

4. МОДУЛЬ АППАРАТНОГО УМНОЖЕНИЯ ДЛЯ МИКРО-ЭВМ

Использование микро-ЭВМ, выпускаемых отечественной промышленностью, в ряде областей науки и техники (непосредственное цифровое управление быстропротекающими процессами, обработка сигналов в реальном времени, управление научными экспериментами и т.п.) определяется их высокой производительностью при выполнении длинных арифметических операций. Так например, время выполнения подпрограммы умножения шестнадцатиразрядных целых положительных чисел в микро-ЭВМ на базе микропроцессора типа 8080 (СМ-1800, Электроника К-1-10) в среднем составляет 0,7-0,9 мс, т.е. быстроедействие на операции умножения на более 1500 оп/сек.

Для уменьшения времени выполнения длинных операций применяются специализированные модули (расширители), которые подключаются к микро-ЭВМ как дополнительные устройства памяти или дополнительные периферийные устройства, аппаратно реализуют заданную операцию или совокупность операций.

Рассмотрим пример построения такого модуля аппаратного умножения на базе БИС ММК серии К589 для универсальной микро-ЭВМ с центральным микропроцессором типа 8080.

4.1. Общая организация расширителей микро-ЭВМ

Структура типичного модуля аппаратного расширителя микро-ЭВМ приведена на рис.4.1. Модуль подключается к системной шине микро-ЭВМ /3,8/, по которой происходит обмен информацией с процессором. Как правило, с "точки зрения" центрального процессора модуль представляет собой контроллер устройства ввода-вывода, обмен данными при этом выполняется командами ввода-вывода. Иногда модуль ассоциируется с некоторой областью памяти, в этом случае обмен информацией между процессором и модулем осуществляется командами обращения к ЗУ.

Расширитель включает следующие основные узлы:

- операционное устройство, предназначено для выполнения требуемой операции или совокупности операций, в последнем случае может быть несколько операционных устройств;
- автомат управления операциями, вырабатывает последовательность управляющих сигналов, с помощью которой реализуется алгоритм выполнения требуемой операции в операционном устройстве;
- регистр состояния (статуса) модуля, хранит информацию о текущем состоянии модуля (тип выполняемой операции, если ее нес-

