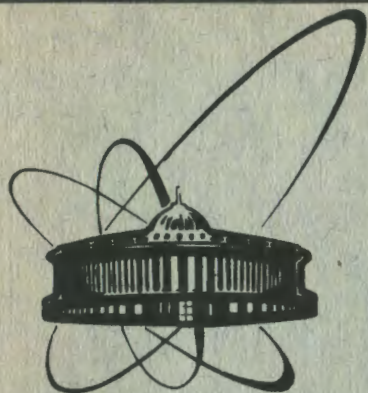


91-43



сообщения
объединенного
института
ядерных
исследований
дубна

P11-91-43

И. А. Емелин, В. М. Кадыков, Ф. В. Левчановский,
М. Ю. Попов, А. П. Сапожников, Т. Ф. Сапожникова,
И. Н. Силин

АРХИТЕКТУРНЫЕ ОСОБЕННОСТИ МКБ-8601,
ИНТЕГРАЛЬНОЙ ЭВМ РЯДА БЭСМ-6

1991

С целью сохранения программного обеспечения, накопленного за длительный период работы, в ЛВТА ОИЯИ создана универсальная ЭВМ, программно совместимая с БЭСМ-6 /1/. Путь аппаратной модернизации ЭВМ БЭСМ-6 представляется экономически более целесообразным, чем перенос и переработка огромного объема оригинального программного обеспечения на другие типы ЭВМ. Устранены также присущие ЭВМ БЭСМ-6 основные недостатки: короткий адрес, малый порядок чисел с плавающей запятой, недостаточный объем оперативной памяти.

В разработанной ЭВМ с точки зрения выполнения команд предусматривается возможность ее работы в следующих трех режимах:

- режим исполнения команд старой БЭСМ-6;
- режим исполнения команд новой машины МКБ-8601, включая аналогии команд БЭСМ-6;
- режим исполнения расширенного списка команд (команд МКБ-8601 и команд ЭЛЬБРУС-Б).

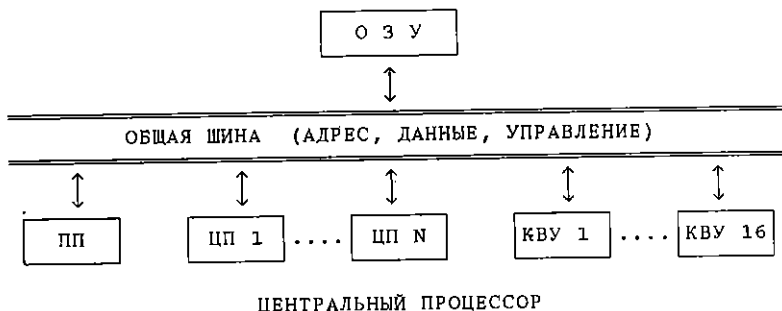
Первый режим позволяет работать с программами, написанными для БЭСМ-6. В третьем режиме возможно применение разработанной ЭВМ в качестве программно совместимой младшей модели вычислительной системы ЭЛЬБРУС-Б. В список новых команд вводятся команды, необходимые для работы с целочисленной арифметикой, команды, обеспечивающие обработку отдельных полей 64-разрядного слова, команды с непосредственным операндом и другие, способствующие улучшению рабочих характеристик ЭВМ и ее операционной системы.

В архитектуру процессора МКБ-8601 введен тег, что позволяет осуществлять пословную защиту памяти от несанкционированного доступа и организовать удобную отладку программ пользователем. Предусмотрена также возможность работы в многопроцессорном и мультипрограммном режимах, введены средства для работы в режиме реального времени.

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ИНТЕГРАЛЬНОЙ Э В М

ЭВМ МКБ-8601 является микропрограммно управляемой машиной общего назначения с модульной организацией. Основными модулями, входящими в состав машины, являются: оперативное запоминающее устройство (ОЗУ), центральный процессор (ЦП) (их может быть несколько), контроллеры для связи с внешними

устройствами (КВУ), а также пультовой процессор (ПП). Связь между ними осуществляется через общую шину.



