

4053/88

88-317

Лайх Х.ндр.



ЛВЭ

ин

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ

Б 2-11-88-317

ДЕПОНИРОВАННАЯ ПУБЛИКАЦИЯ

Дубна 19 88

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ
Лаборатория вычислительной техники и автоматизации

Б2 - 11-88-317

Х.Лайх, В.И.Приходько, К.Фогт

АППАРАТНО-ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ОТЛАДКИ СИСТЕМ
НА МИКРОПРОЦЕССОРНЫХ СЕКЦИЯХ

Рукопись поступила
в издательский отдел

11.05.88 г.

Дубна, 1987

Объединенный институт
ядерных исследований
БИБЛИОТЕКА

На базе микропроцессорных секций строятся мощные процессоры. Использование принципа микропрограммирования обеспечивает большую гибкость для удовлетворения специальных требований в конкретных применениях. Набор команд в этих процессорах определяет сам разработчик. Система на микропроцессорных секциях в основном состоит из двух главных блоков — из схемы управления с микропрограммной памятью и арифметико-логического устройства (см. рис. I) /1/, для реализации которых промышленность предлагает специальные серии интегральных схем.

В то время, как для поддержки разработки и отладки систем на основе универсальных микропроцессоров существует большое разнообразие средств, аналогичные средства для систем на микропроцессорных секциях мало распространены. На этапе разработки таких систем требуются, как минимум, микроассемблер, программа имитации микропрограмм и устройство для программирования быстрых ПЗУ. При этом следует иметь в виду, что требования тестирования и отладки должны учитываться уже на самых ранних стадиях разработки.

В данной работе предлагается один из возможных путей решения проблемы отладки систем на микропроцессорных секциях. В нашем распоряжении имелся комплект аппаратно-программных средств ИНТЕЛЛЕК MDS800 /2/, предназначенный для разработки и отладки 8-разрядных микро-ЭВМ на базе процессоров Интел 8080. В состав этого комплекта входит также внутрисхемный эмулятор ICE-80, включающий в себя достаточно сложную микропроцессорную аппаратуру и программное обеспечение, имитирующее выполнение программы пользователя в подлежащей разработке микропроцессорной системе. Таким образом, для разработки и отладки микро-ЭВМ конкретного назначения используются все ресурсы системы ИНТЕЛЛЕК и возможности программы-отладчика. На базе этого оборудования была создана система имитации микропрограмм.

Для тестирования и отладки систем на микропроцессорных секциях потребовалось создать следующие дополнительные возможности:

- реализация пошагового режима;
- считывание/запись из/во внутренние регистры процессора;
- замена микропрограммной памяти (PROM) частью основной памяти микро-ЭВМ (RAM) .

Для удовлетворения этих требований и полного использования возможностей отладочного комплекса ИНТЕЛЛЕК проблема отладки аппаратуры на микропроцессорных секциях была переведена на уровень отладки обычных микро-ЭВМ, т.е. секционный процессор управляется 8-разрядным микропроцессором, который в свою очередь, имитируется внутрисхемным эмулятором.

Имитатор микропрограмм состоит из двух основных частей - управляющей программы и адаптера, который предназначен для аппаратного согласования микро-ЭВМ с устройством управления системой на микропроцессорных секциях. Адаптер содержит формирователи, регистры и декодеры (см. рис.2 и рис.5).

На рис.3 представлен алгоритм работы управляющей программы. Программа написана на языке PL/M-80 и занимает около 4 Кбайт. Адресное пространство распределяется в соответствии с рис.4. Сама управляющая программа загружается начиная с нулевого адреса, а подлежащие проверке микропрограммы расположены в области памяти от 2000H по 5FFFH . Выше этой области определены некоторые управляющие адреса, которые декодируются в адаптере. С их помощью образуются сигналы выбора соответствующих регистров и формирователей.

Для работы с системой были реализованы следующие макрокоманды:

1. Для выполнения микрокоманд:

.G [START] [,BREAKPOINT 1] [BREAKPOINT2]

GO - выполнение команд от начальной точки до останова 1 или 2;

.S [START]

STEP - выполнение команд от начальной точки в пошаговом режиме;

.C [START] [,BREAKPOINT]

CYCLE - петля от начальной точки до останова.

