	SUB r r*	r:=r-r*, PC:=PC+1	+	+	+
Инкремент	INC r	r:=r+1, PC:=PC+1	+	+	+
Декремент	DEC r	r:=r-1, PC:=PC+1	+	+	+
ЧТЕНИЕ В РЕГИСТР	LD r A	r:=M[A], PC:=PC+2	+	+	+
ЗАПИСЬ ИЗ РЕГИСТРА	MV r A	M[A]:=r, PC:=PC+2	+	+	+
Запись НУЛЕЙ	MV0 A	M[A]:=00000000, PC:=PC+2	0	1	0
Запись ЕДИНИЦ	MV1 A	M[A]:=11111111, PC:=PC+2	1	0	0
ПЕРЕХОД	BR A	PC:=A	-	-	
ПЕРЕХОД, ЕСЛИ НУЛЬ	BEQ A	Если PZ=1, то PC:=A, иначе PC:=PC+2	-	-	
ПЕРЕХОД, ЕСЛИ МИНУС	BMI A	Если PS=1, то PC:=A, иначе PC:=PC+2	-	-	-
ОСТАНОВ	HLT	Останов	-	-	-

SUB r2 r2	Очистка регистра r2 частного Z
LD r1 AY	Загрузка в r1 делителя Y
BEQ m1	Если признак Z=1 (Y=0), то переход на метку m1
LD r0 AX	Загрузка в r0 делимого X
m3 SUB r0 r1	Вычитание из делимого X делителя Y
BMI m2	Если признак S=1, то переход на метку m2
INC r2	Увеличение на единицу частного Z
BR m3	Переход на метку m3
m1 MV1 AP	Запись единиц в ячейку признака P
BR m4	Переход на метку m4
m2 MV0 AP	Запись нулей в ячейку признака P
MV r2 AZ	Запись частного из регистра r2 в ячейку частного Z
m4 HLT	Останов

Рис. 4. Программа деления чисел нацело

Программа деления чисел нацело с использованием символики ассемблера, составленная в предположении, что исходные данные и результаты размещаются в блоке памяти программ и данных, приведена на рисунке 4.

**3. Кодирование программы и распределение памяти программ и данных.** Двоичные коды операций и начальные адреса их микропрограмм в блоке памяти микропрограмм приведены в таблице 2.

Ввод данных и команд в БППД производится с помощью экранной формы ввода ПР-В. Двоичные коды данных и команд для рассматриваемого примера занимают 28 ячеек памяти. Распределение памяти программ и данных для первых 23 ячеек показано на рисунке 5а.

Таблица 2

## Коды операции и начальные адреса микропрограмм операций

Мнемоника	Код операции	Начальный адрес МП
HLT	0000	00101111
ADD	0001	00101111
SUB	0010	00000011
INC	0011	00001110
DEC	0100	00010100
LD	1000	00000101
MV	1001	00100111
BR	1010	00010111
BEQ	1011	00001011
BMI	1100	00010001
MV1	1101	00011001
MV0	1110	00100000