Центральный процессор предназначен для исполнения команд, записанных в ОЗУ и состоит из:

- блока микропроцессорных секций, содержащего АЛУ и регистровое запоминающее устройство;
- блока микропрограммного управления, состоящего из преобразователей начального адреса, схемы управления адресом микрокоманды, управляющей памяти микропрограмм и конвейерного регистра;
- ППЗУ констант;
- арифметических блоков умножения/деления, однокантного сдвига на N разрядов и поиска левой единицы;
- блоков обмена информацией (БОИ) с общей шиной (64 разряда для команд и данных и 8 разрядов для тега);
- ОЗУ модификаторов;
- стандартизатора команд, преобразующего команды БЭСМ-6 к общему виду команд МКБ-8601 с 8-разрядным кодом операции, и схемы размножения знака адресной части команды;
- регистров исполнительного адреса, номера группы модификаторов (РНГ), номера процесса (РНП);
- регистра и таймера счетного времени процесса;
- блока преобразования виртуальных адресов в физические;
- блока обработки внешних и внутренних прерываний;
- блока учета активности страниц памяти;
- арбитра шины;
- блока связи с пультовым процессором, состоящего из ОЗУ обмена, микропроцессора со своими ППЗУ и ОЗУ для обслуживания линий связи с пультовым процессором и загрузки микропрограмм;
- блока останова работы программы по заданному адресу;
- микропрограммно-управляемого тактового генератора.

ОПЕРАТИВНАЯ ПАМЯТЬ ЭВМ

Оперативная память ЭВМ состоит из блока управления и матрицы запоминающих устройств. Максимальная емкость оперативного запоминающего устройства - 1 миллион 72 - разрядных слов (9 Мбайт). Для повышения отказоустойчивости работы ОЗУ в блок управления встроены генератор кода Хемминга.

Блок управления ОЗУ выполняет следующие операции:

- регенерация памяти;
- чтение слова с коррекцией однократной ошибки, перезапись исправленного слова и фиксации ошибок кратности 2 и более;
- запись слова с проверкой и без проверки бита тега "запрещенная запись" (при проверке производится предварительное считывание слова);
- чтение семафорного слова (чтение-модификация-запись);
- чтение и запись в режиме блочной передачи;
- тестирование генератора Хемминга.

ПУЛЬТ - ПРОЦЕССОР

Пульт-процессор представляет из себя микроЭВМ с собственной памятью, возможностью доступа к основной памяти и ко всем регистрам центрального процессора. Пульт-процессор выполняет следующие функции:

- загрузка памяти микропрограмм ЦП;
- имитация инженерного пульта машины (считывание и запись содержимого регистров ЦП, запуск микропрограммных тестов);
- ввод двоичных программ в общую память;
- запуск диспетчера или тестов в ЦП;
- "пульт-процессор" отладка программ в ЦП (пошаговое выполнение команд, останов по адресу);
- обслуживание операторского терминала.

Пульт-процессор может прерывать ЦП или приостанавливать его работу. В свою очередь, деятельность самого ЦП может быть прервана по одной из следующих причин:

- по запросу от ЦП;
- по аварии в ЦП;
- по сбою в памяти;
- по инициативе оператора.

В связи с разнородностью исполняемых функций обмен информацией ЦП с различными узлами машины осуществляется двумя способами:

а) через последовательные линии связи - с несколькими центральными процессорами для доступа к регистрам и памяти ЦП, памяти микропрограмм, для организации пультового останова;

б) через общую шину данных - для обмена с общей памятью и внешними устройствами, подключенными к общей шине.

ОБЩАЯ ШИНА

Обмен информацией между блоками ЭВМ (процессор, пультовой процессор, общая память, контроллеры внешних устройств) осуществляется через общую шину асинхронно-синхронного типа. Все блоки ЭВМ могут быть как ведущими, так и ведомыми, кроме блока общей памяти, который всегда ведомый. Асинхронизм обмена поддерживается сигналами BRQ (запрос шины), BUSY (шина занята), BRDC, BWRC, BRMWC (команды чтения, записи, чтения-модификации-записи) и ACK (подтверждение обмена). Общее число проводников шины - 100.