2. Для считывания внутренних регистров и буфера циклов:

.DR

Вывод на дисплей содержимого всех процессорных регистров и счетчика;

.DC

Вывод на дисплей адресов выполняемых команд (последний I28).

3. Для загрузки внутренних регистров:

.WR[RNR, DATA)

Запись данных в регистр RNR ;

.WQ[DATA]

Запись данных в регистр Q .

Перед началом отладки управляющая и тестовая программы с помощью ISE загружаются в ИНТЕЛЛЕК (по правилам, изложенным в руководстве по эксплуатации ISE-80) и имитатор выдает запрос на ввод команды.

Если требуется выполнить микрокоманду, то вводится соответствующая команда (GO , STEP , CYCLE). После проверки правильности ввода, через порт с адресом 6008H (MI \times ADR \times IN) считывается приложенный к входу микропрограммной памяти адрес. Если в введенной команде указан стартовый адрес, то он перед считыванием с помощью микрокоманды перехода загружается в устройство управления последовательностью микрокоманд, так что именно этот адрес появляется на входе памяти. Из считанного адреса вычисляется текущий адрес, под которым в RAM микро-ЭВМ запоминается соответствующая микрокоманда. Так как в нашем конкретном случае ширина микрокоманды составляет 40 разрядов, то в регистр микрокоманд адаптера (OUT PORT 6008H по 6004H) подряд загружается 5 байтов. Указанный регистр непосредственно связан с панельками для интегральных схем микропрограммной памяти процессора на микропроцессорных секциях. Таким образом микрокод подается прямо на вход конвейерного регистра. Тактовый импульс передает его в регистр и запускает выполнение команды. Для GO -команды этот процесс повторяется до достижения адреса останова. Во время выполнения, считанные устройством управления последовательностью микрокоманд, адреса запоминаются в буфере циклов; это позволяет проверить правильность следования адресов микрокоманд.

Команду CYCLE полезно применять для поиска аппаратных ошибок. Микрокоманды циклически выполняются с указанного начального адреса и до конечного, а с помощью осциллографа или логического анализатора можно исследовать последовательность сигналов. Для проверки правильности выполнения микроопераций существует возможность считывания регистров арифметико-логического устройства. Для

этого в конвейерный регистр загружается та микрокоманда, которая передает содержимое регистра на шины данных. Через управляющие порты с адресами 6008H по 600AH (DAT & BUF) побайтно считываются 24 разряда шины данных. Это повторяется 17 раз, затем содержимое всех регистров и счетчика из промежуточного буфера выдается на дисплей. Подобным же образом с помощью команд WR и WQ происходит загрузка в регистры нужной информации.

После того, как с помощью имитатора микропрограмм проверена правильность выполнения команд, микропрограмма записывается, начиная с нулевого адреса, в соответствующие ПЗУ, а система на микропроцессорных секциях отключается от адаптера.

Для подготовки микрокода микропрограмм применяется микроассемблер /3/. Способы определения микрокоманд подробно описываются в /4/. Микроассемблер генерирует объектный код, который нельзя обрабатывать на отладочном комплексе. Поэтому объектный код с помощью специальной программы WINEX преобразуется в HEX-файл и локализуется по заданным адресам. После этого программа HEXOBJ транслирует этот файл в готовый к выполнению объектный код.

Представленный имитатор микропрограмм уже доказал свою эффективность во время отладки и тестирования специализированного процессора на микропроцессорных секциях /4/.

Литература

1. Мик Дж., Брик Дж. Проектирование микропроцессорных устройств с разрядномодульной организацией, в 2-х книгах. Мир, М., 1984.
2. INTELLEC MDS OPERATOR'S MANUAL; INTEL CORP., DOC. No 98-132, 1976 .
PL/M-80PROGRAMMING MANUAL; INTEL CORP., DOC. No 98-268, 1976
ISIS-II ICE-80 OPERATOR'S MANUAL; INTEL CORP., DOC No98-185 C, 1976.
3. Rudalics M. The Mica/1-Microassembler .
JINR E11-82-82. Dubna, 1982.
4. Leich A. et al. JINR E11-84-364. Dubna, 1984.