Состав модуля расширения

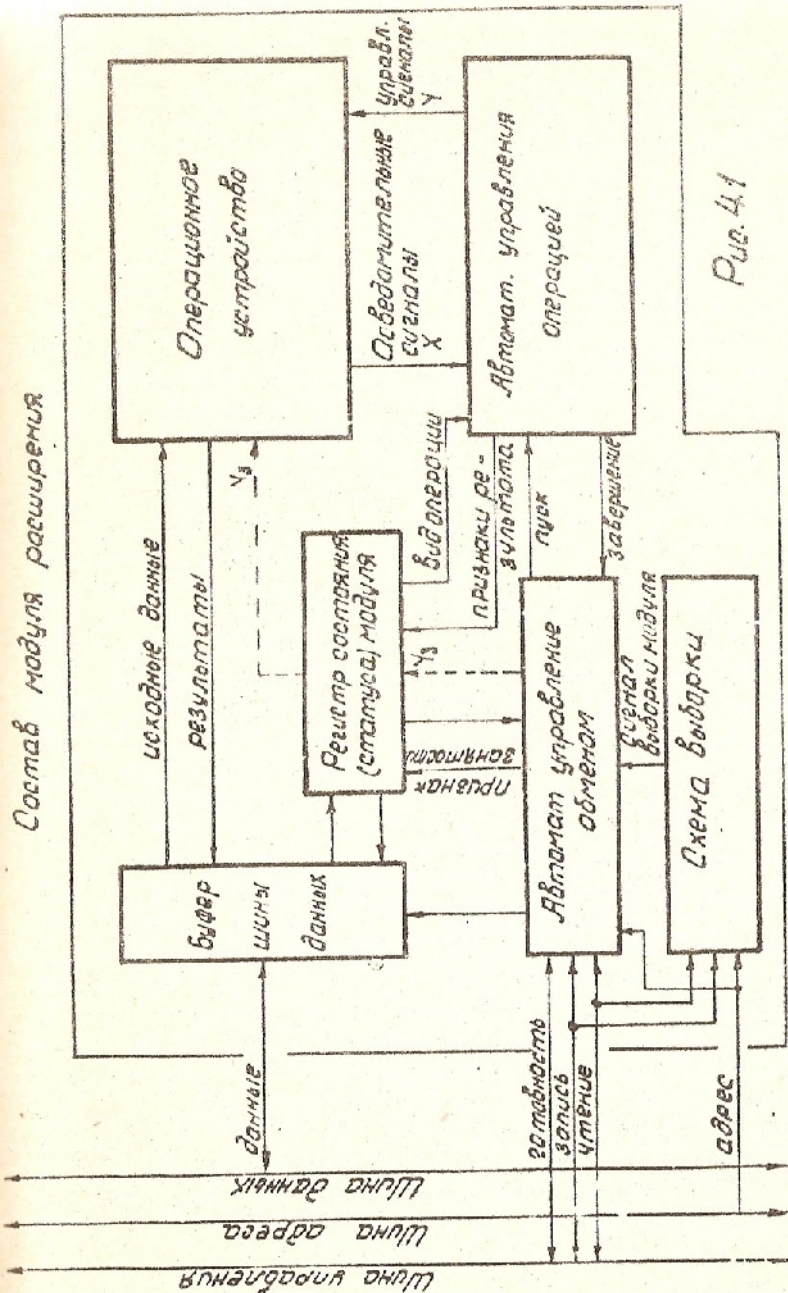


Рис. 4.1

колько, готовность модуля, признаки результата и т.п.);

- буфер шины данных, обеспечивает обмен данными между шиной данных микро-ЭВМ и элементами памяти модуля (регистрами, ЗУ);

- схема выборки, формирует сигнал выборки модуля при любом обращении к нему центральным процессором, т.е. когда на шине адреса установлен один из адресованных модулю адресов;

- автомат управления обменом, обеспечивает правильное (во времени) выполнение операций обмена между модулем и системной шиной. В некоторых случаях функции автомата управления обменом распределяются между схемой выборки и автоматом управления операцией.

Работу расширителя можно разбить на три основных этапа.

Этап 1. Прием информации. На этом этапе центральный процессор передает необходимые данные в расширитель. Как правило, сначала передается команда, которая определяет требуемую операцию и количество байт исходных данных. Команда через буфер данных принимается в регистр статус, откуда необходимые сигналы поступают в автомат управления обменом (количество байт исходных данных, куда принимать эти данные) и в автомат управления операцией (вид операции).

Затем в регистры (или ЗУ) операционного устройства побайтно принимаются исходные данные. Адресация регистров при этом может выполняться центральным процессором или, если их много (ЗУ), автоматом управления обменом.

Этап 2. Выполнение операции. Запуск автомата управления операцией осуществляется сигналом "Пуск", поступающим из автомата управления обменом. Момент генерации этого сигнала может определяться двояко. Первый способ - сигнал генерируется после приема последнего байта исходных данных. Второй способ - сигнал генерируется только после обращения центрального процессора к расширителю по заранее оговоренному адресу. В любом случае сигнал "Пуск" вызывает выполнение требуемой операции совместно операционным устройством и автоматом управления операцией. На время выполнения операции автомат управления обменом снимает сигнал готовности и (или) устанавливает в регистре статус признак занятости. При этом центральный процессор не имеет доступа к регистрам операционного устройства. Выполнение операции завершается установкой признаков результата в регистре статуса и выдачей в автомат управления обменом сигнала "Завершение". Автомат устанавливает сигнал готовности на управляющей шине и (или) снимает признак занятости.

Этап 3. Считывание результата. Обычно центральный процессор сначала считывает содержимое регистра статуса, который содержит

признаки результата, например: нормальное завершение, нулевой результат, переполнение и т.п. Затем, если необходимо, побайтно из регистров операционного устройства считывается результат выполнения операции. Адресация регистров при этом выполняется так же, как и на первом этапе.

4.2. Основные требования к проектируемому модулю

Разрабатываемый модуль предназначен для формирования 32-разрядного произведения двух целых 16-разрядных чисел при включении в состав микро-ЭВМ с универсальной микропроцессорной шиной, описанной в /3/.

Сформулируем основные технические требования, которым должен отвечать проектируемый аппаратный умножитель.

Представление информации. Сомножители задаются 16-разрядными двоичными целыми числами в дополнительном коде. Произведение должно быть сформировано в виде целого 32-разрядного двоичного числа в дополнительном коде.

Модуль должен также формировать байт состояния, в котором указываются признаки а) нулевого результата и б) короткого произведения (разрядность произведения меньше 16).

Соприжения. Модуль должен отвечать всем требованиям (алгоритмическим, электрическим и конструктивным) универсальной микропроцессорной шины /3/. Модуль должен реагировать только на команды ввода - вывода с адресами из диапазона F0 - FF. Тип модуля ведомый.

Порядок обмена. Обмен информацией между модулем и центральным процессором побайтный. Модуль должен обеспечить прием двух байт первого сомножителя, двух байт второго сомножителя и команды пуска операции. Порядок приема сомножителей произвольный. В каждом сомножителе первым должен приниматься младший байт.

Модуль должен обеспечить выдачу четырех байт результата (в произвольном порядке) и выдачу одного байта слова состояния (статуса).

Модуль не должен вызывать задержек при обмене байтами данных.