ЗАХВАТ ШИНЫ. Для захвата шины в ЭВМ реализован децентрализованный арбитраж. Основная идея арбитража заключается в следующем. При незанятой шине каждый из требующих ее блоков выдает сигналы кода своего приоритета BRQ0:BRQ7 и одновременно считывает код приоритета, установившийся на шине. Если код приоритета, считанный с шины, равен коду собственного приоритета, то блок захватывает шину, в противном случае ждет следующего цикла захвата. Блок, получивший шину, устанавливает сигнал BUSY и начинает обмен. Коды приоритетов BRQ0:BRQ7 на общей шине логически суммируются (проводное ИЛИ) и имеют следующий вид:

BRQ7 BRQ1 BRQ0

0: 0 0 0 0 0 0 0 0 - самый высокий приоритет

6: 0 1 1 1 1 1 1 1
7: 1 1 1 1 1 1 1 1 - самый низкий приоритет.

Такое решение позволяет, в отличие от известных методов, захватывать шину за один такт синхросигнала.

ПРОЦЕСС ОБМЕНА. С целью уменьшения количества проводников общей шины 20-разрядная адресная шина (A0:A19) совмещена с шиной данных, поэтому процесс обмена происходит в две фазы: фазы передачи адреса и фазы передачи данных.

При обращении к блоку общей памяти возможны следующие ненормальные ситуации:

- запись данных в выбранную ячейку ОЗУ запрещена;
- блок коррекции ошибок (генератор кода Хемминга) обнаружил двойную или многократную ошибку;
- обращение к физически несуществующей матрице памяти.

Для привязки таких случаев к конкретному активному ведущему блоку (процессору) каждая ситуация сопровождается своим сигналом на общей шине, которые фиксируются на триггерах схемы-арбитра этого процессора передним фронтом сигнала АСК и поступают как внутренние прерывания на его блок прерываний.

ОБРАБОТКА ПРЕРЫВАНИЙ. Для экономии проводников общей шины прерывания от контроллеров внешних устройств также передаются по шине данных (D0:D63) в виде унитарного кода. Все прерывания конкретного контроллера внешнего устройства логически суммируются и поступают на шину IRQ (запрос прерывания), на которой могут также логически суммироваться запросы на прерывание и от других контроллеров. Реагируя на сигнал IRQ, процессор выдает, предварительно захватив общую шину, сигнал INTA (подтверждение прерывания), который совместно с BRDC открывает шинные формирователи для выдачи активных прерываний на шину данных. Для сброса активных прерываний от одного или нескольких контроллеров одновременно, процессор производит выдачу принятых прерываний на шину данных, сопровождая их сигналами INTA и BWRС.

ОСНОВНЫЕ ФУНКЦИИ ПРОЦЕССОРА МКБ-8601

ФОРМИРОВАНИЕ ИСПОЛНИТЕЛЬНЫХ АДРЕСОВ. У каждой программы, исполняющейся в машине, имеется 15 индекс-регистров или модификаторов (М), предназначенных для модификации адреса, указанного в команде. Разрядность индексных регистров - 32. Модификация выполняется сложением содержимого указанного в команде индекс-регистра с адресной частью команды. В режиме эмуляции (РЭ) БЭСМ-6 из адресной части команды используются только младшие 15 разрядов, в новом режиме - 20 разрядов. Старший разряд адресной части команды перед модификацией размножается до 32 разряда.

Имеются специальные команды UTC и WTC, позволяющие осуществлять модификацию адреса следующей за ними команды. В этих случаях для модификации используются соответственно исполнительный адрес или содержимое слова по исполнительному адресу команд UTC и WTC, которые заносятся в специальный

регистр (С) с номером 16. Одновременно устанавливается признак изменения адреса (ПИА) в регистре режимов (РР).

Исполнительный адрес большинства команд формируется следующим образом.

Адрес типа U:

$$U = N + M \text{ (ПИА=0)} \quad \text{или} \quad U = N + M + C \text{ (ПИА=1)}.$$

Адрес типа V:

$$V = N \text{ (ПИА=0)} \quad \text{или} \quad V = N + C \text{ (ПИА=1)},$$

где N - адресная часть команды, C - содержимое 16-ого индекс-регистра.

Значение 16-ого регистра добавляется к исполнительному адресу только при активном ПИА. Все команды, кроме UTC и WTC, гасят этот признак. С помощью команд UTC и WTC может осуществляться модификация любой кратности.