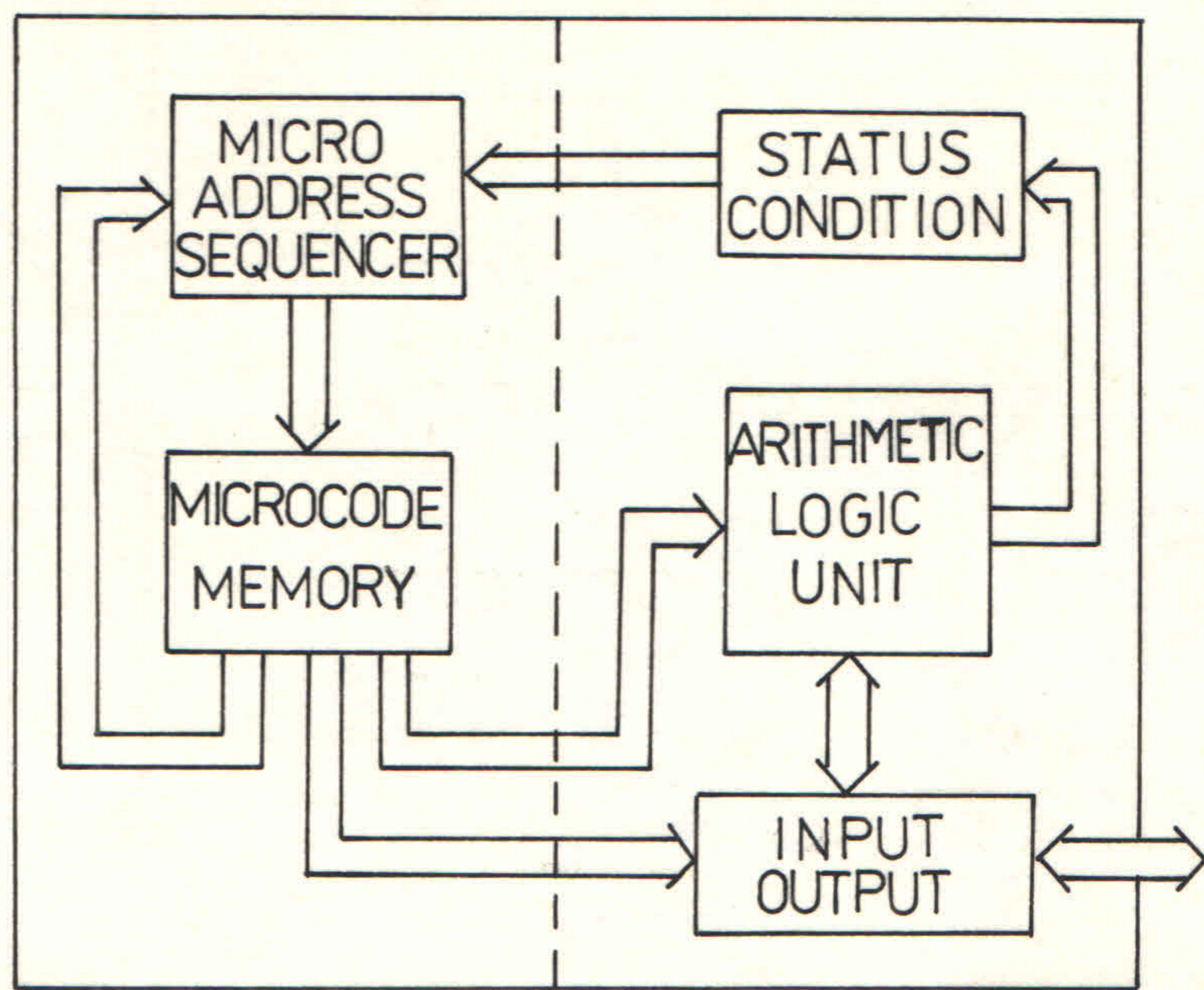


Рис.1. Структура микропроцессорной системы