

МИНИСТЕРСТВО ОБРАЗОВАНИЯ, МОЛОДЕЖИ И СПОРТА УКРАИНЫ
СЕВАСТОПОЛЬСКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

Кафедра кибернетики и
вычислительной техники

Пояснительная записка
к курсовому проекту
по дисциплине «Цифровые ЭВМ»
на темы:
«СИНТЕЗ ЦЕНТРАЛЬНОГО ОБРАБАТЫВАЮЩЕГО УСТРОЙСТВА ЦВМ»

Выполнил: студент гр. М-43д
Кущинский Д. В.
Вариант № 58
Проверил:
профессор Апраксин Ю. К.

Севастополь
2012

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. ПОСТАНОВКА ЗАДАЧИ.....	5
2. ОПИСАНИЕ ФОРМАТА ДАННЫХ И КОМАНД.....	8
2.1. Формат данных	8
2.2. Описание формат команд	9
3. СТРУКТУРНАЯ СХЕМА ЦОУ	12
4. СОДЕРЖАТЕЛЬНАЯ ГСА ФУНКЦИОНИРОВАНИЯ ЦОУ	15
5. АРХИТЕКТУРА ВНЕШНИХ ВЫВОДОВ ПРОЦЕССОРНОГО БЛОКА.....	22
6. СИНТЕЗ УА.....	24
6.1. Общая структура	24
6.2. Адресация микропрограммы	25
6.3. Кодирование поля микроопераций	26
6.4. Разработка принципиальной схемы управляющего устройства	28
7. ОПРЕДЕЛЕНИЕ ВРЕМЕННЫХ ХАРАКТЕРИСТИК РАБОТЫ ЦОУ	29
ЗАКЛЮЧЕНИЕ	31
Приложение А. Граф совместимости МО.....	32
Приложение Б. Микропрограмма функционирования ЦОУ.....	33

ВВЕДЕНИЕ

В данном курсовом проекте разработчику предлагается разработать центральное обрабатывающее устройство (ЦОУ).

Предлагается системы команд содержащая 5 видов команд: арифметическая, логическая, команда перехода, команда чтения ОП, команда чтения с устройства В/В.

Используя справочную литературу, проектировщик разработает форматы команд, составит алгоритм и построит по этому алгоритму граф-схему.

Затем рассмотрит общую структуру ЦОУ как систему, анализируя при этом из каких подсистем (отдельных устройств) должна состоять ЦОУ, чтобы функционально покрыть заданную систему команд, в результате построит структурную схему своего ЦОУ.

На этапе написания микропрограммы (МП) управления ЦОУ разработчик сначала рассмотрит структуру управляющего автомата(УА), затем закодирует условную, операционную и адресную часть микрокоманд и на этом этапе перейдёт к построению принципиальной электрической схемы, которая будет содержать: блок памяти программ, формирователь микроопераций(ФМО), формирователя адреса(ФА), блок управления сбросом и запуском ЦОУ и блок синхронизации. После чего будет проведён анализ схемы ЭП установка дополнительных элементов с целью повышения надёжности (повышение помехоустойчивости). После проектировщик рассчитает временные задержки для синхросигналов.

Главным результатом проектирования будет построенная ГСА, ЦОУ, структурная схема и принципиальная электрическая схема УУ.

1. ПОСТАНОВКА ЗАДАЧИ

Объектом курсового проектирования является центральное обрабатывающее устройство (ЦОУ), реализующее заданную совокупность команд из системы команд абстрактной ЦВМ.

Предполагается, что проектированию подлежит процессор с традиционной принстонской архитектурой.

К функциям процессорного блока относятся:

- управление потоком обработки команд исполняемой компьютером программы;
- управление процессом исполнения команд;
- управление процессом взаимодействия всех блоков ЦОУ.

ЦОУ обеспечивает реализацию хранящейся в ОП программы, команды которой принадлежат ограниченному (в учебных целях) множеству типовых команд, исполняемых компьютером.

К ним относятся:

- арифметическая команда,
- логическая команда,
- команда пересылки данных (команда обмена данными между регистровой памятью (РП) процессора и ОП),
- команда обращения к устройству ввода/вывода,
- команда передачи управления,
- команда «стоп».

Процессор, обеспечивающий исполнение каждой команды, должен:

- 1) осуществить выборку команды из ОП в строгом соответствии с форматом команды,
- 2) расшифровать код операции в команде,
- 3) выполнить расшифрованную операцию,
- 4) подготовить компьютер к выполнению следующей команды.

Обобщенная структурная схема процессора, в котором связь между составляющими этот процессор компонентами осуществляется через систему управляемых отдельных шин, приведена на рисунке 1.

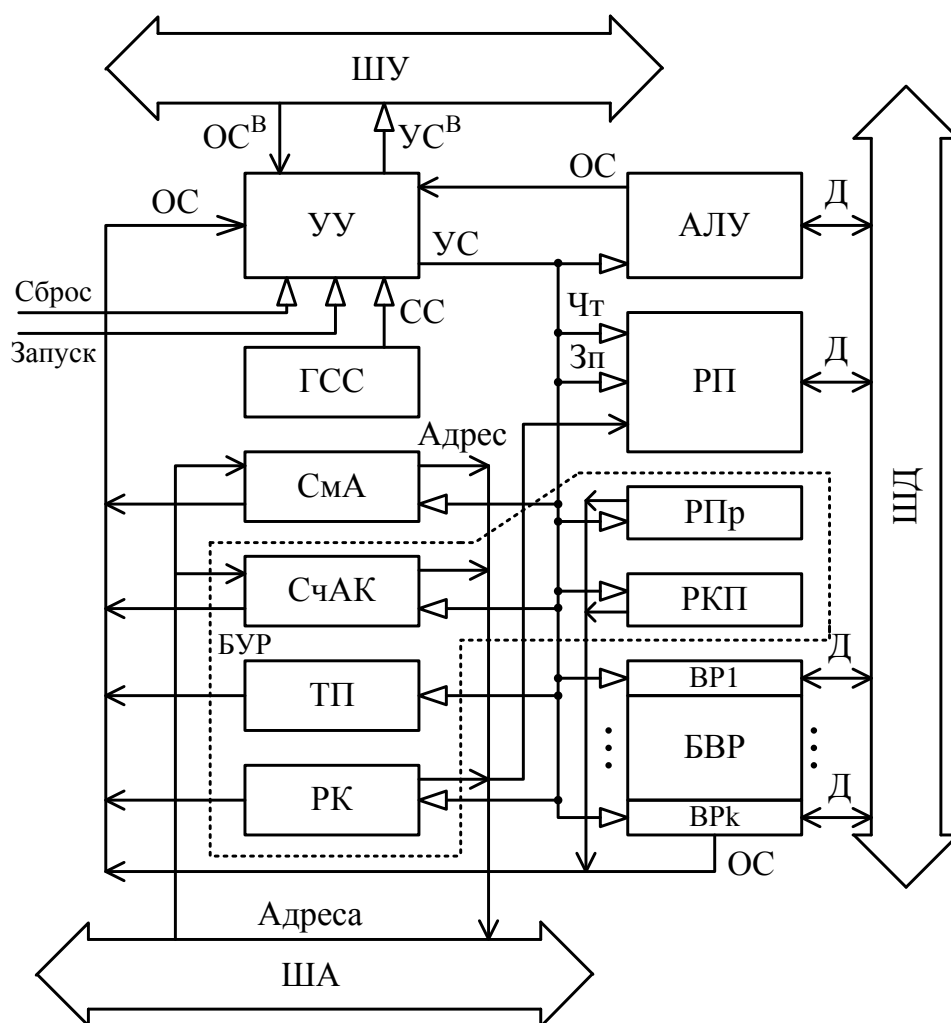


Рисунок 1.1 – Обобщенная структурная схема процессора

На рисунке 1 используются следующие сокращения:

УУ – устройство управления;

ГСС – генератор синхросигналов,

АЛУ – арифметико-логическое устройство;

РП – сверхоперативная регистровая память;

СмА – сумматор адресный;

БУР – блок управляющих регистров, в числе которых:

- РК – регистр команд,
- СчАК – счетчик адреса команд,
- РПР – регистр признака результата,
- РКП – регистр кода прерывания (регистр флагов прерывания),
- ТП – триггер переходов;

БВР – блок внутренних регистров прямого доступа (ВР1, ..., ВРk), используемых для эффективной организации процесса исполнения команд;

ОС^В – внешние осведомительные сигналы;

УС^В – внешние управляющие сигналы;

Д – данные.

Исходные данные:

Разработать центральное обрабатывающее устройство (**ЦОУ**), которое реализует заданную систему команд.

№ вар	Команды АЛУ				Дополнит. команды			ОП		РП		Способы адресации					
	КА	ДА	КЛ	ДЛ	ПУ	ОбрП	ВВ	Е _{ОП}	ШВ	Е _{РП}	Тип	Н	П	К	Р	КР	О
58	–	F4	∧	L1	ПВз	ЗП I4	ПМ	128	4	8	Р	+	+	+		+	+
Устройство управления – автомат с программируемой логикой																	
Способ адресации микрокоманд – принудительная адресация (П)																	
Способ кодирования микроопераций –вертикально-горизонтальный (ВГ)																	

Разработать систему команд, состоящую из 5 команд.

Команда арифметическая – вычитание четырехбайтных чисел с плавающей точкой (**F4**).

Команда логическая – конъюнкция над операндами размером один байт(**L1**).

Команда передачи управления – переход с возвратом.

Команда обращения к памяти – запись в оперативную память (**ЗП**) слова(**I4**).

Команда ввода/вывода – обратная передача.

Совокупность команд должна включать следующие способы адресации операндов в команде:

- непосредственная,
- прямая,
- косвенная,
- косвенная через регистр,
- относительная.

Способ адресации микрокоманд – принудительная.

Способ кодирования микроопераций – вертикально-горизонтальный (**ВГ**)

Объём ОП – 128 Мбайт.

Объём регистровой памяти – 8 ячеек.

Объём регистровой памяти регистровая память для чисел с плавающей точкой (**РПП**) – 8 ячеек.

Рекомендуемая серия микросхем – К155.

Рекомендуемая микросхема постоянного запоминающего устройства (**ПЗУ**) – К155РЕЗ.