МАГАЗИННОЕ ОБРАЩЕНИЕ. В машине обеспечена возможность адресации к любому участку оперативной памяти по принципу "магазина". Индексный регистр с номером 15 является счетчиком магазина и содержит адрес первого свободного слова этой области памяти. Обращение к памяти "магазинным способом" выполняется во всех командах записи или чтения из памяти целого слова, если в разрядах номера индекс-регистра указан 15-ый регистр и исполнительный адрес команды (адрес типа V) содержит нулевой код. Иначе обращение к ОЗУ происходит по адресу типа U без изменения счетчика магазина.

Кроме того, существуют специальные магазинные команды, при выполнении которых собственно операции предшествует запись в "магазинную память" содержимого сумматора (XTS, ITS), или операция сопровождается последующим чтением из "магазинной памяти" (STX, STI, UZAS, UIAS). Содержимое индекс-регистра и адресной части в такой команде определяют адрес самой операции и не влияют на обращение к "магазину".

ФОРМАТЫ ДАННЫХ

Общий формат данных имеет следующий вид (нумерация разрядов в слове - справа налево):

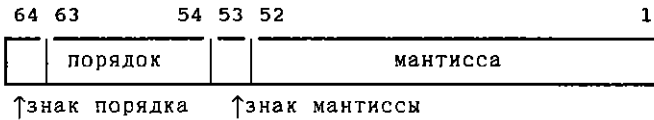
72...65	64	1
тег		информационная часть	

Информационная часть слова содержит либо одно 64-разрядное число, либо два 32-разрядных целых числа, либо 4 16-разрядных

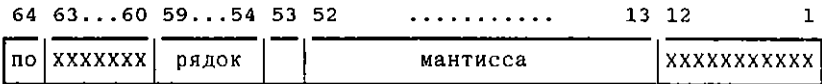
фрагмента, либо 8 байтов, либо 64 бита, либо две команды. Разрядность команды: 32 - в новом режиме, 24 - в режиме эмуляции БЭСМ-6.

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ

ПРЕДСТАВЛЕНИЕ ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ. Положительная мантисса представляется в прямом коде с "0" в разряде знака. Отрицательная мантисса представляется в дополнительном коде с "1" в разряде знака. Положительный порядок представляется в прямом коде с "1" в разряде знака. Отрицательный порядок - в дополнительном коде с "0" в разряде знака.



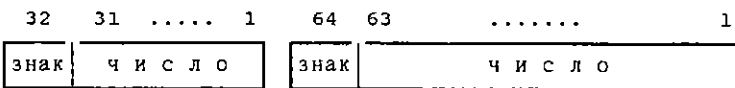
ПРЕДСТАВЛЕНИЕ ЧИСЕЛ БЭСМ-6. 48-разрядное число БЭСМ-6 получается из общего представления усечением старших 4 разрядов порядка и младших 12 разрядов мантиссы:



Такое представление дает возможность:

- а) производить арифметические операции одинаковым способом как в новом режиме, так и в режиме эмуляции БЭСМ-6;
- б) расширять диапазон порядков и увеличивать точность вычислений в режиме эмуляции.

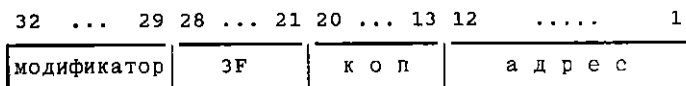
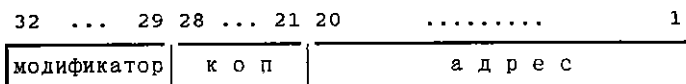
ЦЕЛЫЕ 64- И 32-РАЗРЯДНЫЕ ЧИСЛА. Для выполнения арифметики с такими числами предназначены отдельные наборы операций, работающих только в новом режиме. 32-разрядные числа размещаются по два в слове. Они могут прямо адресоваться специальными командами, исполнительные адреса в которых 21-разрядные (за счет индекс-регистров). Аналогично, в новом режиме существуют команды для работы с 16-разрядными фрагментами, байтами и битами. Разрядность исполнительного адреса у этих команд равна соответственно 22, 23 или 26.