Программное обеспечение. Для центрального процессора (микропроцессор 8080) должна быть написана программа-драйвер, обеспечивающая передачу исходных данных в модуль, запуск операции умножения, и прием результата.

Принцип работы. При необходимости выполнения операции умножения центральный процессор с помощью программы-драйвера передает со-

множители в модуль. Затем в модуль выдается команда пуска операции. Сигнал готовности до время выполнения операции должен отсутствовать. После завершения операции модуль восстанавливает сигнал готовности. Далее центральный процессор читает байт статуса модуля и анализирует его, считывает байты результата.

При работе модуля один из сомножителей должен сохраняться.

Временные требования. Время выполнения собственно операции умножения не должно превосходить 200 мкс.

Элементарная база. Модуль должен быть выполнен на ТТЛ БИС серий К155, К589, К556.

Дополнительные требования. Модуль должен допускать возможность введения дополнительных операций без существенных изменений схемы.

4.3. Алгоритм умножения

Для построения модуля воспользуемся алгоритмом умножения чисел в дополнительных кодах, основанным на анализе двух разрядов множителя.

По алгоритму формирования 32-разрядного произведения требуется 16 тактов работы: $i = 0, 1, \dots, 15$, где i - номер такта (анализируемого разряда множителя). Умножение на i -й разряд производится следующим образом. Проверяется содержимое i -го и $(i-1)$ -го разрядов множителя (для $i=0$, $i-1$ разряд полагается равным нулю). Если в анализируемых разрядах содержится комбинация 10, то из старших разрядов частичной суммы вычитается множимое, если комбинация 01 - к старшим разрядам частичной суммы добавляется множимое. Для комбинаций 00 и 11 никаких операций сложения/вычитания не выполняется. Умножение на i -й разряд завершается арифметическим сдвигом (знаковый разряд распространяется) вправо частичной суммы на один разряд.

Примеры умножения 5-разрядных чисел по данному алгоритму приведены на рис. 4.2. Рассмотрим умножение чисел +10 и -5. В двоичной системе счисления в дополнительном коде множимое +10 = 01010 и множитель -5 = 11011. Десятиразрядная частичная сумма перед умножением обнуляется.

Для $i=0$, комбинация в младших разрядах множителя 10, то есть необходимо выполнять вычитание множимого из частичной суммы (сложение с 10110) и сдвиг. В результате нулевого такта получаем 1101100000. Далее $i=1$, анализируется комбинация в первом и нулевом разрядах множителя - 11. Следовательно, необходимо выполнить только сдвиг частичной суммы. Первый такт дает 1110110000. Затем анализируются второй и первый разряды множителя - 01. Выполняется операция сложения

ਮਹਾਨਾਦੀਸ਼ਵਰ
ਪਦਮਪੁਰੀ

Dono della Biblioteca
- 100385 -

[illegible]

ния с множимыми старших разрядов частичной суммы и сдвиг - 0001110000. Третий такт - анализ третьего и второго разрядов, которые содержат комбинацию 10. Снова выполняется операция вычитания множимого из старших разрядов частичной суммы и сдвиг. Получаем 111001110. И, наконец, последний (4) такт. В четвертом и третьем разрядах множителя содержится комбинация 11, выполняется только сдвиг частичной суммы - 1111001110. Таким образом за пять тактов сформировано 10-разрядное произведение в дополнительном коде (левый разряд знаковый). Переведем 1111001110 в прямой код - 0001100110 и в десятичную систему счисления - 50. Операция умножения выполнена верно. Второй пример на рис. 4.2 рекомендуется разобрать самостоятельно.

Изложенный алгоритм обладает двумя положительными качествами. Во-первых, для получения результата с правильным знаком не требуется выполнять коррекцию и, во-вторых, алгоритм допускает ускорение за счет пропуска операций сложения/вычитания при комбинациях 00 и 11.

При анализе соседних разрядов множителя можно ограничиться только младшим (нулевым) и дополнительным (минус первым) разрядами, сдвигая вправо совместно с частичной суммой и множитель, что упрощает реализацию алгоритма.

4.4. Основные этапы проектирования модуля

Для построения вычислительной части модуля будем использовать микросхему МК 1589, обладающую алгоритмической универсальностью и высоким быстродействием. За основу примем стандартную структуру процессора без конвейерного регистра (рис. 3.2). Оценки, приведенные в /4,21/, показывают, что на базе микросхем этой серии достаточно просто строятся вычислители с производительностью 2-3 миллиона микроопераций в секунду. Следовательно, при длительности операции умножения 200 нс, микропрограмма должна содержать не более 500-600 микрокоманд.

В настоящем пособии ограничимся рассмотрением вопросов структурного и логического проектирования модуля, которые решаются в три этапа.