2. ОПИСАНИЕ ФОРМАТА ДАННЫХ И КОМАНД

2.1. Форматы данных

На рисунке 2.1 показаны основные структурные единицы данных, обрабатываемых проектируемым устройством (а) и форматы их представления в процессе обработки (б).

Целые числа могут быть представлены как со знаком (S), так и без знака (для представления адресов). Числа со знаком представляются в дополнительном коде. Диапазон представления целых чисел – $[-2^n, 2^n-1]$, где n – количество разрядов числа без учета разряда знака.

Двоичные числа с плавающей точкой представляются в виде двух чисел с фиксированной точкой: порядка (P) и мантииссы (M), при этом P – целое со знаком, $|M| < 1$. В случае так называемой нормализованной мантииссы $\frac{1}{2} \leq |M| < 1$ для двоичной системы счисления. Диапазон представления чисел с плавающей точкой – $[M_{\min} \cdot 2^{P_{\min}}, M_{\max} \cdot 2^{P_{\max}}]$.

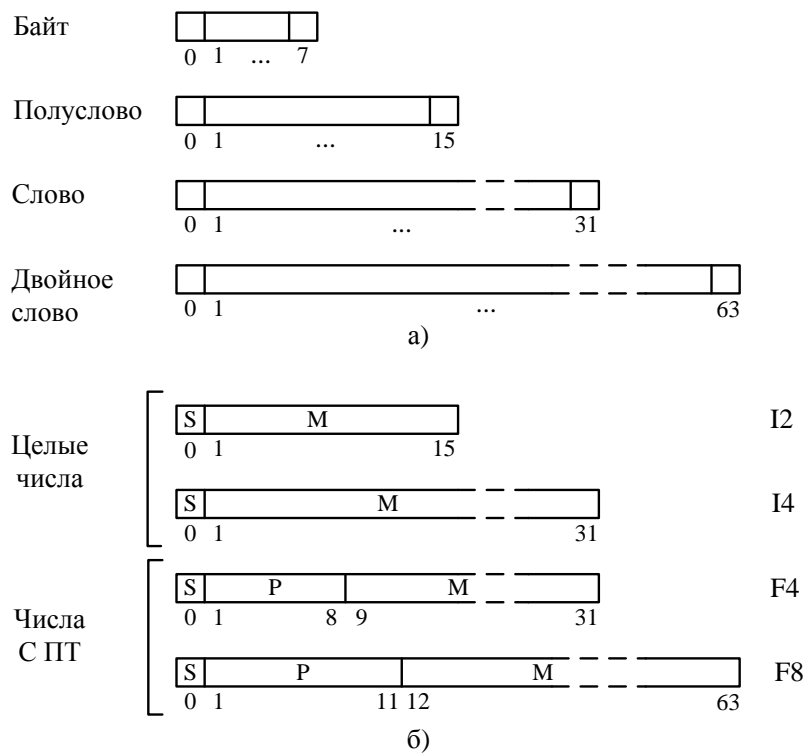


Рисунок 2.1 – Форматы данных

2.2. Описание формат команд

Все форматы команд будут кратны 2 байтам (для облегчения процесса выборки команды из ОП). Адрес команды в ОП должен быть кратен 2 (согласно методу целочисленных границ). Под поле КОП в команде отводится 1 байт (принято, что система команд процессора состоит из 128-256 команд, а мы разрабатываем фрагмент).

Разрабатывается система команд с учётом заданных параметров устройств.

Объём ОП – 128Мбайт – 2^{27} ячеек.

Следовательно, для адресации к ОП необходимо выделить 27 разрядов.

Объём РП (регистровая память) – 8 ячеек – 2^3 ячеек.

Следовательно, для адресации к РП необходимо выделить 3 разряда.

Объём РПП (регистровая память для чисел с плавающей точкой) – 8 ячеек – 2^3 ячеек.

Следовательно, для адресации к РПП необходимо выделить 3 разряда.

Опишем форматы команд.

Вычитание с плавающей точкой (класс 2).

Формат данных – 4х байтное число с плавающей точкой(**F4**).

Определим формат команды. (Рисунок 2.2)

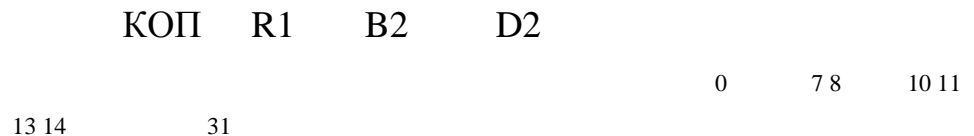


Рисунок 2.2 – Структура формата команды

Способы адресации, используемые в команде: Относительная, Регистровая. Команда вырабатывает следующие признаки результата:

- сумма равна нулю,
- сумма больше нуля,
- сумма меньше нуля,
- переполнение.

Команда может вызывать следующие прерывания программы:

- адресация,
- спецификация,

Конъюнкция (класс 4).

Операнды – двоичные вектора, длиной 1 байт(**L1**).

Так как операнды короткие, удобно использовать непосредственную адресацию.

Определим формат команды. (Рисунок 2.3)

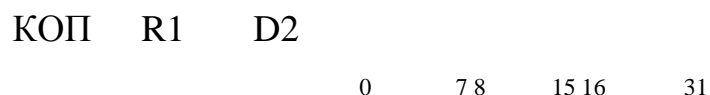


Рисунок 2.3 – Структура формата команды

Способы адресации, используемые в команде: Непосредственная, Косвенная.

Команда вырабатывает следующие признаки результата:

- результат равен нулю,
- результат не равен нулю.

Команда может вызывать следующие прерывания программы:

- адресация.

Команда перехода по индексу меньше (класс 6).

Команда сохраняет адрес текущей команды и передает управление в другое место.

Определим формат команды. (Рисунок 2.4)

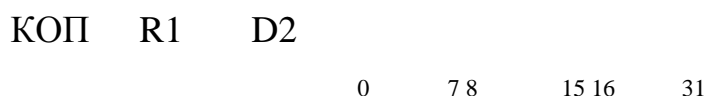


Рисунок 2.4 – Структура формата команды

Способы адресации, используемые в команде: Относительная, Регистровая.

Команда не вырабатывает признаки результата.

Команда может вызывать следующие прерывания программы:

- адресация.

Команда чтения из памяти (класс 1).

Операнд, содержащий число с фиксированной точкой длиной 4 байта (I4) переписывается из ячеек ОП, по адресу в регистре, номер которого содержит R2 в регистровую память по адресу заданному в регистре, номер которого хранится в R1. 8-байтные операнды располагаются в регистровых парах РП. Номер первого регистра пары четный.

R1 - номер регистра, в который будет записано число из ОП.

R2(кр) - номер регистра, который содержит адрес ОП по которому располагается число.

Определим формат команды. (Рисунок 2.5)

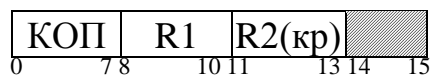


Рисунок 2.5 – Структура формата команды

Способы адресации, используемые в команде: Регистровая, Косвенная через регистр.

Команда не вырабатывает признаки результата.

Команда может вызывать следующие прерывания программы:

- адресация,
- спецификация.

Нарушение адресации и спецификации (адрес должен заканчиваться на три нуля, номер регистра, указанного в поле R1 должен быть четным).

Команда ввода-вывода – ПМ (класс 7).

Байт данных из порта процессора пересылается в порт устройства, адрес которого задан в команде (прямая адресация).

8-разрядный адрес дает возможность подключить к MBV 256 устройств ввода/вывода.

Определим формат команды. (Рисунок 2.6)

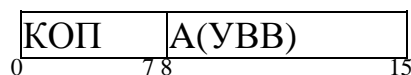


Рисунок 2.6 – Структура формата команды

Способы адресации, используемые в команде: Прямая.

Команда не вырабатывает признаки результата.

Команда не вызывает прерывания программы.

Таким образом, все заданные вариантом способы адресации реализованы.

Будем считать, что проектируемое устройство, реализующее пять определенных вариантом задания команд, является фрагментом процессора, реализующего от 128 до 256 команд. При этом в поле КОП команды будем выделять три поля: КОП (0:1) – для кода формата команды, КОП (2:4) – для кода класса команды, КОП (5:7) – для номера команды в списке класса. Код длины формата: 00 – 2 байта, 01 – 4 байта.

Результаты разработки системы команд сведем в таблицу 2.1.

Таблица 2.1 - Система команд процессора

Номер класса	Номер подкласса	Название	Содержание	ПР	Флаг	Код двоичн.			Код шест.
						01	234	567	
2	1	Вычитание с плавающей точкой	$РП[R1] = РП[R1] - ОП[(B2)+D2]$	>0 <0 =0 пр.	A S ППФ	01	010	001	8Ah
4	1	Логическая операция – конъюнкция	$ОП[(B2)+D2] = Im1 \wedge ОП[(B2)+D2]$	=0 ≠0	A	01	100	001	86h
6	1	Переход с возвратом	$РСРПф:=000.РПр(0:1). СЧАК(0:26). СЧАК = (B2)+D2$	нет	A	01	110	001	8Eh
1	1	Чтение из памяти	$РП[R1] = ОП[РП[R2]]$	нет	A S	00	001	001	90h
7	1	Команда В/В – из процессора в MBV	$УВВ[A_{УВВ}] = ПортД$	нет	нет	00	111	001	9Ch

3. СТРУКТУРНАЯ СХЕМА ЦОУ

Схема обмена данными между Процессором, ОП и МВВ изображена на рисунке 3.1.