ФОРМАТЫ КОМАНД

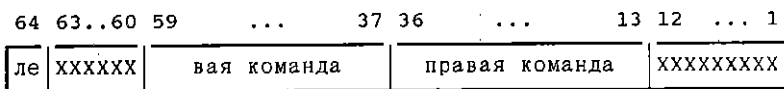
КОМАНДЫ НОВОГО РЕЖИМА. Длина каждой команды - 32 бита. В

одном слове размещается 2 команды. Формат команд имеет 2 структуры:



20-й разряд адреса в команде основного формата является знаковым. Положительный адрес представляется в прямом коде, отрицательный - в дополнительном коде. Предполагается, что программы пользователей располагаются в области положительных адресов, а программы операционной системы - в области отрицательных адресов. Разрядность всех индексных регистров (модификаторов) равна 32. При установке адреса знаковый разряд адреса размножается в 21-32 разряды индекс-регистра.

КОМАНДЫ ЭМУЛЯТОРА БЭСМ-6. Длина каждой команды - 24 бита. Структура команды - такая же, как у БЭСМ-6. Размещаются 2 команды в одном 64-разрядном слове в тех же позициях, что и числа.



Это решение продиктовано тем, что многие программы БЭСМ-6 занимаются самомодификацией. Более того, иногда запись команд производится обычной командой записи с предварительной установкой режима записи команд.

Т Е Г

Каждое машинное слово снабжено тегом, занимающим 72÷65 разряды слова. Назначение отдельных разрядов тега:

65 (1) разряд - различает команды и числа.

0 - число, 1 - команда. При попытке выборки числа в качестве команды происходит внутреннее прерывание "контроль команды". Команды, так же как и числа, могут быть использованы в качестве операндов. При записи в память 65 разряд тега устанавливается равным значению признака записи команд в регистре PPP (расширение регистра режимов), копия которого находится в теге результата (аналог признака командной свертки в БЭСМ-6). Обычно в режиме математика признак записи команд равен 0.

66 (2) разряд - различает две группы команд:

0 - команды нового режима, 1 - команды эмулятора.

Для чисел этот разряд используется аналогичным образом.

67 (3) - разрешено (0) или запрещено (1) чтение операнда.

68 (4) - разрешена (0) или запрещена (1) запись операнда.

Комбинация: "чтение запрещено", а "запись разрешена" может создаваться загрузчиком для обнаружения обращений к неинициализированным переменным.

69 (5) - разрешена (0) или запрещена (1) выборка команды (т.н. останов по номеру команды).

70 (6) - разрешена (0) или запрещена (1) передача управления на эту команду.

71 (7) - резерв.

72 (8) - признак программной (1) или аппаратной (0) интерпретации тега. При установке 72 разряда в "1" происходит внутреннее прерывание "программная интерпретация тега команды" или "программная интерпретация тега операнда".

При чтении слова из памяти все разряды тега проверяются на предмет возможной защиты и соответствия режиму работы процессора. Прерывания, возникающие при этом, могут быть блокированы соответствующими признаками из РР. При записи в память тег пишется из регистра тега результата, который может быть сформирован программно. При записи также возможно прерывание по тегу.

ПРИПИСКА И ЗАЩИТА ПАМЯТИ

При обращении в память за числами и командами 20+11 разряды исполнительного адреса слова определяют номер регистра приписки (РП): 0,1,...,1023 и при отсутствии блокировки приписки заменяются на содержимое поля номера физической страницы из этого РП. Очевидно, что часть виртуальной памяти должна использоваться для целей операционной системы. Программам пользователей эта часть памяти должна быть ограниченно доступна. Поэтому в схему приписки введена проверка виртуального номера страницы.

В режиме диспетчера разрешается обращаться за числами и командами ко всем виртуальным страницам, в режиме математика - только к страницам с положительными номерами. Проверка может быть блокирована соответствующим признаком в РР. В режиме диспетчера он установлен всегда, а перед входом в режим математика - устанавливается операционной системой.

Формат регистра приписки:

8+1 - поле номера процесса (ПНП), использующего данный регистр приписки. При обращениях в память содержимое ПНП каждый раз сравнивается с РНП - регистром номера процесса. При несовпадении возникает прерывание "чужой РП", за исключением случая, когда в ПНП - число 255. Страница с этим номером доступна всем процессам. РНП переустанавливается при смене процесса на процессоре.