Z	AA	AB	AC	AD	AE	AF	AG	AH		
1										
2		<b>БПД</b>								
3										
N	АЯП	D	DI	vm	Мнемоника	Комментарий				
5	0	00000000	00000110	00000110	1	SA	Начальный адрес программы			
6	1	00000001	00000111	00000111	1	X	Делимое			
7	2	00000010	00000111	00000111	1	Y	Делитель			
8	3	00000011	00010000	00000000		Z	Частное			
9	4	00000100	00000000	01010101		P	Признак переполнения			
10	5	00000101	00000000	00000000	1		Свободная ячейка памяти (ЯП)			
11	6	00000110	00101010	00101010	1	SUB r2 r2	Очистка регистра r2 частного Z			
12	7	00000111	10000100	10000100	1	LD r1	Загрузка в r1 делителя Y из ЯП с адресом AY			
13	8	00001000	00000010	00000010	1	AY				
14	9	00001001	10110000	10110000	1	BEQ m1	Если PZ=1 (Y=0), то переход на метку m1			
15	10	00001010	00010011	00010011	1	Am1				
16	11	00001011	10000000	10000000	1	LD r0	Загрузка в r0 делимого X из ЯП с адресом AX			
17	12	00001100	00000001	00000001	1	AX				
18	13	00001101	00100001	00100001	1	SUB r0 r1	Вычитание из делимого X делителя Y			
19	14	00001110	11000000	11000000	1	BMI m2	Если PS=1, то переход на метку m2			
20	15	00001111	00010111	00010111	1	Am2				
21	16	00010000	00111000	00111000	1	INC r0	Увеличение на единицу частного Z			
22	17	00010001	10100000	10100000	1	BRm3	Переход на метку m3			
23	18	00010010	00001101	00001101	1	Am3				
24	19	00010011	11010000	11010000	1	MV1	Запись единиц в ЯП признака P			
25	20	00010100	00000100	00000100	1	AP				
26	21	00010101	10100000	10100000	1	BRm4	Переход на метку m4			
27	22	00010110	00011011	00011011	1	Am4				
28	23	00010111	11100000	11100000	1	MV0	Запись нулей в ЯП признака P			

а)

F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	
1																			
2			<b>БМП</b>																
3			3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
N	AMC	k	A	u	p	a	b	f	c0	w	m	pc	ep	ec	ma	mb	Пояснение		
5	0	00000000	0	00000000	0	000	000	000	000	0	0	010	00	0	0	0	0	RA=PC	COP
6	1	00000001	0	00000000	0	000	000	000	000	0	0	011	11	0	1	0	0	PC=M[A], PC=PC+1	RC
7	2	00000010	1	00000000	0	000	000	000	000	0	0	000	00	0	0	0	0	Переход по КОП	RC
8	3	00000011	0	00000000	0	000	000	000	110	0	0	000	00	1	0	1	1	RB=r-r*, RP=<S,Z,P>	SUB
9	4	00000100	0	00000000	1	000	000	110	000	0	1	000	00	0	0	1	0	r=RB, к МК 0	SUB
10	5	00000101	0	00000000	0	000	000	000	000	0	0	010	00	0	0	0	0	RA=PC	LD
11	6	00000110	0	00000000	0	000	111	101	000	0	1	011	11	0	0	0	0	R[7]=M[A], PC=PC+1	LD
12	7	00000111	0	00000000	0	000	111	000	000	0	0	000	00	0	0	0	0	RB=R[7]	LD
13	8	00001000	0	00000000	0	000	000	000	000	0	0	001	00	0	0	0	0	RA=RB	LD
14	9	00001001	0	00000000	0	000	000	101	000	0	1	011	00	0	0	1	0	r=M[A]	LD
15	10	00001010	0	00000000	1	000	000	000	000	0	0	000	00	1	0	1	0	RB=r, RP=<S,Z,P>, к МК 0	LD
16	11	00001011	0	00001101	0	110	000	000	000	0	0	010	00	0	0	0	0	RA=PC, при Z=1, к МК 1101	BEQ
17	12	00001100	0	00000000	1	000	000	000	000	0	0	000	11	0	0	0	0	PC=PC+1, к МК 0	BEQ
18	13	00001101	0	00000000	1	000	000	000	000	0	0	011	10	0	0	0	0	PC=M[A], к МК 0	BEQ
19	14	00001110	0	00000000	0	000	111	100	000	0	1	000	00	0	0	0	0	R[7]=0	INC
20	15	00001111	0	00000000	0	000	000	111	111	1	0	000	00	1	0	1	0	RB=r+R[7]+1, RP=<S,Z,P>	INC
21	16	00010000	0	00000000	1	000	000	110	000	0	1	000	00	0	0	1	0	r=RB, к МК 0	INC
22	17	00010001	0	00010011	0	101	000	000	000	0	0	010	00	0	0	0	0	RA=PC, при S=1 к МК 10011	BMI
23	18	00010010	0	00000000	1	000	000	000	000	0	0	000	11	0	0	0	0	PC=PC+1, к МК 0	BMI
24	19	00010011	0	00000000	1	000	000	000	000	0	0	000	10	0	0	0	0	PC=M[A], к МК 0	BMI