Этап I. Выполняется предварительная разработка микропрограммы как собственно умножения, так и приема и выдачи данных. Микропрограмма составляется без учета организации безусловных переходов в БМУ, а направления условных переходов задаются mnemonic. Такая "приблизительная" микропрограмма позволяет более точно оценить вре-

ми выполнения операции и разработать подробную структуру модуля.

Этап 2. Составляется полная микропрограмма в mnemonicных обозначениях с учетом переходов в БМУ. Для этого в процессе составления микропрограммы производится распределение памяти микрокоманд.

Этап 3. Выбирается окончательно формат микрокоманды, синтезируются функциональные узлы модуля и составляется микропрограмма в двоичных адресах. Оценивается допустимая частота синхронизации схем модуля и время реализации операции умножения. Разрабатываются времязадающие и синхронизирующие схемы модуля.

4.5. Предварительная разработка микропрограммы модуля

Микропрограмму работы аппаратного умножителя удобно разбить на две части: собственно микропрограмму умножения и микропрограмму, обеспечивающую обмен данными с центральным процессором.

Составим сначала микропрограмму умножения. Как уже отмечалось, будем разрабатывать ее без учета безусловных переходов, направление условных переходов будем отмечать mnemonicески.

Начнем с распределения регистров общего назначения ЦПЭ. Так как модуль выполняет операцию

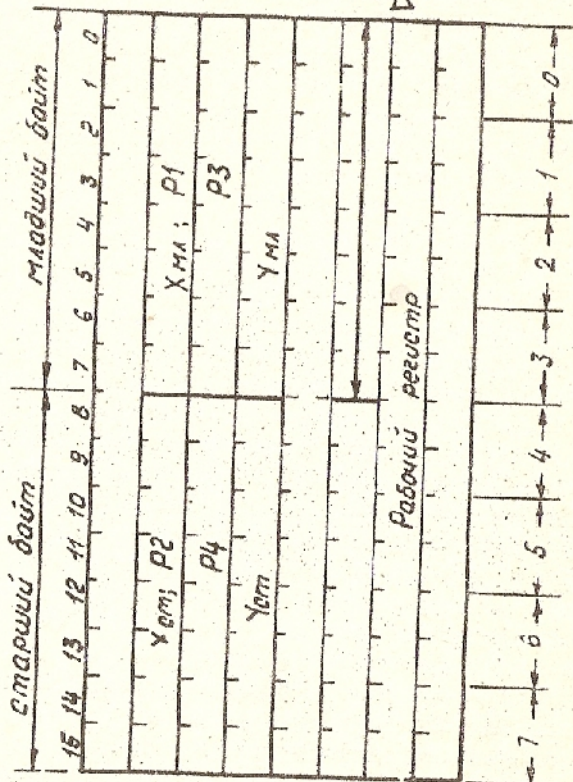
$$Z = Y \cdot X$$

над 16-разрядными числами X , Y , необходимо, во-первых, иметь восемь ЦПЭ, образующих шестнадцатиразрядный массив. Нумерация разрядов и нумерация ЦПЭ приведена на рис. 4.3. Во-вторых, для хранения X и Y необходимо выделить два регистра. Примем, что множитель X располагается в регистре $R1$, множимое Y - в регистре $R3$. Тридцатидвухразрядное произведение Z будем формировать в регистрах $R2, R4$. Обратите внимание, множитель при выполнении операции исчезает, на его место помещаются младшие шестнадцать разрядов (байты $P2, P1$) результата. В регистр $R2$ помещаются старшие шестнадцать разрядов (байты $P4, P3$) результата. При формировании частичной суммы необходимо складывать или вычитать Y . Для организации операции вычитания в регистре $R4$ будем хранить число $-Y$ (дополнительный код Y).

Так как принятый алгоритм умножения имеет циклический характер, необходимо иметь счетчик, для организации которого отведем регистр $R5$. Здесь же (в младшем байте регистра 5) будем формировать байт состояния. Положим, что нулевое состояние байта результата соответствует нормальному завершению микропрограммы, единица в младшем (нулевом) разряде байта состояния соответствует нулево-

Распределение регистров ЦПЭ

Регистр №



0 0	нормальное завершение
0 1	нулевой результат
1 0	результат меньше 215

Рис. 4.3

му произведению, аддитива в первом разряде - произведение по модулю меньше 2^{15} . Распределение регистров иллюстрируется рис.4.3.

Ориентируясь на систему микрофункций ЦПЗ (приложение, табл.П.2) составим блок-схему и предварительную микропрограмму умножения по алгоритму из п.4.3. Блок-схема приведена на рис.4.4, а микропрограмма - в табл. 4.1. Будем предполагать, что в регистре R1 записан код множителя X, а в регистре R3 - код множимого Y.