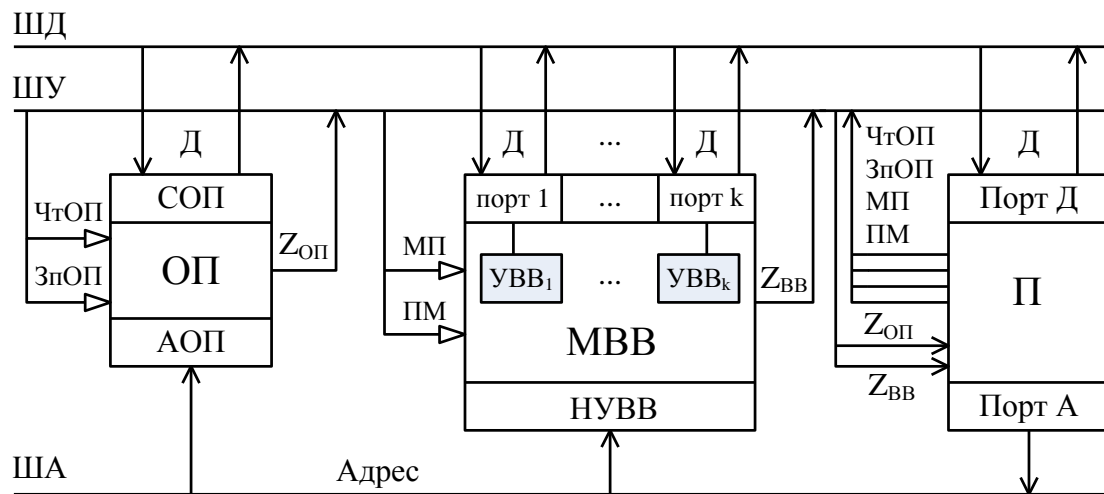


Рисунок 3.1 - Схема обмена данными в ЦОУ

Используются следующие обозначения:

- СОП – входы/выходы данных (слово ОП),
- АОП – адресные входы ОП (адрес слова ОП),
- порт 1 ... порт k – порты (регистры), через которые внешние устройства $УВВ_1...УВВ_k$ обмениваются данными с процессором (П),
- НУВВ – адресные входы МВВ (на них поступает номер (адрес) устройства ввода/вывода),
- порт Д – порт процессора, через который он обменивается данными с внешними устройствами (ОП и МВВ),
- порт А – порт, через который процессор выдает адрес слова ОП или номер УВВ.

Обмен типа «П-МВВ» реализуется при исполнении команд ввода (МП), обеспечивающих перемещение данных из адресуемого устройства ВВ в П, или вывода (ПМ), обеспечивающих обратную передачу от П к устройству ВВ. Предполагается, что передача данных осуществляется 8-разрядными двоичными векторами. Формат команды ввода/вывода представлен на рисунке 3.2.



Рисунок 3.2 – Формат команды ввода/вывода

Передача данных при вводе осуществляется следующим образом:

- 1) П направляет адрес выбранного устройства ввода на МВВ по адресной шине, определяя с тем самым с какого порта ввода должны быть получены данные;
- 2) по шине управления П посылает МВВ сигнал ввода (МП);
- 3) данные, представленные в адресуемом порте ввода, пересылаются в порт данных П, если $Z_{ВВ}$ (осведомительный сигнал) определяет выбранное устройство как готовое к выполнению операции.

Операция вывода осуществляется аналогичным вводу способом, но передача данных осуществляется в обратном направлении.

Обмен типа «П-ОП» происходит в следующих случаях:

- 1) при реализации команд загрузки регистра (ЗР) и записи в память (ЗП);
- 2) в процессе выборки команды из ОП;
- 3) в процессе исполнения арифметической или логической команды при выборке из ОП операндов и записи в ОП результата.

Главным элементом процессорного блока является АЛУ.

Условное обозначение АЛУ показано на рисунке 3.3.

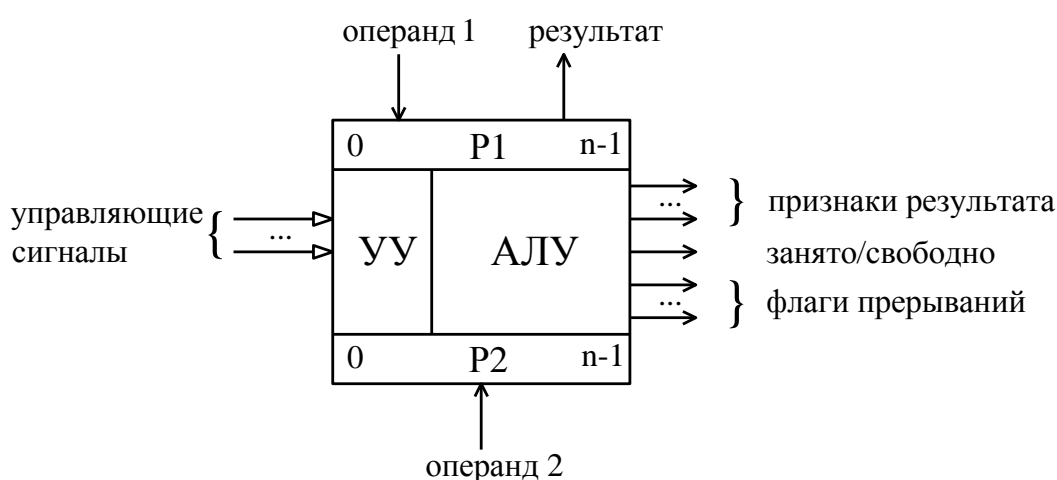


Рисунок 3.3 – Условное обозначение АЛУ

Двуместные операции в АЛУ выполняются над операндами, хранящимися в регистрах P1(0:n-1) (первый операнд) и P2(0:n-1) (второй операнд). Результат выполненной операции помещается в P1. Разрядность регистров n выбирается равной максимальной длине операндов. Каждая операция АЛУ инициируется своим управляющим сигналом, поступающим на вход устройства управления (УУ). АЛУ вырабатывает три группы осведомительных сигналов: признаки результата, флаги прерываний, признак занятости.

Регистровая память (РП) организуется как сверхоперативная память с прямой адресацией. РП допускает выборку слова в течение одного такта. По типу РП определяется как раздельная (для данных в формате с фиксированной точкой и с плавающей точкой) или раздельная, состоящая из двух самостоятельных блоков (регистры общего назначения (РОН) и регистры данных формата с плавающей точкой (РПТ)). Условное обозначение блока РП показано на рисунке 3.4.

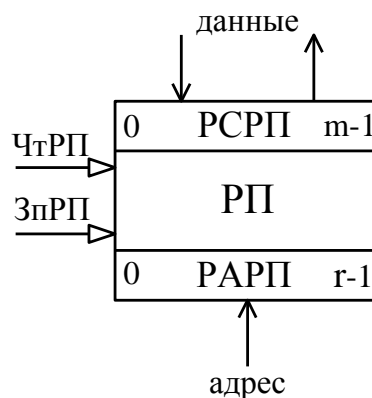


Рисунок 3.4 – Условное обозначение РП

Разрядность m регистра слова регистровой памяти (РСПП) определяется длиной информационного слова. Для проектируемого устройства $m=32$. Разрядность r регистра адреса регистровой памяти (РАРП) определяется как $\lceil \log_2 E_{РП} \rceil$, где $E_{РП}$ – количество регистровых схем в блоке памяти.

На регистр команд (**РК**) выбирается из ОП команда, подлежащая исполнению. Разрядность РК определяется максимальной длиной формата команды (4 байта).

Счетчик адреса команды (**СЧАК**), предназначенный для хранения адреса очередной выбираемой из ОП команды, обеспечивает адресацию к любому байту ОП.

Сумматор адреса (**СмА**), предназначен для вычисления исполнительного адреса ($A_{ИСП}$) в случае его задания в виде нескольких компонент (например, при относительной адресации $A_{ИСП}=(B)+D$). По разрядности СмА отличается от СЧАК дополнительным старшим разрядом, фиксирующим переполнение.

Триггер перехода (**ТП**) сбрасывается в нулевое состояние, если исполняемая команда относится к группе команд обработки данных, и устанавливается в 1 в случае формирования адреса перехода при исполнении команды передачи управления. Учет состояния ТП может потребоваться при выборке команд из ОП.

Регистр признака результата (**РПр**) состоит из двух разрядов.

Регистра флагов прерываний (**РФП**) состоит из трёх триггеров:

- переполнения с фиксированной точкой (**Тпфт**),
- флаг нарушения спецификации (**S**),
- флаг нарушения адресации (**A**).

Буферный регистр (**БР**) выбирается размерностью полуслово.

Регистр слова оперативной памяти (**РСОП**) выбирается размером слово.

Регистр адреса оперативной памяти (**РАОП**) выбирается размером 26 разрядов.

Порт адреса (**ПортА**) выбирается размером 26 разрядов.

Порт данных (**ПортД**) выбирается размером слово.

Структурная схема ЦОУ представлена на чертеже 2012.М-43д.02.

4. СОДЕРЖАТЕЛЬНАЯ ГСА ФУНКЦИОНИРОВАНИЯ ЦОУ

Алгоритм работы ЦОУ должен обеспечивать выполнение следующих действий:

- 1) Выборка команды;
- 2) Выполнение команды;
- 3) Если обнаружены нарушения адресации или спецификации – сформировать прерывание;
- 4) Подготовиться к выборке следующей команды.

Если очередная команда Стоп, то ЦОУ прекращает работу. Упрощенная схема алгоритма функционирования ЦОУ представлена на рисунке 4.1.