9 - бит защиты страницы по записи.

"0" вызывает прерывание при записи в любое слово из этой страницы.

10 - бит защиты страницы по обращению (чтение или запись). "0" вызывает прерывание при обращении к любому слову этой страницы.

20+11 - поле номера физической страницы, заменяющее старшие разряды в исполнительном адресе.

В режиме диспетчера обращение к памяти также происходит через схему приписки и защиты.

Для поддержки алгоритма управления памятью используется схема аппаратно-микропрограммного пересчета приоритетов страниц, реализующая алгоритм расстановки приоритетов в зависимости от времени обращения к странице после запроса коррекции списков приоритетов страниц /2/. Также регистрируются факты обращения и записи в страницы памяти.

РЕГИСТРЫ

Всего в ЦП имеется 1024 32-разрядных, 16 64-разрядных и ряд специальных регистров.

32-разрядные регистры делятся на группы по 32 регистра. Для выбора группы достаточно записать в регистр номера группы (РНГ) код, определяющий группу регистров.

Каждому процессу, занимающему центральный процессор, отводится одна или несколько групп 32-разрядных регистров, часть из которых используется в качестве области упрятывания регистров общего назначения, находящихся вне группы. Для смены процессов или организации вызова экстракода в область упрятывания регистров текущего процесса достаточно скопировать содержимое регистров общего назначения, записать другой номер в РНГ и восстановить содержимое регистров общего назначения нового процесса. Эта работа выполняется специальной микропрограммой.

СИСТЕМА ПЕРЕРЫВАНИЙ И ЭКСТРАКОДОВ

Внутренние прерывания исполняются безусловно и имеют более высокий приоритет по сравнению с внешними. Внешние прерывания исполняются при отсутствии признака БВП. Всего может быть 64 различных типа внешних прерываний. Порядок отработки их определяется номером позиции в главном регистре прерываний (ГРП). При отсутствии БВП запросы на обработку прерываний от внешних устройств поступают по шине IRQ в блок реакции на прерывания в виде логической суммы запросов от всех устройств. По этому сигналу после выполнения текущей команды инициируется работа микропрограммы входа в прерывание, которая производит чтение запросов прерывания в ГРП и их гашение. При многопроцессорной системе на внешние прерывания реагирует тот процессор, который первый "откликнулся" (у которого нет БВП). Решено отказать от главного регистра маски в процессоре. Существуют только регистры масок в контроллерах и устройствах. При обработке внешних прерываний система сама определяет количество запросов от внешних устройств, анализируя состояние ГРП, и сама же сбрасывает разряды ГРП после обработки соответствующих прерываний.

Каждому процессу отводится одна или несколько групп регистров. Самый "нижний уровень" используется собственно программой, другие используются экстракодами.

Наличие у процесса запасных наборов регистров снимает весьма серьезную проблему, стоявшую перед разработчиками всех ОС БЭСМ-6 - организацию упорядочивания регистров при прерываниях и экстракодах.

СИСТЕМА КОМАНД

Арифметические, логические и команды чтения, изменяющие сумматор, изменяют признак группы (5+3 разряды регистра режимов) и признаки результата (10+7 разряды РР) в зависимости от типа операндов и результата:

а) команды чтения в сумматор 64-разрядного слова целого и с плавающей запятой, а также логические команды с 64-разрядными операндами устанавливают признаки результата следующим образом:

- N = значению 64 разряда результата;
- Z = 1, если результат равен 0;
- устанавливается логическая группа.

б) арифметические команды над целыми 64-разрядными операндами и 32-разрядными, у которых происходит размножение 32 разряда до 64-ого:

- OVR = 1 при переполнении разрядной сетки (прерывания по переполнению не происходит);
- N = значению 64 разряда результата;
- Z = 1, если результат равен 0;
- устанавливается группа умножения.

в) арифметические команды с плавающей запятой (одинарной и двойной точности):

- OVR = 1 при переполнении в разрядах порядка (прерывание по переполнению блокируется установкой БПП в РР);
- C = 1, если мантисса или порядок равны 0;
- N = значению 53-его разряда результата (знак мантиссы);
- Z = 1, если мантисса результата равна 0;
- устанавливается группа умножения.