После входа в микропрограмму, в блоке 2 выполняется обнуление регистров R2, R5 ЦПЗ и флага ФЭ БМУ. Для этого (см. табл.4.1) сначала обнуляется аккумулятор А микрофункцией CLA, одновременно в БМУ с помощью микрофункции приема флага STF в триггер ФЭ записывается нулевой перенос из массива ЦПЗ. Затем микрофункциями SDR(R1), SDR(R5) содержимое аккумулятора А (накоп) переносится в R2, R5.

В блоке 3 проверяется содержимое регистра R1. Если R1 содержит код нуля ($X=0$) операция умножения не выполняется и управление передается в блок I3, где формируется соответствующий байт состояния. Для этого в ЦПЗ подается микрофункция TZR(R1), с помощью которой на выходе переноса массива ЦПЗ формируется сигнал, равный 0, если R1 равен нулю, и противном случае сигнал равен 1. Этот сигнал поступает на вход F1 БМУ, где с помощью функции перехода JFL управление передается на блок I3 (метка M13) или блок 4 (метка M4). Обратите внимание - все это выполняется одной микрокомандой!

Затем (блок 4) в регистре R4 формируется дополнительный код Y. Сначала содержимое регистра R3 переносится в аккумулятор А (микрофункция ЦПЗ IIR) и из аккумулятора - в регистр R4 (микрофункция SDR). Затем содержимое регистра R4 инвертируется (микрофункция CMR) и увеличивается на 1 (микрофункция IIR). Для формирования дополнительного кода Y в регистре R4 требуется 4 микрокоманды. В следующем блоке в регистр R5 заносится код числа -16. Для этого микрофункцией LTM код IIII IIII IIII 0000 с шины M принимается в аккумулятор А и переносится (микрофункция SDR) в регистр R5.

Блок 6 начинает цикл умножения. В нем проверяется содержимое триггера ФЭ БМУ, который соответствует дополнительному разряду множителя. Если ФЭ = 1, управление передается в блок 8, в противном случае - в блок 7. Проверка ФЭ выполняется с помощью функции перехода БМУ JZF. При этом в ЦПЗ никаких операций не выполняется. Блоки 7,8 аналогичны (они расположены в разных ветвях).