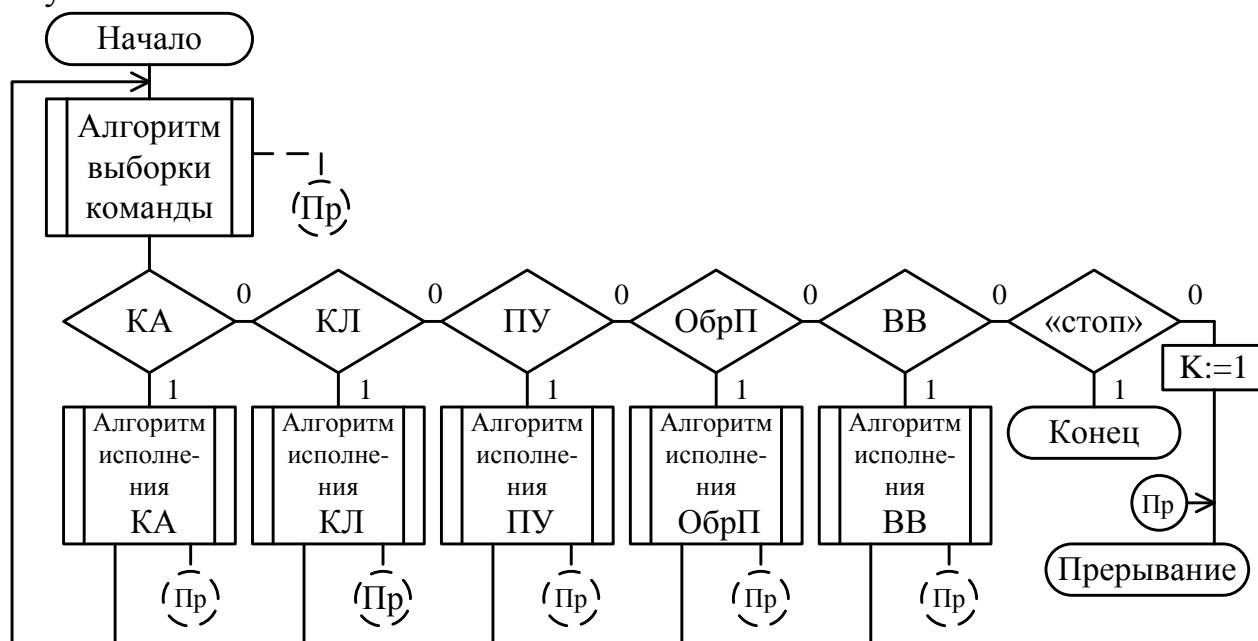


Рисунок 4.1 – Упрощенная схема алгоритма функционирования ЦОУ

Объединенная ГСА функционирования процессора представлена на чертеже 2012.М-43д.01.

Длины команд составляют 2 и 4 байта, а ширина выборки – 4 байта. Корректный адрес команды должен быть кратен 2, а его значение не превышать предельно допустимое, определяемое емкостью ОП. Все возможные случаи размещения команд в ОП представлены на рисунке 4.2

0	СОП		31
...			
K _{2B}		K _{2B}	
1/2 K _{4B}		1/2 K _{4B}	
K _{2B}		1/2 K _{4B}	
1/2 K _{4B}		K _{2B}	
1/2 K _{4B}		1/3 K _{6B}	
1/3 K _{6B}		1/2 K _{4B}	
1/3 K _{6B}		1/3 K _{6B}	
K _{2B}		1/3 K _{6B}	
1/3 K _{6B}		K _{2B}	
...			
0	АОП		26

Рисунок 4.2 – Размещение команд в ОП

ГСА выборки команды изображено на рисунке 4.3.

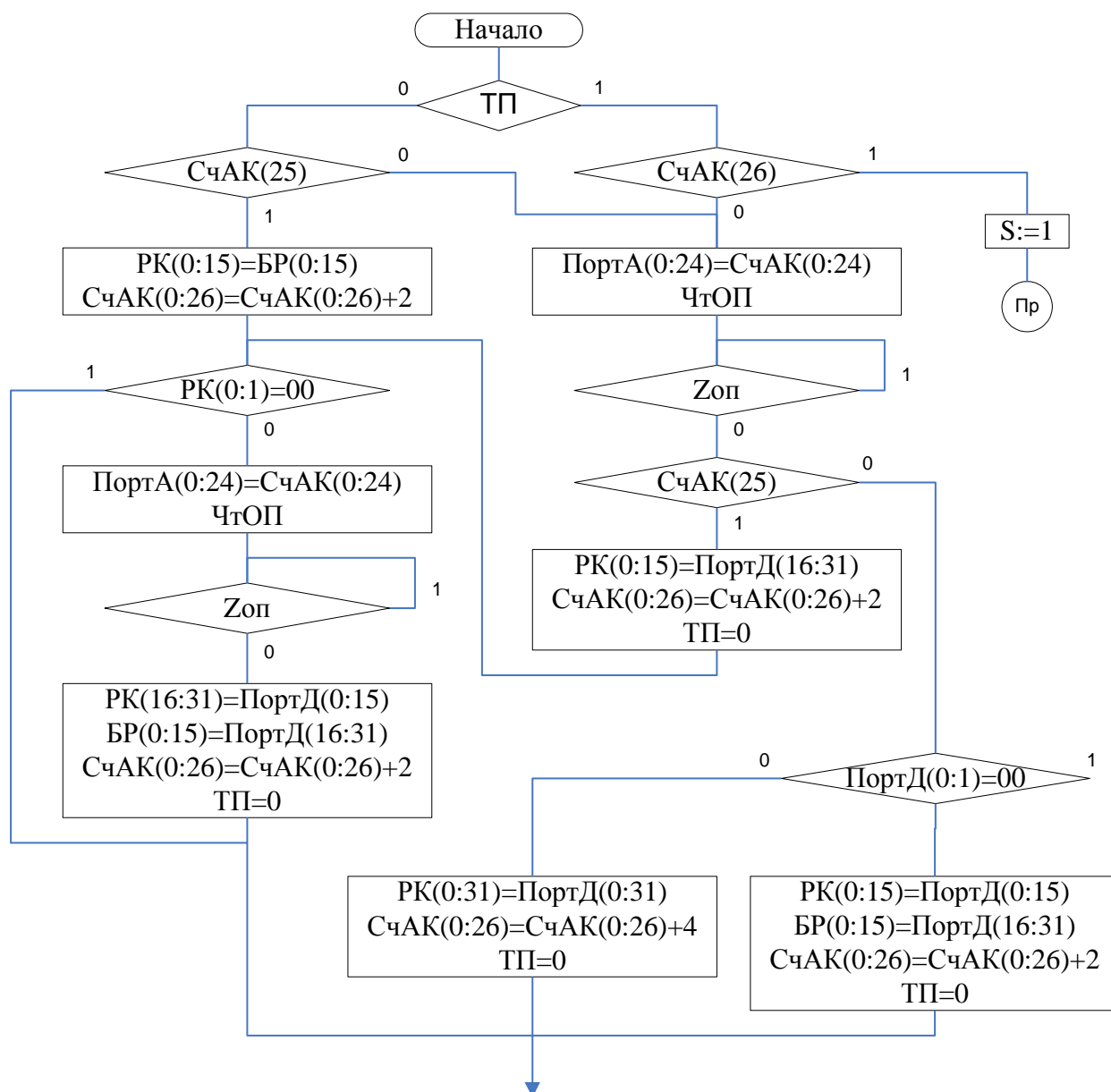


Рисунок 4.3. – ГСА выборки команд

Для реализации команд необходимы следующие модели устройств.

РК – регистр команд, хранит команду.

СмА (0:26) – сумматор адреса, состоит из 27 разрядов (0-й разряд используется для выявления факта переполнения), используется для вычисления полного адреса путём сложения базы и смещения.

АЛУ - арифметико-логическое устройство, выполняет операции: сложение, конъюнкция, сравнение. Разрядность регистров Р1 и Р2 АЛУ выбирается в соответствии с максимальной длиной данных, обрабатываемых на АЛУ. В данном случае $n=32$.

РП - регистровая память, состоит из 8 регистров, используется для временного хранения чисел.

РПП - регистровая память для чисел с плавающей точкой, состоит из 8 регистров.

РПр (0:1) - регистр признака результата, состоит из двух разрядов.

Тпфт - триггер переполнения с фиксированной точкой, состоит из одного разряда.

Триггер S - флаг нарушения спецификации, состоит из одного разряда.

Триггер А - флаг нарушения адресации, состоит из одного разряда.