на процессорных секциях

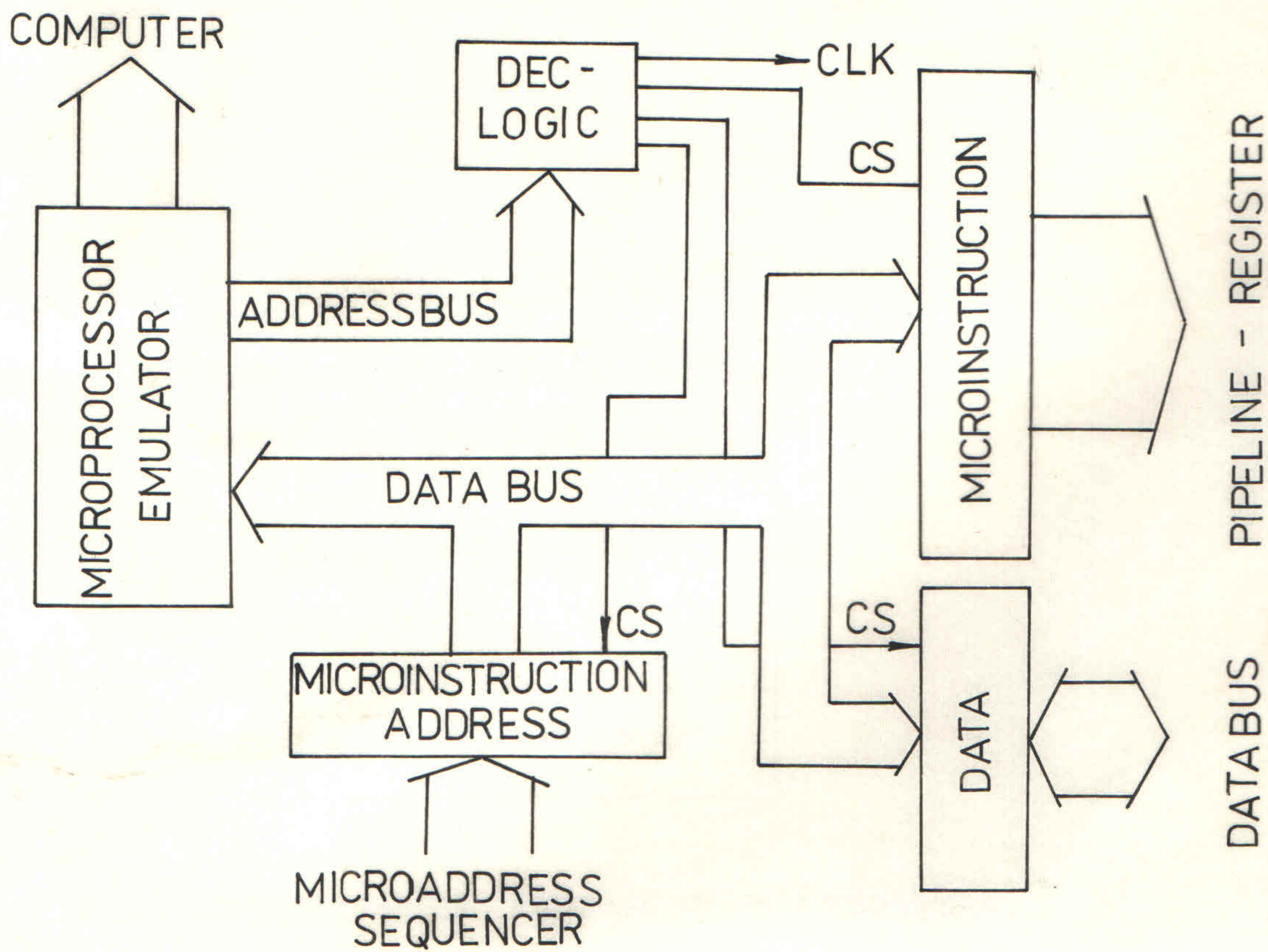


Рис.2. Функциональная