Формирование кодов операций (КОП) происходит при выборке команды. Там же анализируется 7-й разряд короткоадресного КОП в режиме эмулятора и в зависимости от его значения дополняется адресная часть "0" или "1" в 15+13 разрядах, как это принято в БЭСМ-6.

Система команд эмулятора является подмножеством системы команд нового режима. В ЭЛБРУС-Б введено ряд новых по сравнению с БЭСМ-6 команд, имеющих коды 00H - 7FH в основном и дополнительном форматах. Новые команды МКБ-8601 имеют соответственно коды 80H - FFH в основном и дополнительном форматах. Одинаковые команды ЭЛБРУС-Б и МКБ-8601 имеют одинаковые коды.

СПИСОК КОМАНД М К Б - 8601

1. Команды пересылок

ATX запись слова в память	XTS чтение слова магазинное
STX запись слова магазинная	XTA чтение слова из памяти

2. Арифметические команды с плавающей запятой

A+X сложение с плав. запятой	A*X умножение	- " -
A-X вычитание	- " -	E+X сложение порядка с кодом
X-A обратное вычитание	- " -	E-X вычитание кода из порядка
AMX вычитание модулей	- " -	E+N сложение порядка с адресом
AVX изменение знака	- " -	E-N вычитание адр. из порядка
A/X деление	- " -	UTA выдача RMP

3. Логические команды

AAH логическое умножение	AUX разборка по маске
AEH сравнение	ACX подсчет числа единиц
ARX циклическое сложение	ANX номер старшей единицы
AOX логическое сложение	ASX сдвиг по коду
APX сборка по маске	ASN сдвиг по адресу

4. Индексные команды БЭСМ-6

ATI установка модификатора	J+M сложение модификаторов
STI ---"--- _---"---_ магазинная	UTC префиксация адресом
ITA выдача модификатора	WTC префиксация кодом
ITS ---"--- _---"---_ магазинная	VTM установка регистра адресом
MTJ пересылка модификатора	UTM сложение регистра с адресом

5. Команды передачи управления БЭСМ-6

UZA условный переход по W=0	JMEQ п/у, если модификатор = 0
UIA условный переход по W=1	JMNE п/у, если модиф. не равен 0
JMP безусловный переход	VLM конец цикла, +1 к модиф.
VJM переход с возвратом	

6. Команды с регистром режимов

XTR установка PP кодом	RTE выдача PP
NTR установка PP адресом	*80 экстракод

7. Команды с непосредственным операндом

F+U сложение с литералом	A/U деление на литерал
A-U вычитание литерала	A*U умножение на литерал
U-A обратное вычитание	ACU подсчет числа единиц
UTA выдача исполнит. адреса	ANU номер старшей единицы
AAU логическое умнож. на литерал	UTY выдача исп. адреса в RMP
AEU сравнение с литералом	UTS запись адреса магазинная
AOU логическое сложение	

8. Команды с RMP (Y)

XTY чтение из памяти в Y	YTX запись в память из Y
YAX логическое умножение на Y	AAU логич. умножение A на Y
AEX сравнение с Y	AEY сравнение A с Y
YOX логическое сложение с Y	AOY логич. сложение A с Y

9. Индексные команды

MSN сдвиг модификатора влево	A+I целое слож. модификатора
J-m вычитание модификаторов	A-I целое выч. модификатора
VTMN уст. модиф. адр. со сдв. на 1	I-A обр. цел. выч. модифик.
VTMQ уст. модиф. адр. со сдв. на 2	A/I целое деление на модиф.

VTMB уст.модиф.адр.со сдв.на 3	A*I целое умнож.на модиф.
VTMF уст.модиф.адр.со сдв.на 6	AAI лог.умножение на модиф.
UTCS префиксация адр.со сдвигом	AEI сравнение с модиф.
MPAC упаковка 4-х модиф. в A,Y	AOI лог.сложение с модиф.
MUNP распаковка 4-х модиф.из A,Y	MCJ обмен модификаторов