б)

Рис. 5. Экранная форма ввода ПР-В: область ввода данных и программ в БПД (а), область ввода микропрограмм в БМП (б)

**4. Разработка микропрограммы командного цикла и составление таблиц прошивок ПНА и БПМП.** При разработке микропрограммы командного цикла (интерпретатора команд) производится распределение регистров УПР. В примере регистры программистской структуры r0–r3 занимают в блоке регистров регистры R[0]–R[3], программный счетчик и регистр признаков помещаются в одноименные регистры: PC и RP соответственно. Также в качестве вспомогательного, не являющегося регистром программистской структуры, используется регистр R[7] блока регистров. Микропрограмма командного цикла состоит из 48 микрокоманд. В первых трех ячейках БПМП находятся микрокоманды, которые обеспечивают: пересылку адреса команды из программного счетчика в регистр адреса БППД; чтение команды из БППД в регистр команд и увеличение содержимого программного счетчика на единицу; переход по коду операции к начальному адресу соответствующей микропрограммы (загрузку в RA из ПНА адреса первой микрокоманды микропрограммы выбранной операции). В остальных 45 ячейках размещаются микропрограммы 12 операций в соответствии с системой команд УПР. Ввод микропрограмм в БПМП производится с помощью экранной формы ввода ПР-В. Двоичные коды микрокоманд для рассматриваемого примера для первых 20 ячеек показаны на рисунке 5б.

**5. Ввод и отладка микропрограммы командного цикла и программы решения задачи.** Ввод данных осуществляется с помощью экранной формы ввода ПР-В. Для каждой операции в ячейку памяти ПНА, адрес которой совпадает с кодом операции, с клавиатуры заносится двоичный код начального адреса микропрограммы этой операции. В экранной форме ПР-В слева от каждой ячейки ПНА предусмотрено дополнительное поле для ввода мнемонического обозначения введенного кода операции. Это мнемоническое обозначение отображается на экранной форме исследования ПР-Ф над регистром команд при загрузке в него команды с данным кодом операции.

Микропрограмма командного цикла вводится в БПМП путем занесения с клавиатуры двоичных кодов микрокоманд. В экранной форме ПР-В справа от каждой ячейки БПМП предусмотрены два вспомогательных поля: «Пояснение» (для ввода описания выполняемых в микрокоманде микроопераций) и «СОР» (для ввода мнемонического обозначения подмикропрограммы, к которой относится данная микрокоманда).

Отладка микропрограммы командного цикла осуществляется с помощью экранной формы исследования ПР-Ф в режимах выполнения микрокоманд по полутактам и тактам. Для упрощения отладки на экранной форме ПР-Ф при выполнении микрокоманд отображается следующая дополнительная информация: источник адреса при занесении адреса в регистр адреса БППД (RA:=PC или RA:=RB); двоичный код данного, поступающий с выбранного входа мультиплексора данных для записи в БР; двоичные коды номеров регистров и находящихся там данных, поступающие на входы БАЛ для выполнения микрооперации; символическое обозначение выполняемой в БАЛ микрооперации; содержание вспомогательных полей микрокоманд («Пояснение» и «СОР») в продолжении окна микрокоманд. Кроме того, значения управляющих сигналов из микрокоманды дублируются на управляющих входах узлов и блоков, на которые они подаются. В процессе отладки микропрограммы командного цикла осуществляется выполнение необходимых последовательностей микрокоманд путем занесения в регистр адреса БПМП начального адреса первой микрокоманды, который предварительно вводится с клавиатуры в поле АМК. Перед выполнением любой микрокоманды также возможно изменение содержимого выбранных регистров в БР с помощью пультового дублера БР.

Ввод двоичных кодов команд и исходных данных программы в БППД производится в форме ввода ПР-В с помощью ячеек пультового дублера БППД. Команда может занимать одну или две ячейки БППД. В экранной форме ПР-В справа от каждой ячейки БППД предусмотрены два вспомогательных поля: «Мнемоника» (для ввода символического обозначения команд и данных) и «Комментарий» (для ввода описания выполняемой командой операции или данного).