схема адаптера

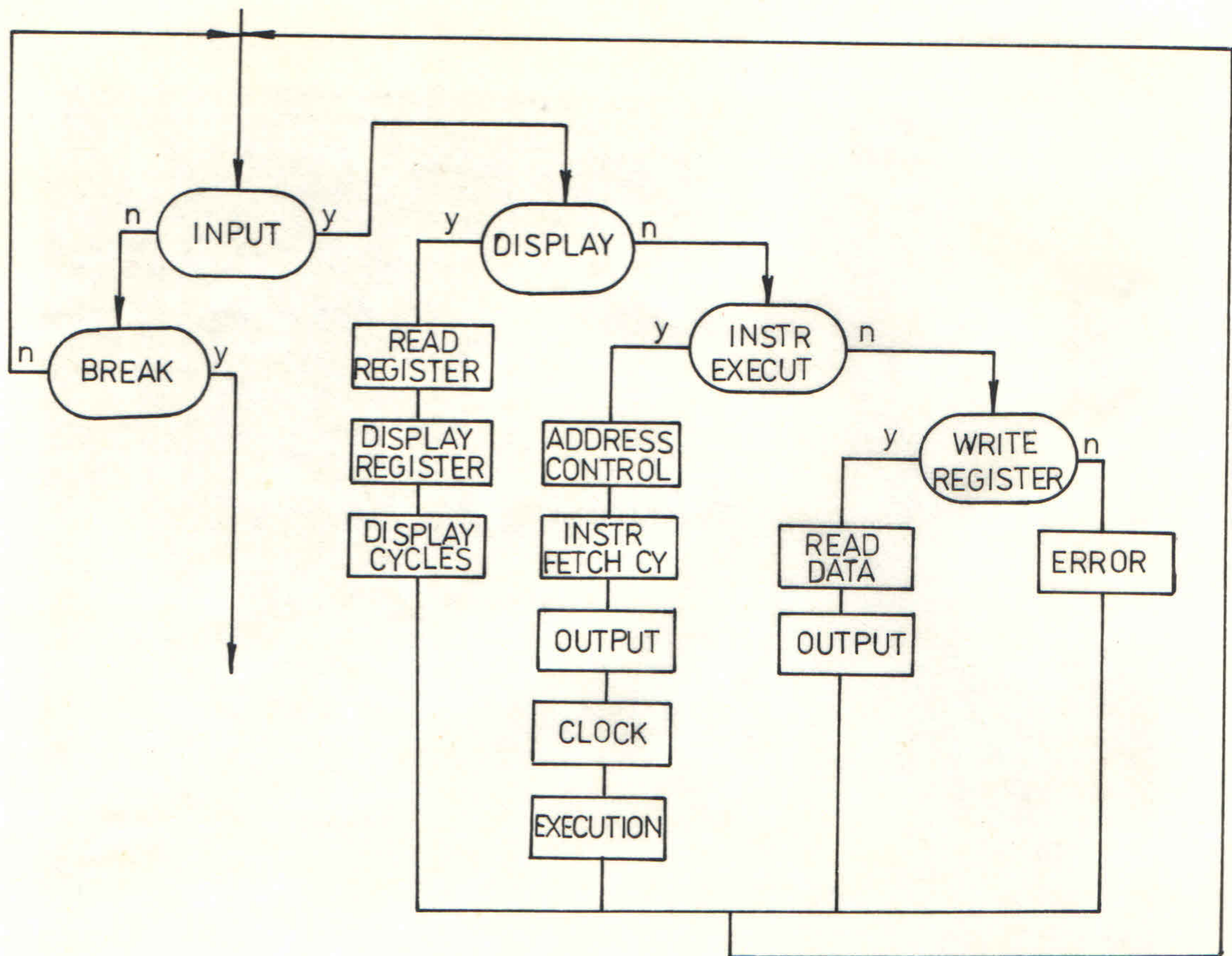


Рис.3. Алгоритм управляющей программы