Блок-схема микропрограммы
УМНОЖЕНИЯ

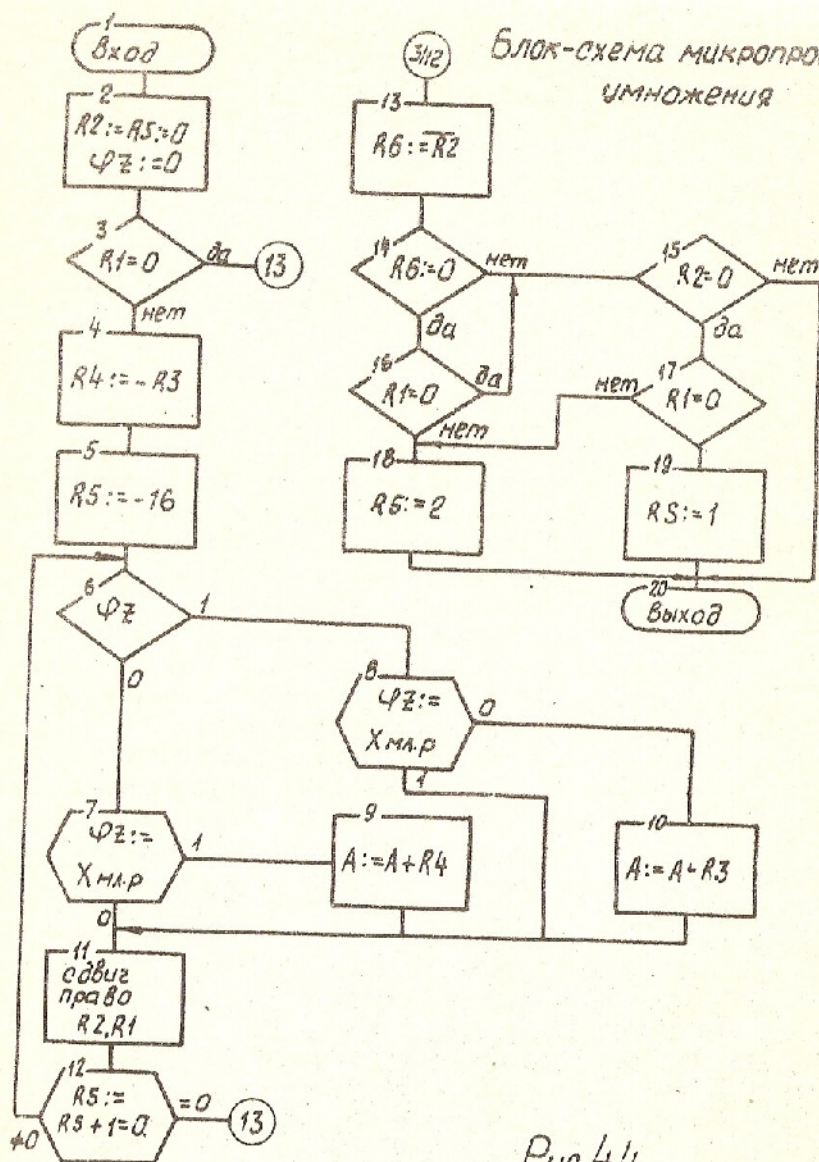


Рис. 4.4

Здесь происходит сдвиг вправо множителя X , запоминание спадающего разряда X в триггере ΦZ и анализ этого разряда. Для этого сначала содержимое регистра $R4$ переписывается в A , который сдвигается с помощью микрофункции SRA . При сдвиге спадающий разряд запоминается в триггере ΦZ (функция приема флага $EMU STZ$) и выполняется условный переход BE на блок 9 (спадающий разряд равен 1) или блок 11 (спадающий разряд равен 0). Если проверка выполняется в блоке 8, то переходы по значению разряда следующие: на блок 11, если спадающий разряд равен 1; на блок 10, если спадающий разряд равен нулю. Такая организация микропрограммы обеспечивает проверку двух соседних разрядов множителя. При комбинациях 00 и 11 управление будет передано в блок 11, при комбинации 10 - на блок 9 (сложения частичной суммы с множимым Y), при комбинации 01 - на блок 10 (сложение частичной суммы с дополнением множимого, т.е. вычитание Y).

Накопление частичных сумм (блоки 9, 10) выполняется следующим образом. Сначала содержимое регистра $R4$ (блок 9 - вычитание Y) или регистра $R3$ (блок 10 - сложение Y) записывается в аккумулятор A с помощью микрофункции ILR . Затем содержимое регистра $R2$ складывается с аккумулятором и результат записывается в регистр $R2$ (микрофункция ALR).

После блоков 9, 10 следует блок 11, в котором происходит сдвиг вправо частичной суммы и множителя X . Для этого выполняется арифметический сдвиг двух регистров $R2$, $R1$. Сначала с помощью микрофункции TZR ($R2$) и функция приема $EMU STC$ в триггере ΦC запоминается старший разряд регистра $R2$ - знак частичной суммы. Обратите внимания на код шина $K - 8000_{16}$, с помощью которого выделяется знаковый разряд. Затем содержимое регистра $R2$ переписывается в аккумулятор A и сдвигается вправо (SRA). При сдвиге в старший разряд заносится содержимое триггера ΦC (функция управления выдачей флагов $EMU FFC$), а спадающий разряд записывается снова в триггер ΦC (функция приема флагов $EMU STC$). Сдвинутый аккумулятор переписывается в регистр $R2$ ($SDR(R2)$). Затем в аккумулятор переписывается регистр $R1$ ($ILR(R1)$) и сдвигается вправо (SRA), при этом в старший разряд записывается содержимое триггера ΦC EMU , т.е. младший разряд регистра $R2$. Сдвинутое содержимое аккумулятора переписывается в регистр $R1$. Таким образом для арифметического сдвига частичной суммы и множителя X требуется шесть микрокоманд.

В блоке 12 выполняется проверка на окончание цикла умножения.

Предварительная микропрограмма умножения

Метка	Операция ЦП (F)	Управ. ЦП FO	Шина K	Управление адресом БМУ	Прием адресов	Примечание
1	2	3	4	5	6	7
M2	CLA SDR (R2) SDR (R5)	FF0 FF1 FF1	00 00 FF FF FF FF		STZ	A := 0; R2 := 0 R5 := 0 R1 = 0000 A := R3 R4 := A R4 := R4 R4 := R4 + 1 A := -16 / прием кода с шина M/ R5 := A
M3	ZER (R1)	FF0	FF FF	JFL (M15, M4)		проверка FZ
M4	ILR (R3) SDR (R4) CMR (R4) ILR (R4)	FF0 FF1 FF0 FF1	00 00 FF FF 00 00 00 00			
M5	LTM (A) SDR (R5)	FF0 FF1	FF FF FF FF			
M6	NOP	FF0	0	JZF (M7, M8)		
M7	ILR (R1) SRA	FF0 FF0	0 0	JFL (M11, M9)	STZ	A := R1 Проверка младшего разряда X, запись в FZ
M8	ILR (R4) ALR (R2)	FF0 FF0	0 F	JMO (M11)		A := R4 R2 := A + R2