Отладка программы решения задачи производится с помощью экранной формы исследования ПР-Ф в режиме выполнения программы по командам. При этом биты *vm* в БППД на форме ввода ПР-В перед выполнением программы устанавливаются в нулевое состояние. Для упрощения отладки на экранной форме исследования ПР-Ф при выполнении команд на выходе БППД отображается символическое обозначение из поля «Мнемоника» формы ввода ПР-В для считываемого содержимого ячейки памяти.

В процессе отладки программы выполняются отдельные команды и необходимые последовательности команд путем занесе-



ния в программный счетчик РС начального адреса первой команды, который предварительно вводится с клавиатуры в поле SA. Перед выполнением любой команды также возможно изменение содержимого выбранных регистров в БР (ячеек БППД) с помощью пульта дублера БР (БППД).

После завершения отладки микропрограммы командного цикла и программы решения задачи работа УПР исследуется в режиме выполнения программы. При этом изменяются параметры Microsoft Excel, связанные с вычислением формул: выбираются вычисления в книге «вручную» и включаются итеративные вычисления с предельным числом итераций, превышающим удвоенное максимальное число микрокоманд, необходимых для выполнения программы. В процессе исследования программы могут быть получены результаты решения задачи, подтверждающие корректность выполнения программы по всем ветвям алгоритма.

### Заключение

В применяемой функциональной модели процессора с микропрограммируемой архитектурой, кроме базовых функций, необходимых для реализации учебных процессоров, предусмотрены следующие дополнительные возможности:

- наличие на экранных формах, предназначенных для экспериментального исследования и ввода информации, специальных полей, в которых отображаются мнемонические обозначения и пояснения, облегчающие понимание изменений состояний процессора при проведении экспериментальных исследований;

- использование пультовых дублеров регистров и ячеек памяти, позволяющих заносить данные в регистры и ячейки памяти перед выполнением любой микрокоманды (команды) с использованием клавиатуры, что упрощает и ускоряет отладку микропрограмм и программ;

- наличие различных режимов работы учебного процессора, позволяющих выполнять отдельные микрокоманды, команды и всю программу.

Микропрограммная реализация и экспериментальное исследование учебных процессоров осуществляются по индивидуальным заданиям и способствуют более глубокому изучению архитектуры, структуры и рабочего цикла процессора, позволяют разработать простейший учебный процессор, создать и исследовать его действующую модель. При наличии у обучающегося необходимой предварительной теоретической подготовки решение задачи построения процессора в рассмотренной постановке не вызывает больших затруднений и формирует целостное представление о процессоре. Поскольку экспериментальные исследования выполняются в табличном процессоре Microsoft Excel, с которым, как правило, обучающиеся уже знакомы, им не требуется дополнительного времени для изучения программного интерфейса.

### Список литературы

1. CPULator Simulator Documentation. URL: [https://cpulator.01xz.net/doc/#sim\\_processors](https://cpulator.01xz.net/doc/#sim_processors) (дата обращения: 31.03.2023).
2. Титовский С.Н., Титовская Н.В. Учебный программный эмулятор процессора 18086 // Наука и современность. 2014. № 27. С. 192-196.
3. Молчанов О.Е., Васильев А.С., Белая Т.И. Компьютерный тренажер-эмулятор учебной цифровой вычислительной машины // Инженерный вестник Дона. 2015. № 2-2 (36). С. 110.
4. Коннов Н.Н., Пчелинцев А.И., Федулов В.Д., Бычков А.С. Учебный кроссплатформенный эмулятор микропрограммируемой ЭВМ // Информатика: проблемы, методы, технологии: материалы XXII Международной научно-практической конференции им. Э.К. Алгазина (Воронеж, 10-12 февраля 2022 г.) / Под ред. Д.Н. Борисова. Воронеж: Издательство ООО «Вэлборн», 2022. С. 1403-1411.
5. Страбыкин Д.А. Функциональная модель учебного процессора с микропрограммируемой архитектурой в Microsoft Excel // Современные наукоемкие технологии. 2023. № 2. С. 102-120.