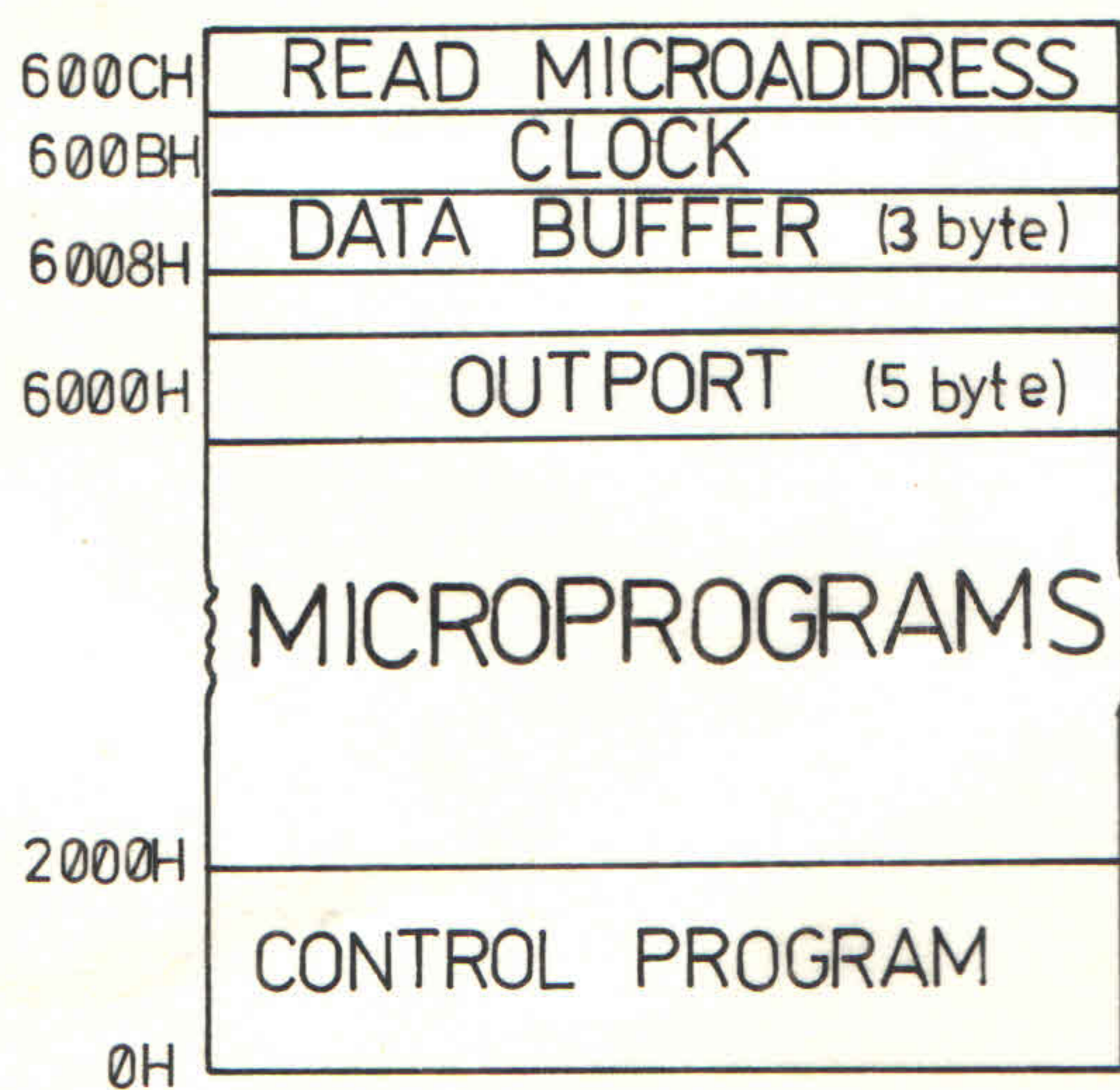


Рис.4. Распределение адресного