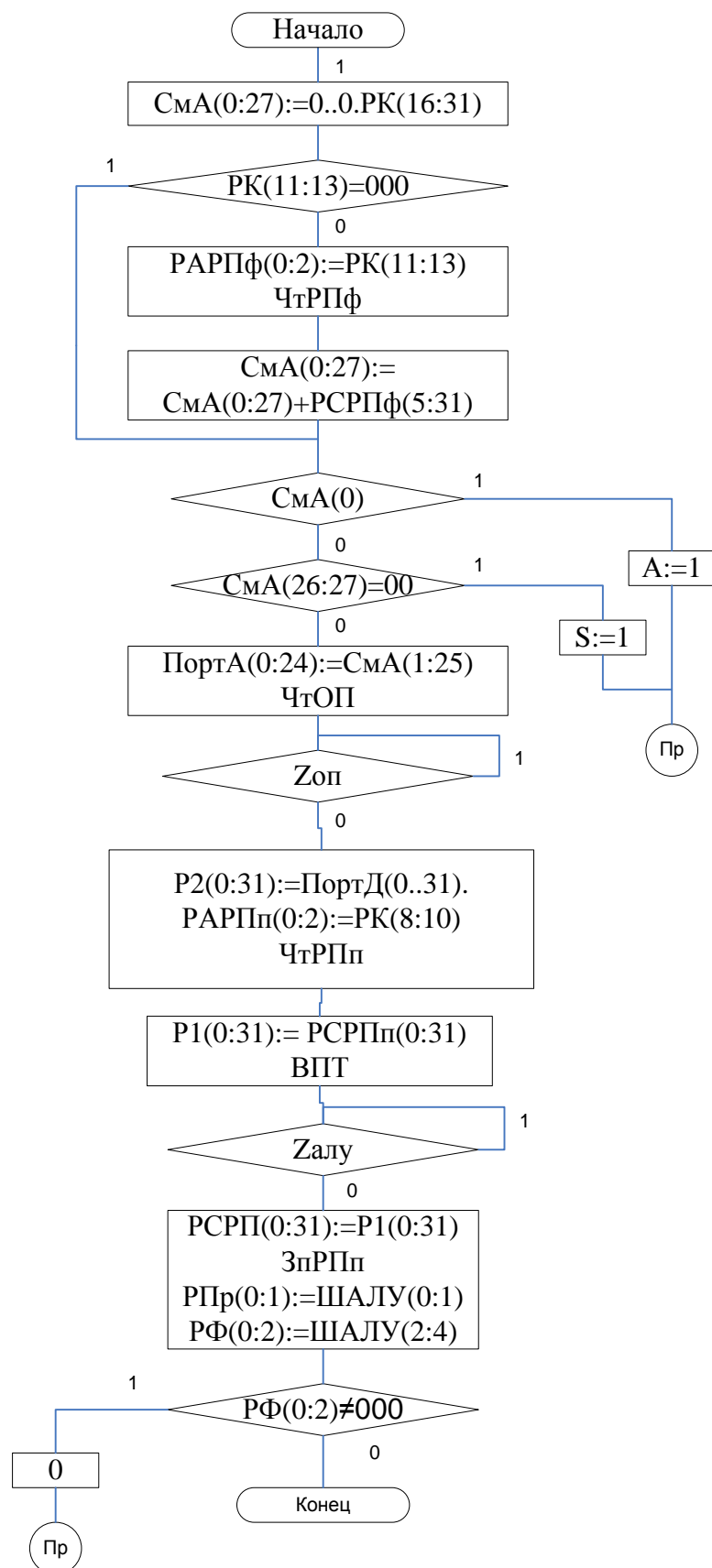


Рисунок 4.4. – ГСА команды – вычитание с плавающей точкой

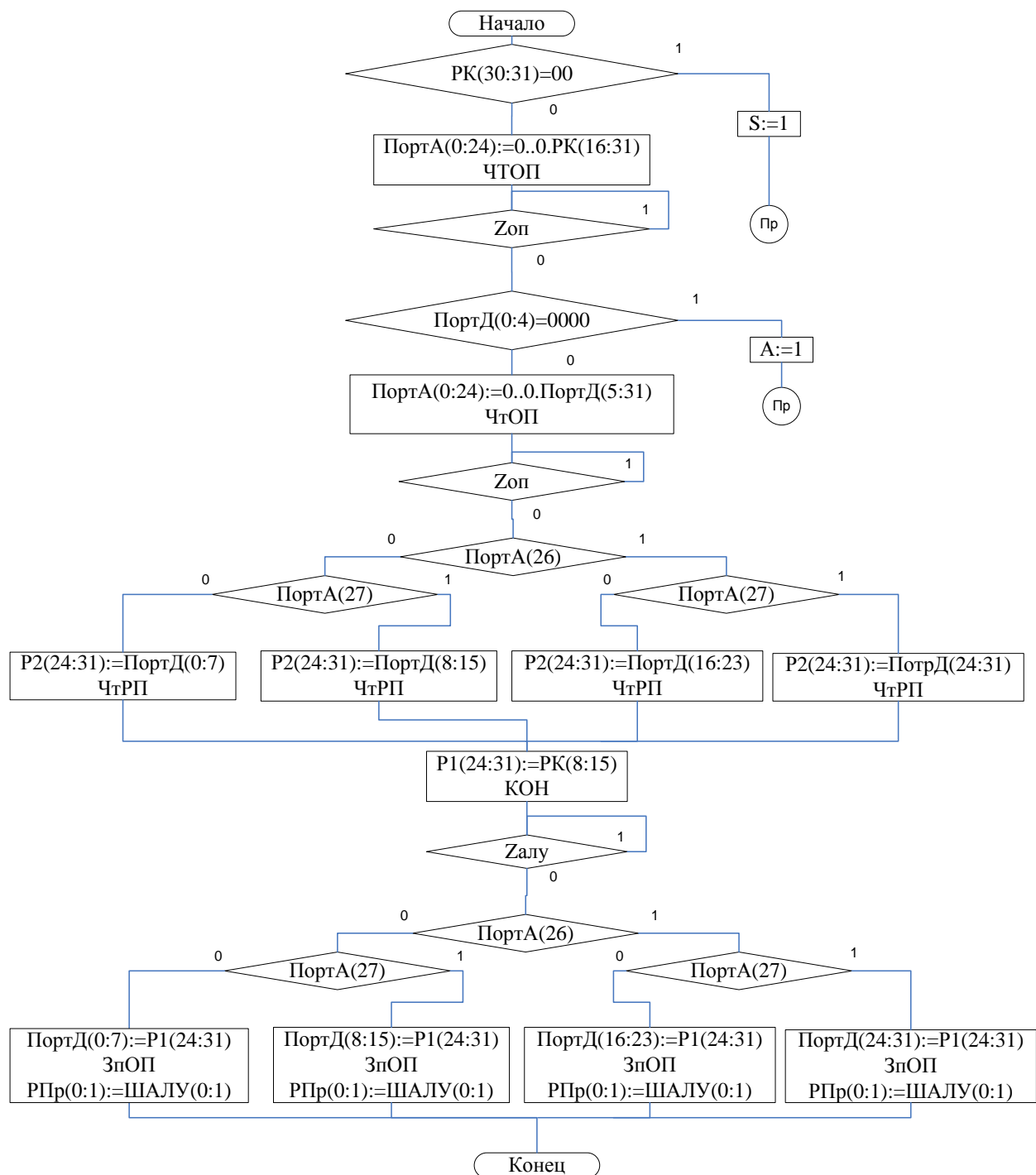


Рисунок 4.5. – ГСА логической команды - конъюнкция

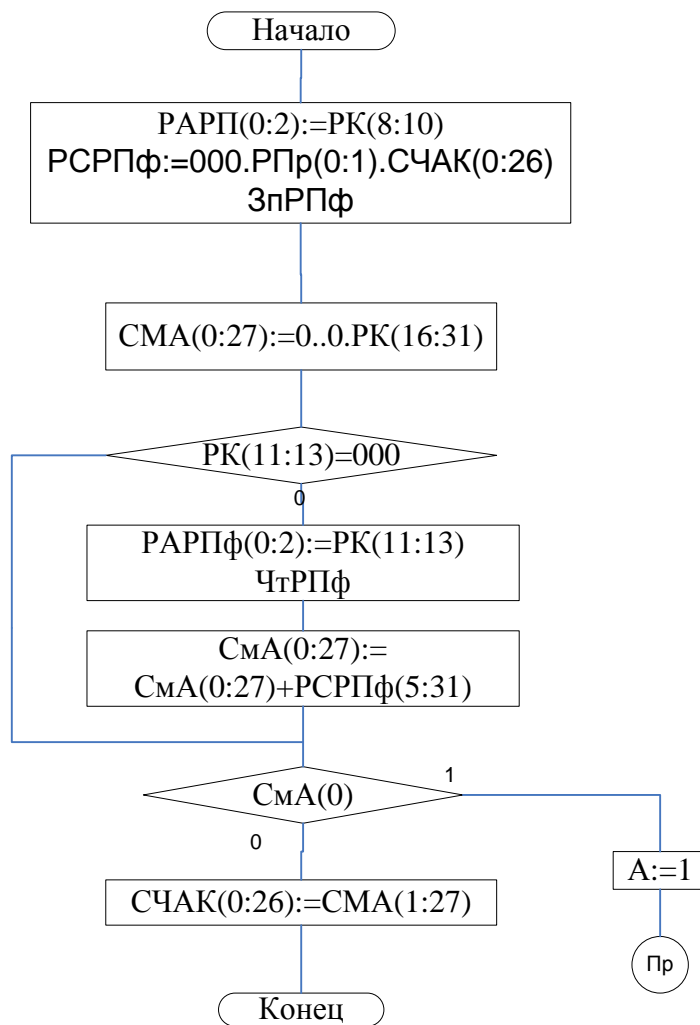


Рисунок 4.6. – ГСА команды – передачи управления

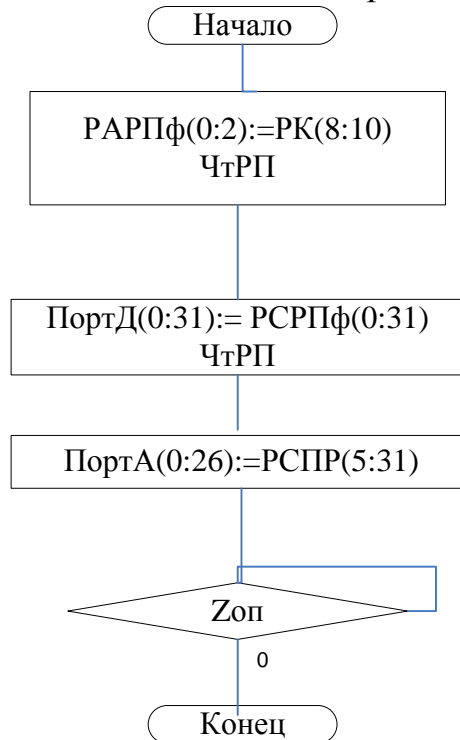


Рисунок 4.7. – ГСА команды – запись в ОП

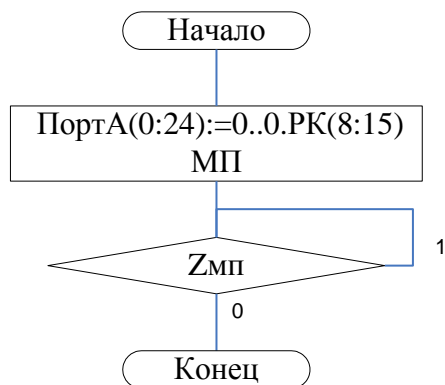


Рисунок 4.8 – ГСА команды – чтение из MBV в процессор

После алгоритма выборке команды проверяется первый байт РК, если он равен одной из команд, то осуществляется переход на алгоритм этой команды и после её выполнения переход на алгоритм выборки команды и т.д., иначе устанавливается триггер конца (**К**) и переход на конец. При возникновении прерывания также осуществляется переход на конец.

Упрощенная структура объединённого ГСА функционирования процессора изображена на рисунке 4.1.

5. АРХИТЕКТУРА ВНЕШНИХ ВЫВОДОВ ПРОЦЕССОРНОГО БЛОКА

Процессор имеет 27 внешних выводов адреса (A_i) и 32 внешних выводов данных (D_i). Выводы RD, WR, IN, OUT используются соответственно для сигналов «чтение» (ЧтОП), «запись» (ЗпОП), «ввод» (МП) и «вывод» (ПМ). Контакт READY1 (готовность) используется для приема сигнала готовности Z_{OP} от медленного внешнего по отношению к процессору ОП. Контакт READY2 используется для приема сигнала готовности Z_{BV} от медленного внешнего по отношению к процессору устройства MBV. Вывод +5V используется для питания, GND – земля, RESET – сброс, SET – установка, CLC – синхронизация.

Если подсчитать количество функциональных выводов на данный момент, то их будет 70. Так как при разработке архитектуры внешних выводов следует ориентироваться на использование стандартных корпусов, имеющих ограниченное количество выводов, а именно 40, 68, 132, 144, 168, то правильным решением будет использование 132-ух контактного корпуса. Распределение свободных выводов будет таковым:

- 30 свободных контактов используется для линий питания;
- 30 свободных контактов используется для GND;
- 2 свободных контакта используется для увеличения разрядности ША.