10. Команды передачи управления

JMGE п/у по 0 знака модиф.	JAGT п/у,если целое A > 0
JMLT п/у по 1 знака модиф.	JAGE п/у,если целое A >, = 0
JMLE п/у,если модиф.< или = 0	JALT п/у,если целое A < 0
JMGT п/у,если модиф.больше 0	JALE п/у,если целое A <, = 0
VRM конец цикла, - 1 модиф.	JAEQ п/у,если целое A = 0
UZAS п/у по 0 рг. A + чт.из маг.	JANE п/у,если целое A не = 0
UIAS п/у по 1 рг. A + чт.из маг.	JMPS переход к подпрограмме
JOVR п/у по призн.переполнения	RETS возврат из подпрограммы
JCRY п/у по призн.переноса	

11. Команды арифметики для целых 64-разрядных чисел

ASNA сдвиг арифметический	L-A обр.целое вычит. слова
A+L целое сложение слова	A/L целое деление на слово
A-L целое вычитание слова	A*L целое умножение на слово

12. Операции над полусловами

ATH запись полуслова в память	H-A обр.цел.вычит.полуслова
STH запись полуслова магазинная	A/H целое делен.на полуслово
HTA чтение полуслова из памяти	A*N целое умнож.на полуслово
HTS чтение полуслова магазинное	AAH лог.умнож.на полуслово
A+H целое сложение полуслова	AEN сравнение с полусловом
A-H целое вычитание полуслова	AON лог.слож.с полусловом

13. Операции над 16-разрядными фрагментами слова

ATQ запись 1/4 слова в память	AEQ сравнение с 1/4 слова
STQ запись 1/4 слова магазинная	AAQ лог.умнож.на 1/4 слова
QTA чтение 1/4 слова из памяти	AOQ лог.слож.с 1/4 слова
QTS чтение 1/4 слова магазинное	

14. Операции над байтами

ATB запись байта в память	AEB сравнение с байтом
STB запись байта магазинная	AAV лог.умножение байт
BTA чтение байта из памяти	AOB лог.сложение с байтом
BTS чтение байта магазинное	

15. Операции над битами

OATF запись бита в память	CLRf установка бита в 0
---------------------------	-------------------------

STF запись бита магазинная	EXTF размножение бита влево
FTA чтение бита из памяти	NTA установка бита рег.А в 1
FTS чтение бита магазинное	NTS устан.бита магазинная
AEF сравнение с битом	AEN сравнение с битом
AAF лог.умножение на бит	AAN лог.умножение на бит
AOF лог.сложение с битом	AON лог.сложение с битом
SETF установка бита в 1	

16. Арифметические команды с двойной точностью

ATD запись с двойной точностью	D-A обр.вычит.с дв.точностью
DTA чтение двойного слова	A/D деление с дв.точностью
A+D сложение с дв.точностью	A*D умнож.с дв. точностью
A-D вычитание с двойной точн.	

17. Привилегированные команды

RMOD чтение спец.регистров	SMON захват глоб.семафора
WMOD запись в спец. регистры	CMON освобод.глоб. семафора
IJP возврат из прерывания	TTX запись слова с тегом
SETR установка PP адресом	TTA чтение с тегом
CLRR гашение PP адресом	RET возврат из экстракода
SEMR семафорное чтение	HLT останов
SEMW семафорная запись	

Несмотря на то, что прошло много времени с момента разработки ЭВМ БЭСМ-6, ее основные архитектурные решения все еще современны и достаточно легко переводятся на новую технологическую базу больших интегральных схем. Малые геометрические размеры разработанной ЭВМ МКБ-8601 открывают возможности ее использования в качестве мощной ЭВМ индивидуального пользования не только как вычислителя, но и в системах управления экспериментальными физическими установками, спектральном анализе, триггерных устройствах детекторов элементарных частиц, автоматизированных рабочих местах проектировщика и т.д.

Л И Т Е Р А Т У Р А

1. Давыдов А.Л., Емелин И.А., Кадыков В.М., Ломидзе О.Н., Левчановский Ф.В., Попов М.Ю., Сапожников А.П., Сапожникова Т.Ф., Силин И.Н., Принципы организации и архитектура процессора - эмулятора МКБ-8601. - Дубна, ОИЯИ, 61-11-88-442, 1988.

2. Силин И.Н., Федюшкин Е.Д., Универсальный алгоритм разделения времени - Программирование, 1980, N=4, стр. 40-54.

Рукопись поступила в издательский отдел

24 января 1991 года.