пространства

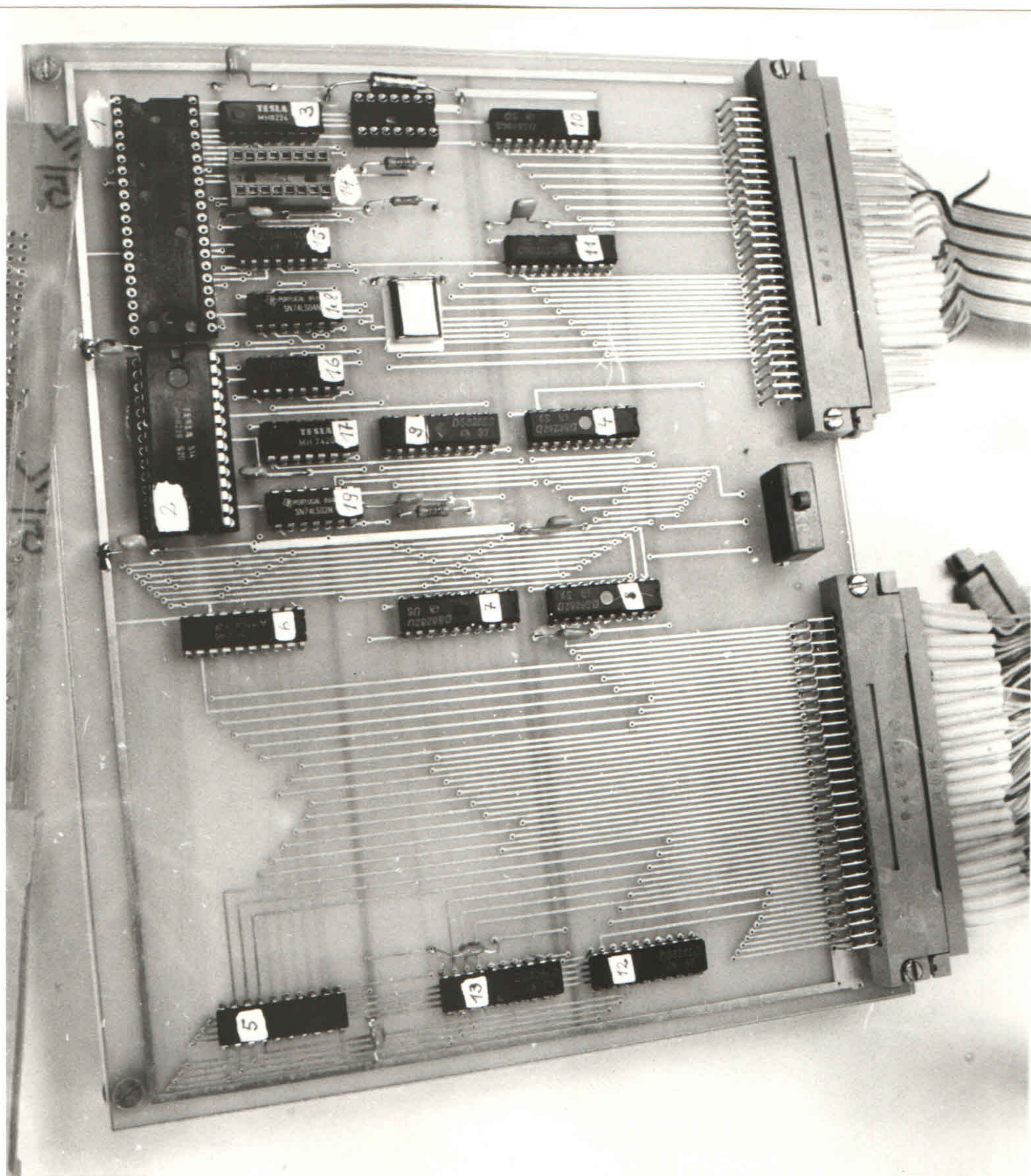


Рис.5. Общий вид платы адаптера