Обобщенное условное графическое обозначение процессорного блока приведено на рисунке 5.1.

Нумерация контактов:

- 1) 1:31 – питание;
- 2) 32 – используется для приема сигнала готовности Z_{OP} от медленного внешнего по отношению к процессору устройства ОП;
- 3) 33 – используется для приема сигнала готовности Z_{BV} от медленного внешнего по отношению к процессору устройства MBV;
- 4) 34 – сброс;
- 5) 35 – установка;
- 6) 36 – синхронизация;
- 7) 37:67 – земля;
- 8) 68:96 – адресные выводы (27 + 2 для увеличения разрядности ША);
- 9) 97:128 – выводы данных;
- 10) 129 – используется для сигнала «ЧтОП»;
- 11) 130 – используется для сигнала «ЗпОП»;
- 12) 131 – используется для сигнала «Ввод»;
- 13) 132 – используется для сигнала «Вывод»;

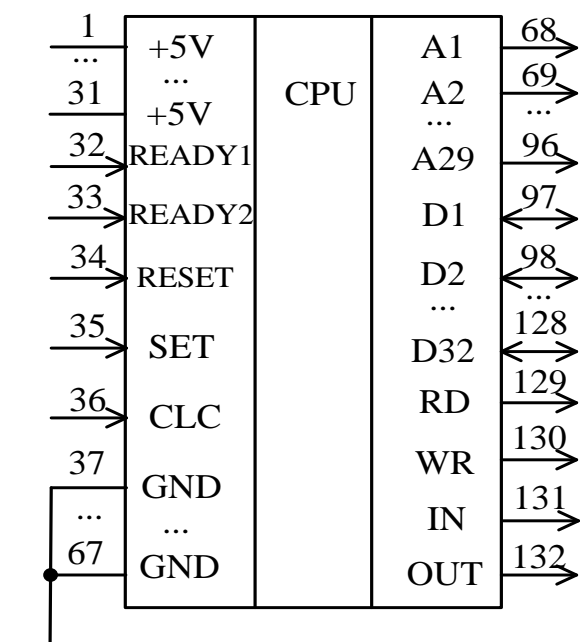


Рисунок 5.1 – Обобщенное условное графическое обозначение процессорного блока

6. СИНТЕЗ УА

6.1. Общая структура

Обобщенная структура УА с программируемой логикой представлена на рисунке 6.1.

Память микропрограмм (ПМП) организована в виде ПЗУ. В качестве АМК используется счетчик (в случае естественной адресации) или регистр (в случае принудительной адресации). Пуск автомата осуществляется подачей в схему управляющего сигнала **В**, разрешающего подачу тактирующих сигналов на ПМП, останов – подачей управляющего сигнала **А**. Сигнал Сброс или Уст (установка) устанавливает на АМК адрес начальной микрокоманды в микропрограмме. Управляющий сигнал ЧтМК выбирает из ПМП на регистр МК (РМК) очередную МК. Схема формирователя сигналов МО (ФСМО) расшифровывает поле МО и вырабатывает управляющие сигналы, иницирующие выполнение процессором конкретной МК. Формирователь адреса (ФА), сравнивая поле ЛУ в МК с двоичным вектором осведомительных сигналов (x_1, x_2, \dots, x_n), принимает решение о передаче кода НА на АМК.

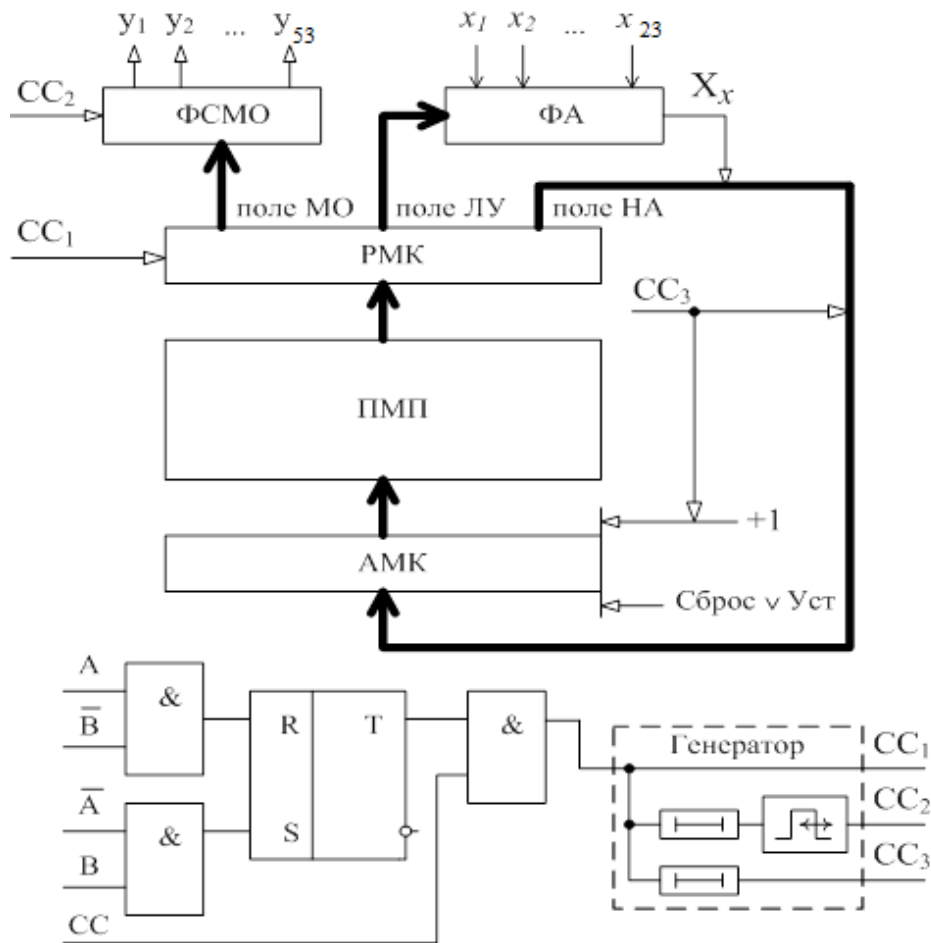


Рисунок 6.1 – Обобщенная структура УА с программируемой логикой (В – сигнал, запускающий УА, А – сигнал, останавливающий УА)

6.2. Адресация микропрограммы

Принцип принудительной адресации МК предполагает единый формат МК типа **М.Х.А**. Причем исполнительный адрес следующей МК определяется в предыдущей МК по правилу

$$A_{\text{исп}} = \begin{cases} A.1, \text{ если } X = i \ (x_i = 1), \\ A.0, \text{ в противном случае.} \end{cases}$$

В процессе выполнения микроопераций может измениться значение осведомительного сигнала x_i , номер которого i задан в поле **X** МК. По окончании выполнения микроопераций и выработки осведомительных сигналов в ОА, рассматриваемая МК инициирует переход к МК, расположенной в ПМК по четному адресу **A.0**, если $x_i = 0$. Если $x_i = 1$, то переход осуществляется к микрокоманде, расположенной по нечетному адресу **A.1**.

Таблица 6.2 – Результат расстановки адресов

Адрес	МК	A14.0	y18y19y20 x0	A28.1	y0x17
A0.0	y0 x1	A14.1	y0 x5	A29.0	y36y40y41 x0
A0.1	-	A15.0	y21y22 x11	A29.1	y37y40y41 x0
A1.0	y0 x2	A15.1	-	A30.0	y38y40y41 x0
A1.1	y0 x3	A16.0	y23y34y25y26 x12	A30.1	y39y40y41 x0
A2.0	y3y4 x5	A16.1	y0x11	A31.0	y0 x19
A2.1	y1y2 x4	A17.0	-	A31.1	y42y43y33 x0
A3.0	y3y4 x5	A17.1	y16 x0	A32.0	y44 x8
A3.1	y51 x0	A18.0	y0 x18	A32.1	-
A4.0	y3y4 x5	A18.1	y0 x14	A33.0	y13y14 x0
A4.1	y0 x7	A19.0	y16 x0	A33.1	y0x9
A5.0	y7y8y2y6 x23	A19.1	y27y4 x5	A34.0	y15 x23
A5.1	y0 x5	A20.0	y0 x15	A34.1	-
A6.0	y0 x2	A20.1	y0 x5	A35.0	y45 x0
A6.1	y0 x5	A21.0	y28y4 x5	A35.1	y53 x0
A7.0	y0 x6	A21.1	y53 x0	A36.0	y0x20
A7.1	y5y2y6 x4	A22.0	y0 x16	A36.1	y42y14 x0
A8.0	y9y10y6 x23	A22.1	-	A37.0	y45y14 x0
A8.1	y8y11y2y6 x23	A23.0	y0 x17	A37.1	-
A9.0	y0 x13	A23.1	y0 x17	A38.0	y47 x5
A9.1	y12 x8	A24.0	y29y20 x0	A38.1	-
A10.0	y13y14 x0	A24.1	y30y20 x0	A39.0	-
A10.1	-	A25.0	y31y20 x0	A39.1	y0 x5
A11.0	y15 x9	A25.1	y32y20 x0	A40.0	y0 x22
A11.1	-	A26.0	y34y35 x11	A40.1	y47y48 x21
A12.0	y0 x10	A26.1	-	A41.0	-
A12.1	y53 x0	A27.0	y0x16	A41.1	y0 x21
A13.0	y16 x0	A27.1	y0x11	A42.0	y49 x0
A13.1	y17y4 x5	A28.0	y0x17	A42.1	y50 x0

6.3. Кодирование поля микроопераций

При вертикально-горизонтальном способе кодирования все множество МО делится на подмножества, однако в каждое подмножество включаются только те МО, которые связаны между собой отношением совместимости по времени исполнения (встречаются вместе хотя бы в одной МК). Для всех этих подмножеств выделяется в операционной части МК одно поле M_3 , длина которого определяется максимальным количеством МО в подмножествах. Принцип кодирования МО в поле M_3 – горизонтальный. Идентифицирующее поле M_2 заполняется вертикальным кодом номера подмножества, зафиксированного в поле M_3 . Отличительной особенностью вертикально-горизонтального способа кодирования является требование несовместимости выделенных подмножеств МО между собой. Удовлетворить этому требованию можно, выделив наиболее часто встречающиеся в МК микрооперации в отдельное подмножество (универсальную группу). Кодирование МО универсальной группы – горизонтальное. Код универсальной группы помещается в поле M_1 операционной части МК.