1	2	3	4	5	6	7
M11	TER (R2) ILR (R2) SRA	FF0 FF0 FFC	0 0 0 0 0 0 0 0 0 0 0 0		STC STC	Занесение знака результата в FC A := R8 Арифметический! сброс и занесение младшего разряда в FC R2 := AL A := R1 Сброс с приемом из FC R1 := A
M12	ILR (R5) TER (R5)	FF1 FF0	0 0 0 0 FF F F F	JFL (M15, M6)		R5 := R5 + 1 R6 := 0 A := R1 Проверка младшего разряда X, запись в FZ
M8	ILR (R1) SRA	FF0 FF0	0 0 0 0 0 0 0 0	JFL (M10, M11)	STC	A := R3 R2 := A - R2
M10	ILR (R3) ALR (R2)	FF0 FF0	0 0 0 0 FF F F F	JMP (M11)		A := R2 R5 := A R6 := R6
M13	ILR (R2) SDR (R6) THR (R6)	FF0 FF1 FF0	0 0 0 0 FF F F F 0 0 0 0			R6 := 0 (R2 = F F F F) R2 := 0 R1 := 0 R1 := 0 R5 := R5 + 1 (R5 := 2) R5 := R5 + 1 (R5 := 1)
M14	TER (R5)	FF0	FF F F F	JFL (M15, M15)		
M15	TER (R2)	FF0	FF F F F	JFL (M11, M10)		
M17	TER (R1)	FF0	FF F F F	JFL (M15, M18)		
M18	ILR (R5)	FF1	0 0 0 0	JFL (M19, M16)		
M19	ILR (R6) ILR (R5)	FF1 FF1	0 0 0 0 0 0 0 0	JMP (M20) JMP (M20)		
M20	NOP	FF0	0 0 0 0	JMP S1		Выход

Сначала содержимое регистра RS увеличивается на 1 (микрофункция ILR(RS)), затем с помощью микрофункции TZR проверяется содержимое регистра RS на ноль. Если содержимое отлично от нуля, то на выходе переноса массива ЦПЗ формируется единичный сигнал и функция перехода БМУ ZFL передаст управление на блок 8 (метка M6). Если в RS содержится ноль, на выходе переноса будет нулевой сигнал и функция перехода БМУ ZFL передаст управление на блок 13.

Оставшаяся часть микропрограммы умножения предназначена для формирования байта состояния. В блоке 13 в регистре RG формируется ливерсия старших разрядов результата. Если это дополнение равно 0, то в старших разрядах результата содержится все 1, т.е. результат отрицательный и меньше, чем 2^{15} . Построение блока 13 аналогично блоку 4, за исключением микрокоманды увеличения на 1. Проверки (блоки 14, 15, 16, 17) организованы с помощью микрофункций TZR. Занесение в младший разряд байта состояния (RS) единицы производится микрофункцией ILR(RS) (блок 19). Занесение единицы в первый разряд байта состояния производится двумя последовательными микрофункциями ILR(RS), ILR(RS) (блок 18). В блоке 20 организуется выход из микропрограммы умножения.

Оценим ориентировочно длительность выполнения разработанной микропрограммы. Основной цикл (блоки 6, 7, 9, 11, 12) выполняется 16 раз и содержит 14 микрокоманд. Блоки 2, 3, 4, 5 включают 10 микрокоманд. Столько содержит блоки 13, 14, 16, 15, 16, 18, 20. Таким образом, для умножения требуется выполнить $16 \times 14 + 20 = 224$ микрокоманд. При производительности 2 миллиона микроопераций в секунду (тактовая частота 2 МГц) для этого потребуется 122 микросекунды, что отвечает поставленным временным требованиям.

Перейдем к рассмотрению организации обмена данными между модулем аппаратного умножения и центральным процессором микро-ЭВМ.

Обращение к модулю со стороны центрального процессора может производиться двумя командами. Командой ввода IM, при которой вырабатывается сигнал чтения (TORC). Модуль, при появлении команды ввода, должен передать в процессор один из четырех байтов результата или байт состояния. Присвоим следующие адреса:

P0 - младший байт шестнадцати младших разрядов результата P1 (расположен в R1);

P1 - старший байт шестнадцати младших разрядов результата P2 (расположен в R1);

P2 - младший байт шестнадцати старших разрядов результата P3 (расположен в R2);

R3 - старший байт шестнадцати старших разрядов
результата R4 (расположен в R2);

R4 - байт остаточного (расположен в R5).

Вторая команда обращения к модулю - команда вывода OUT, при которой вырабатывается сигнал записи (IOWC). По этой команде модуль должен принять один из четырех информационных байтов или приказ пуска операции умножения. Присвоим следующие адреса:

R0 - младший байт множителя X (принимается в R1);

R1 - старший байт множителя X (принимается в R1);

R2 - младший байт множителя Y (принимается в R3);

R3 - старший байт множителя Y (принимается в R3);

R4 - пуск операции умножения.

