**3. ГСА функционирования ЦОУ**

ГСА функционирования ЦОУ приведена на чертеже 2011.М41д.21.01.

Разработке подлежат алгоритмы выполнения пяти операций, определенных вариантом задания, и алгоритм выборки команды из ОП. Взаимосвязь указанных алгоритмов отражена на упрощенной граф-схеме алгоритма функционирования ЦОУ (рисунок 3.1). Если код выбранной на РК команды не совпадает с кодом ни одной из реализованных команд, устанавливается флаг прерывания K (резервная команда).



*Рис 3.1 – Схема алгоритма функционирования ЦОУ (здесь ЗР – команда загрузки регистра; МП – команда передачи байта из МВВ в процессор; СлПТ – команда сложения с плавающей точкой; СдвВпр – команда двойного сдвига вправо; УПМ – команда условного перехода по маске)*

Опишем подробней алгоритмы, входящие в состав общей ГСА:

1. **Алгоритм выборки команды** *(на чертеже представлен блоками 1-15):*

1. Если установлен ТП, следовательно, была выполнена команда ПУ и значение СчАК может быть любым, поэтому проверяется, является ли адрес кратным 2 (однобайтных команд быть не может). Если указан адрес одного байта – нарушение спецификации. Иначе, читаем 4 байта из ОП.

1.1. Если был указан адрес 4 байт, проверяем, является ли команда 4 байтной (проверяются первые два бита Порта Д, а следовательно первые два бита КОП). Если команда 4-байтная, загружаем 4 байта в РК и прибавляем к СчАК 4 (Адрес в СчАК останется кратным 4), иначе загружаем в РК 2 байта, прибавляем к СчАК 2 и адрес в СчАК не будет кратен 4, поэтому запоминаем младшие 2 байта в БР. Сбрасываем ТП и переходим к выполнению команды.

1.2. Если был указан адрес 2 байт (в СчАК после команды ПУ) прибавляем к СчАК 2 (в СчАК теперь будет адрес кратный 4), сбрасываем ТП и проверяем, является ли команда двухбайтной(проверяются первые два бита КОП, а следовательно первые два бита РК). Если команда 2 байтная, переходим к выполнению команды, иначе читаем из ОП данные. Так как старшие 2 байта команды у нас уже хранятся в РК, то из Порта Д выбираются старшие 2 байта и помещаются в младшие 2 байта РК, а младшие 2 байта Порта Д сохраняются в буферный регистр. СчАК увеличивается на 2, а следовательно перестает быть кратным 4.

2. Если ТП не был установлен, то в СчАК хранится адрес либо 2 либо 4 байт.

2.1. Если в СчАК адрес 2 байт, то старшая часть РК хранится в БР. Сохраняем в старшие 2 байта РК содержимое БР и увеличиваем СчАК на 2( он становится кратным 4). Дальше выполняем те же действия, что и в пункте 2 б) начиная с проверки длины команды.

2.2. Если же адрес в СчАК кратен 4, то выполняем действия, указанные в пункте 1, без проверки на нарушение спецификации.