Для более наглядного отображения совместимых микроопераций (МО) построим граф совместимости МО. Данный граф приведен в Приложении А.

Получена универсальная группа: y6 y2 y25 y40 y42 y20

Подмножества несовместимых МО, представлены в Таблице 6.3.

Таблица 6.3. Группы несовместимых подмножеств микроопераций

код группы	микрооперация
0001	y1 y5 y16 y0
0010	y7 y8 y9 y10 y11
0011	y3 y4 y17 y27 y28
0100	y21 y22 y23 y24 y26
0101	y12 y13 y14 y45
0110	y36 y37 y38 y41 y39
0111	y44 y15 y46 y51 y53
1000	y19 y30 y32 y47 y48
1001	y31 y33 y34 y35 y43
1010	y18 y19 y49 y50 y29

Количество осведомительных сигналов ЦОУ – 24, что потребует 5 бит в поле МК для логического условия. Кодировать их будет тривиальным позиционным кодом.

Таблица 6.3.1. Коды осведомительных сигналов

x0	0	0	0	0	0
x1	0	0	0	0	1
x2	0	0	0	1	0
x3	0	0	0	1	1
x4	0	0	1	0	0
x5	0	0	1	0	1
x6	0	0	1	1	0
x7	0	0	1	1	1
x8	0	1	0	0	0
x9	0	1	0	0	1
x10	0	1	0	1	0
x11	0	1	0	1	1
x12	0	1	1	0	0
x13	0	1	1	0	1
x14	0	1	1	1	0
x15	0	1	1	1	1
x16	1	0	0	0	0
x17	1	0	0	0	1
x18	1	0	0	1	0
x19	1	0	0	1	1
x20	1	0	1	0	0
x21	1	0	1	0	1
x22	1	0	1	1	0
x23	1	0	1	1	1

Количество адресов равно 42, следовательно, в поле МК для определения адреса потребуется выделить $\lceil \log_2 42 \rceil = 6$ бит.

Получим следующий формат микрокоманды:

M1						M2				M3					X					A					
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Микропрограмма функционирования центрального обрабатывающего устройства приведена в Приложении Б.

6.4. Разработка принципиальной схемы управляющего устройства

Память МП имеет размерность 84х26. Для реализации ПМП (МРМ) была взята ИМС K155PE3. Микросхема представляет собой электрически программируемое посредством пережигания плавких перемычек постоянное запоминающее устройство (ППЗУ) емкостью 256 бит (32х8). Для реализации ПМП понадобится 3 линейки по 4 микросхемы K155PE3.

На адресные входы всех микросхем подаются разряды 6...2 с регистра адреса микрокоманд (РАМК, МІА) – адресация внутри линейки. Старшие адресные разряды (МІА(1:0)) подаются на дешифратор, реализованный в виде комбинационной схемы с помощью логических элементов 2И-НЕ микросхем K155ЛА3 (DD17-DD18). Выходы дешифратора активируют одну из 3 линеек ПМП.

Регистр микрокоманд имеет разрядность 23 бит. Для реализации его в схеме электрической принципиальной использовались 2 регистра K155ИР13 (DD10-DD11). В исходном состоянии по всем адресам и разрядам записан логический ноль. По стробу СС1 данные с памяти микропрограмм поступают на входы регистра микрокоманд (РМК, МІР).

Регистр адреса микрокоманд имеет разрядность 7 бит и для реализации его в схеме используется два 4-разрядных счетчика K155ИЕ7 при чем первый тактируется сигналом СС3, а второй – выходом прямого переноса первого. По стробу СС3 в РАМК подаются данные из регистра микрокоманд или значение адреса в РАМК увеличивается на 1, в зависимости от информационного бита-маркера (РАМК[0]).

В качестве формирователя разрешающего сигнала Хх использовались два мультиплексора K155КП1. В качестве старшего адресного бита используется бит РМК(5), который подается в прямом и инверсном виде на стробирующие входы мультиплексоров. Сигналы РМК подаются на адресные входы обеих микросхем. Для логического объединения инверсных выходов мультиплексоров для формирования разрешающего сигнала Хх загрузки адреса в РАМК используется логический элемент 2И-НЕ одной из микросхем K155ЛА3.

В качестве формирователя сигналов микроопераций использовались 10 микросхем K155ЛЕ1 (для формирования сигналов подмножеств), 2 микросхемы K155ЛИ1 (для формирования сигналов МО УГ) и дешифратор K155ИД3 (для декодирования номера подмножества, разрешающий сигнал – с участием тактового сигнала).

Схема электрическая принципиальная УУ с перечнем элементов представлена на чертеже 2012.М-43д.03.

7. ОПРЕДЕЛЕНИЕ ВРЕМЕННЫХ ХАРАКТЕРИСТИК РАБОТЫ ЦОУ

Временные характеристики работы ЦОУ рассчитываются, исходя из алгоритма функционирования ЦОУ (алгоритма исполнения МК) и схемотехнических особенностей реальных схем, составляющих процессор.

Промежуток времени, достаточный для реализации процессором любой МК, называется тактом. Другими словами, такт – это период синхросерии, обеспечивающей стабильную работу операционного устройства (ОУ). Так как любое ОУ, в том числе и проектируемое ЦОУ, состоит из управляющего устройства (управляющего автомата) и обрабатывающего блока (операционного автомата), такт операционного устройства в случае последовательной работы УУ и ОБ определяется по формуле:

$$T_{Oy} = T_{yU} + T_{OБ},$$

где T_{yU} – время срабатывания управляющего устройства,

$T_{OБ}$ – время срабатывания обрабатывающего блока, определяемое по времени исполнения самой длительной МО.

Расчетное значение тактовой частоты определяется величиной $F=1/T_{Oy}$.

Рабочая частота F_p выбирается из гостированного ряда частот $\{F\}$ при условии, что $F_p \leq 0,8F$.

Временная диаграмма работы ОУ с учетом реальных задержек в схемах этого устройства может быть правильно построена, если учесть следующие временные соотношения:

$\tau_1 \geq \tau_{выб}$, где $\tau_{выб}$ – максимальное время выборки слова из ПМП;

$\tau_2 \geq \tau_p$, где τ_p – время, необходимое для записи слова в РМК;

$\tau_3 \geq \max\{\tau_{ФСМО}, \tau_{ФА}\}$, где $\tau_{ФСМО}$ – время срабатывания ФСМО (схема этого устройства и время срабатывания зависит от используемого способа кодирования МО), $\tau_{ФА}$ – суммарное время срабатывания всех схем, включенных в ФА;

$\tau_4 = \max_k \{\tau_{МО_k}\}$, где $\tau_{МО_k}$ – время исполнения в ОБ k -ой микрооперации (в случае синхронизации с постоянным тактом);

$\tau_5 \geq \tau_{ФА}$; $\tau_6 \geq \tau_{АМК}$, где $\tau_{АМК} = \tau_p$ – время срабатывания регистра АМК (РАМК) в случае принудительной адресации МК, $\tau_{АМК} = \tau_{Сч}$ – время срабатывания счетчика АМК (СчАМК) при естественной адресации;

$T_{Oy} = \tau_1 + \tau_2 + \tau_3 + \tau_4 + \tau_5 + \tau_6 \geq \tau_n$, где τ_n – цикл выборки слова из ПЗУ, на основе которого построена ПМП.

Воспользовавшись справочной литературой получили такие значения

$$\tau_1 \geq \tau_{\text{выб}} = 65 \text{ нс}$$

$$\tau_2 \geq \tau_p = 40 \text{ нс}$$

$$\tau_3 \geq \max\{\tau_{\text{СМО}}, \tau_{\text{ФА}}\} = 35 \text{ нс}$$

Значение τ_4 выбирается в предположении, что самая длительна операция это операция $\text{СчАК} := \text{СчАК} + 2$, так как СчАК имеет 27 разрядов, следовательно состоит из 7-ми регистров K155ИЕ7,

$$\tau_4 = \max_k \{\tau_{\text{МО}_k}\} = 7 * 35 = 245 \text{ нс}$$

$$\tau_6 \geq \tau_{\text{АМК}} = 35 \text{ нс}$$

$$T_{\text{ОУ}} = \tau_1 + \tau_2 + \tau_3 + \tau_4 + \tau_6 = 65 + 40 + 35 + 245 + 35 = 420 \text{ нс}$$

Рассчитаем значение тактовой частоты $F = 1/T_{\text{ОУ}} = 0,0024 * 1000 \text{ МГц}$

Рабочая частота F_p выбирается из гостированного ряда частот $\{F\}$ при условии, что $F_p \leq 0,8F$.