3. Конец

1. **Алгоритм команды чтения из ОП в РП** *(на чертеже представлен блоками 17-21):*
2. Выполняем чтение адреса ОП, заданного косвенно через РОН: для этого загружаем в РАРП адрес РОН (в «младшей» части универсальной РП – первый бит адреса – 0) и выполняем чтение из регистровой памяти.
3. Если считанный адрес ОП кратен 4, то переход к п.3, иначе прерывание «Спецификация».
4. Загружаем полученный адрес в Порт А и выполняем операцию чтения ОП (по ее окончании – слово из ОП загрузится в Порт Д).
5. Выполняем запись слова ОП в РОН, указанный в первом операнде: в РАРП загружаем адрес РОН, в РСРП – слово из Порта Д и выполняем операцию записи РП.
6. Конец.
7. **Алгоритм команды пересылки данных из порта ввода-вывода в процессор** *(на чертеже представлен блоками 23-24):*
8. Загружаем адрес (номер) устройства ввода-вывода, указанный в коде команды, в порт А.
9. Инициируем операцию пересылки данных из порта ввода-вывода в процессор (операция длительная, поэтому нужно дождаться окончания ее выполнения).
10. Конец.
11. **Алгоритм команды сложения чисел с плавающей точкой** *(на чертеже представлен блоками 26-38)***:**
12. Если адрес РПТ (первый операнд) кратен 2 (64-разрядные числа с плавающей точкой должны храниться в паре РПТ), то переход к п.2, иначе – прерывание «Спецификация».
13. Считываем адрес второго операнда из ОП, который косвенно задан в РОН: в РАРП загружаем адрес РОН и инициируем операцию чтения РП.
14. Если считанный адрес кратен 8 (т.к. 64-разрядное число), то переход к п.4., иначе - прерывание «Спецификация».
15. Выполняем чтение младшей части второго операнда из ОП: загружаем в Порт А адрес, считанный из РОН, и выполняем операцию чтения ОП.
16. В младшую часть Регистра 2 АЛУ загружаем младшую часть второго операнда. Выполняем чтение младшей части первого операнда из первого РПТ регистровой пары. Инкрементируем адрес на Порте А и выполняем операцию чтения ОП (чтение старшей части второго операнда).
17. В младшую часть Регистра 1 АЛУ загружаем младшую часть первого операнда. В старшую часть Регистра 2 АЛУ загружаем старшую часть второго операнда. Выполняем чтение старшей части первого операнда из второго РПТ регистровой пары.
18. В старшую часть Регистра 1 АЛУ загружаем старшую часть первого операнда. Инициируем операцию АЛУ сложения чисел с плавающей точкой.
19. Выполняем запись старшей части результата (Р1АЛУ) по текущему адресу РП (второй РПТ пары). В регистр признака результата загружаем кодированное значение признаков с шины АЛУ. В регистр флагов загружаем значение флагов с шины АЛУ.
20. Выполняем запись младшей части результата (Р1АЛУ) по предварительно уменьшенному на 1 адресу РП (первый РПТ пары).
21. Конец.
22. **Алгоритм команды двойного сдвига вправо** *(на чертеже представлен блоками 40-53)***:**
23. Если укороченный адрес, указанный в коде команды, кратен 4, то переход к п.2, иначе – прерывание «Спецификация».
24. В Порт А загружаем укороченный адрес в младшей части памяти (старшие разряды заполняем 0-ми) и инициируем операцию чтения ОП – получаем адрес первого операнда в ОП.
25. Если адрес кратен 8 (т.к. 64-битный вектор), переход к п.4, иначе – прерывание «Спецификация».
26. Выполняем загрузку считанного из ОП косвенного адреса в Порт А и инициируем операцию чтения ОП. В Регистр 2 АЛУ загружаем второй операнд (константу сдвига), непосредственно заданный в коде команды.
27. В Регистр 1 АЛУ загружаем младшую часть первого операнда, полученную с Порта Д. Инкрементируем значение адреса на Порте А и выполняем чтение ОП – чтение старшей части первого операнда.
28. Догружаем старшую часть первого операнда в Регистр 1 АЛУ и инициируем операцию двойного сдвига вправо.
29. В два приема, подобно чтению первого операнда из ОП, выполняем запись результата в ОП.
30. Конец.
31. **Алгоритм команды условного перехода по маске** *(на чертеже представлен блоками 55-60)***:**
32. Если маска содержит все нули, то переход к п.6.
33. Если адрес РОН, косвенно задающего адрес перехода, содержит все нули, то переход к п.6.
34. Если маска содержит все единицы, то переход к п.5.
35. Если значение регистра признака не соответствует маске (признак в регистре закодирован, поэтому требуется дешифрирования), то переход к п.6.
36. Считать и занести в СчАК адрес перехода, указанный в РОН. Установить ТП.
37. Конец.