$$F_p = 0,8 * 2,4 \text{ МГц} = 1,9 \text{ МГц}$$

По ГОСТу самая приближённая частота $F_p = 1 \text{ МГц}$

На рисунке 7.1 приведена временная диаграмма процесса исполнения МК

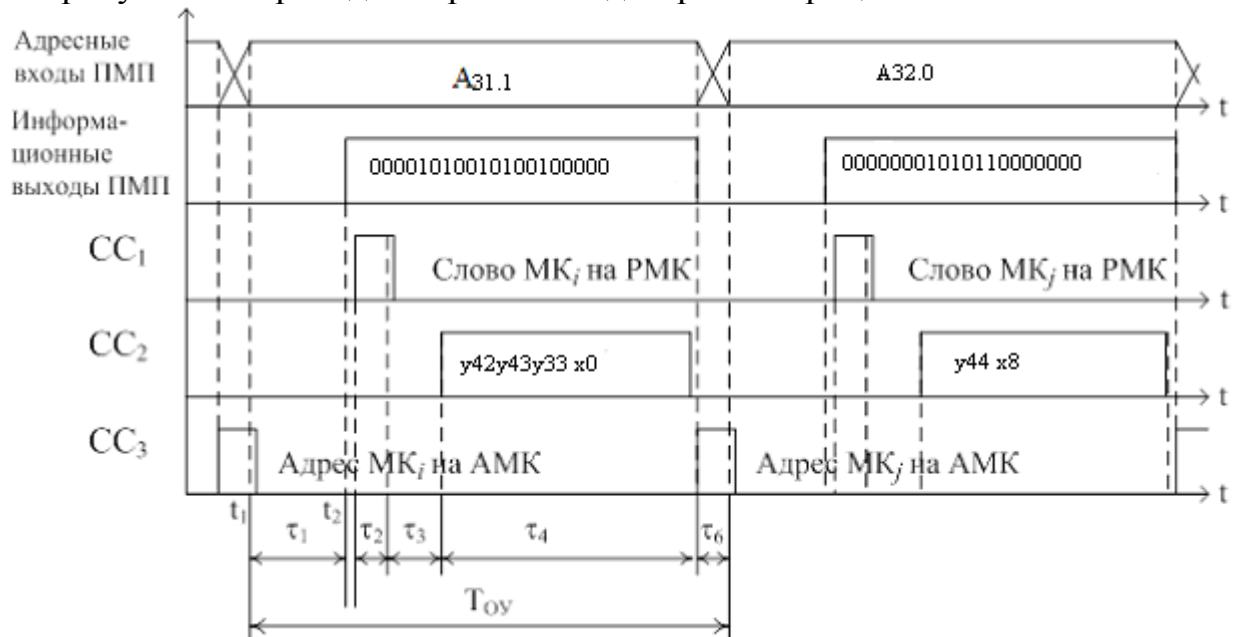


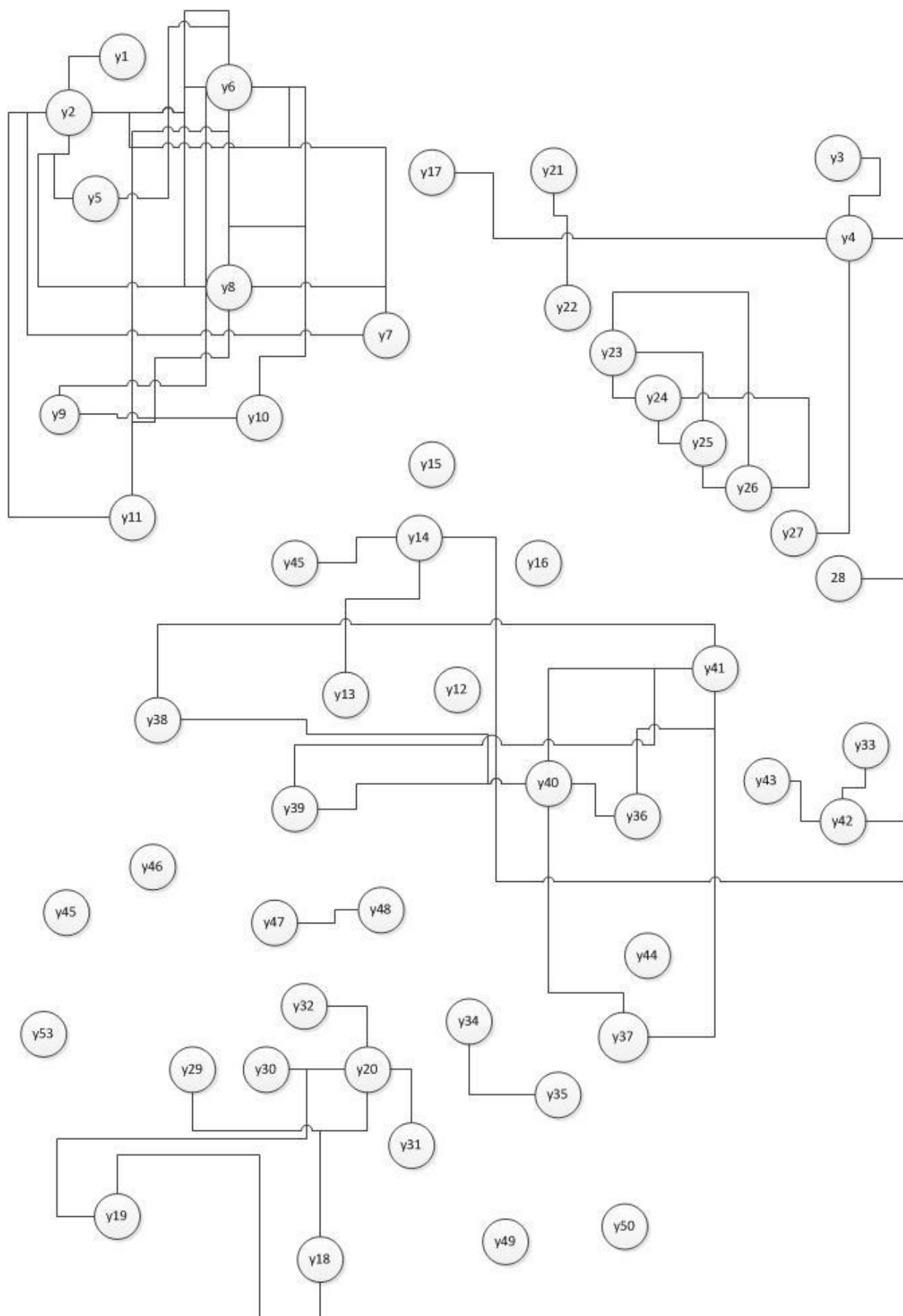
Рисунок 7.1 – Временная диаграмма исполнения МК ($\text{МК}_i \rightarrow \text{МК}_j$)

ЗАКЛЮЧЕНИЕ

В ходе курсового проектирования было спроектировано простейшее центральное обрабатывающее устройство, реализующее систему из пяти команд, и разработана его структурная схема. Для УА была построена микропрограмма его функционирования и спроектирована электрическая принципиальная схема. Были рассчитаны временные характеристики синтезированного устройства. Рабочая частота $F_p = 1$ МГц.

Это операционное устройство обеспечивает все пункты поставленной задачи на курсовое проектирование.

Приложение А. Граф совместимости МО



Приложение Б. Микропрограмма функционирования ЦОУ.

		Микропрограмма																									
		M1					M2				M3					X					A						
	МК	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A0.0	y0 x1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1
A0.1	-																										
A1.0	y0 x2	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
A1.1	y0 x3	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	1
A2.0	y3y4 x5	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1	0	1	0	0	0	1	1	0
A2.1	y1y2 x4	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
A3.0	y3y4 x5	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1	0
A3.1	y51 x0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1
A4.0	y3y4 x5	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1
A4.1	y0 x7	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	0	1	0	0	1
A5.0	y7y8y2y6 x23	1	1	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0	1	1	1	0	0	1	0	0	1
A5.1	y0 x5	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0	1
A6.0	y0 x2	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	1
A6.1	y0 x5	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	1	1	0
A7.0	y0 x6	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0
A7.1	y5y2y6 x4	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0
A8.0	y9y10y6 x23	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	1	0	0	0	1	0	0
A8.1	y8y11y2y6 x23	1	1	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	1	1	1	0	0	0	1	0	0
A9.0	y0 x13	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	1	0	1	0	0	1	0
A9.1	y12 x8	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
A10.0	y13y14 x0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1
A10.1	-																										
A11.0	y15 x9	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	1	0	0	1	0	0	1	1	0	0
A11.1	-																										
A12.0	y0 x10	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	0	0	1	1	0	1
A12.1	y53 x0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0	1	0	1	0
A13.0	y16 x0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0
A13.1	y17y4 x5	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	1	0
A14.0	y18y19y20 x0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
A14.1	y0 x5	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0
A15.0	y21y22 x11	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	1	0	1	1	0	1	0	0	0	0
A15.1	-																										
A16.0	y23y34y25y26 x12	0	0	1	0	0	0	0	1	0	0	0	0	1	1	1	0	1	1	0	0	0	1	0	0	0	1
A16.1	y0x11	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	0	1	0	0	0	0
A17.0	-																										
A17.1	y16 x0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0
A18.0	y0 x18	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	1	1	1	1	1
A18.1	y0 x14	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	0	1	0	0	1	1
A19.0	y16 x0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0
A19.1	y27y4 x5	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	0
A20.0	y0 x15	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	1	0	1	0	1	0	1
A20.1	y0 x5	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	1	0	1
A21.0	y28v4 x5	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	1	0	1	0	1	1	0

A21.1	y53 x0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0
A22.0	y0 x16	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	1
A22.1	-																								
A23.0	y0 x17	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	0
A23.1	y0 x17	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	1
A24.0	y29y20 x0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
A24.1	y30y20 x0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
A25.0	y31y20 x0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0
A25.1	y32y20 x0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
A26.0	y34y35 x11	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	1	1
A26.1	-																								
A27.0	y0x16	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0
A27.1	y0x11	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	0	1	1	1
A28.0	y0x17	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0	1	1	0
A28.1	y0x17	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0	1	1	0
A29.0	y36y40y41 x0	0	0	0	1	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
A29.1	y37y40y41 x0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
A30.0	y38y40y41 x0	0	0	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
A30.1	y39y40y41 x0	0	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
A31.0	y0 x19	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	1	1	0	0	0
A31.1	y42y43y33 x0	0	0	0	0	1	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0	0
A32.0	y44 x8	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0	1	0	0	1
A32.1	-																								
A33.0	y13y14 x0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0
A33.1	y0x9	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	1	0	1	1
A34.0	y15 x23	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	1	0	1	1	1	1	0	0	1
A34.1	-																								
A35.0	y45 x0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
A35.1	y53 x0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0	1	0
A36.0	y0x20	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	1	0	0	0
A36.1	y42y14 x0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	1	1
A37.0	y45y14 x0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0	0	0	1	0	1	0
A37.1	-																								
A38.0	y47 x5	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0
A38.1	-																								
A39.0	-																								
A39.1	y0 x5	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	1	0	1	1
A40.0	y0 x22	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	1	0	1	0	1	0
A40.1	y47y48 x21	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	1	0	1	1	0	0	1
A41.0	-																								
A41.1	y0 x21	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	1	1	0	0	1
A42.0	y49 x0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0
A42.1	y50 x0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1