Примем следующий общий принцип организации обмена. Модуль аппаратного умножения всегда находится в состоянии ожидания обращения к нему, анализируя адреса на шине адреса. При появлении одного из адресов (R0 - R4) и сигнала IOWC или IORS происходит выполнение соответствующей операции. При получении команды пуска операции модуль снимает сигнал готовности и не реагирует на все обращения до завершения умножения. Блок-схема микропрограммы обмена приведена на рис.4.5, а предварительный вариант микропрограммы - в табл.4.2.

Напомним, что в БМУ имеется четыре входа K7 - K4, состояние которых можно анализировать функцией перехода JPK. При обращении к модулю, будем формировать на входах K7 - K4 комбинацию сигналов, несущую информацию о адресе обращения и состоянии линий IOWC, IORS. Микрофункция перехода JPK передает управление на ту ветвь микропрограммы, которая соответствует адресу и сигналам чтения или записи.

Выдача информации (блоки B12, B13, B14, B15, B17) производится аналогично, одной микрокомандой ILR, переписывающей содержимое соответствующего регистра в аккумулятор, откуда код поступает на линии данных ЦПЗ через выходной буфер данных. Обратите внимание, что для чтения младших байтов R2, R1 и старших байтов R4, R3 результата используются по одной микрокоманде: ILR(R1), ILR(R2), соответственно. Выбор младшего или старшего байта осуществляется управлением выходными буферами данных младших четырех или старших четырех ЦПЗ.

Запись X и Y в регистры (соответственно блоки B8, B9, и B10, B11) выполняется одинаково. Прием происходит следующим образом

Блок-схема микропрограммы обмена

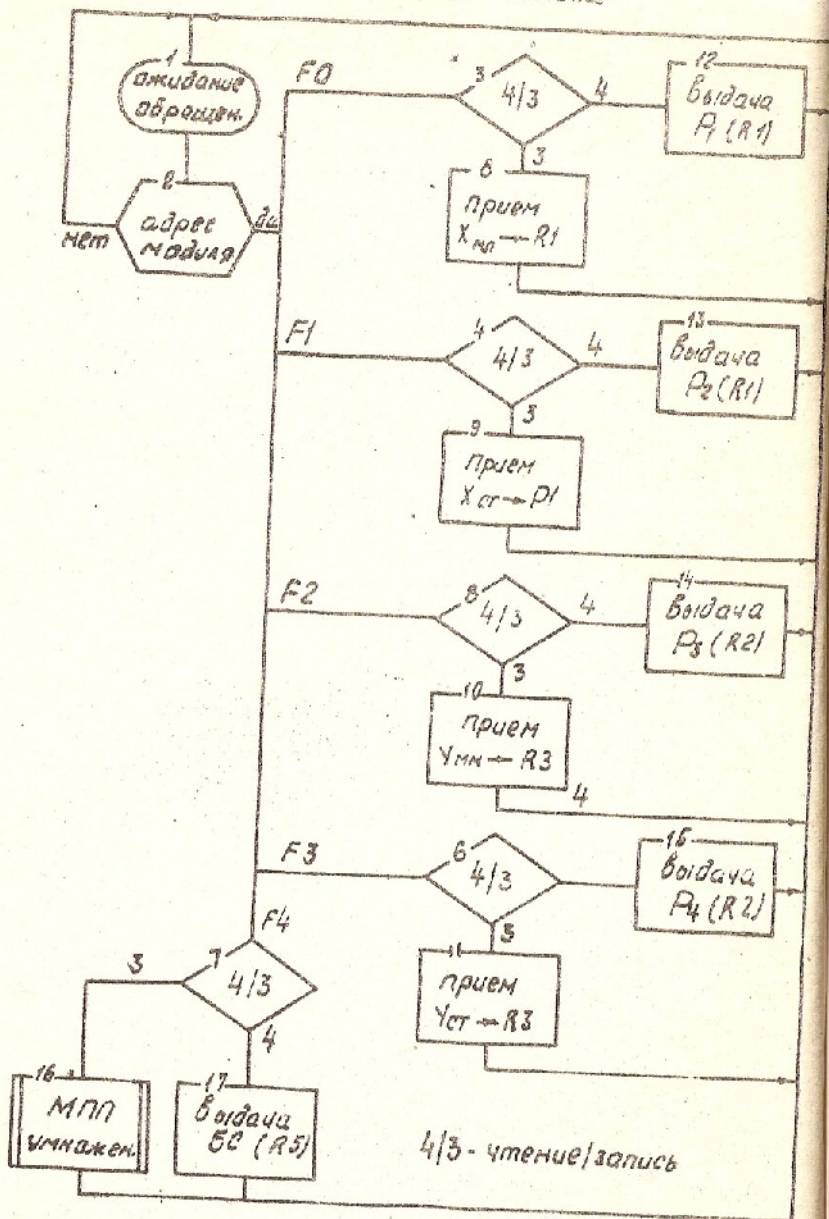


Рис. 4.5

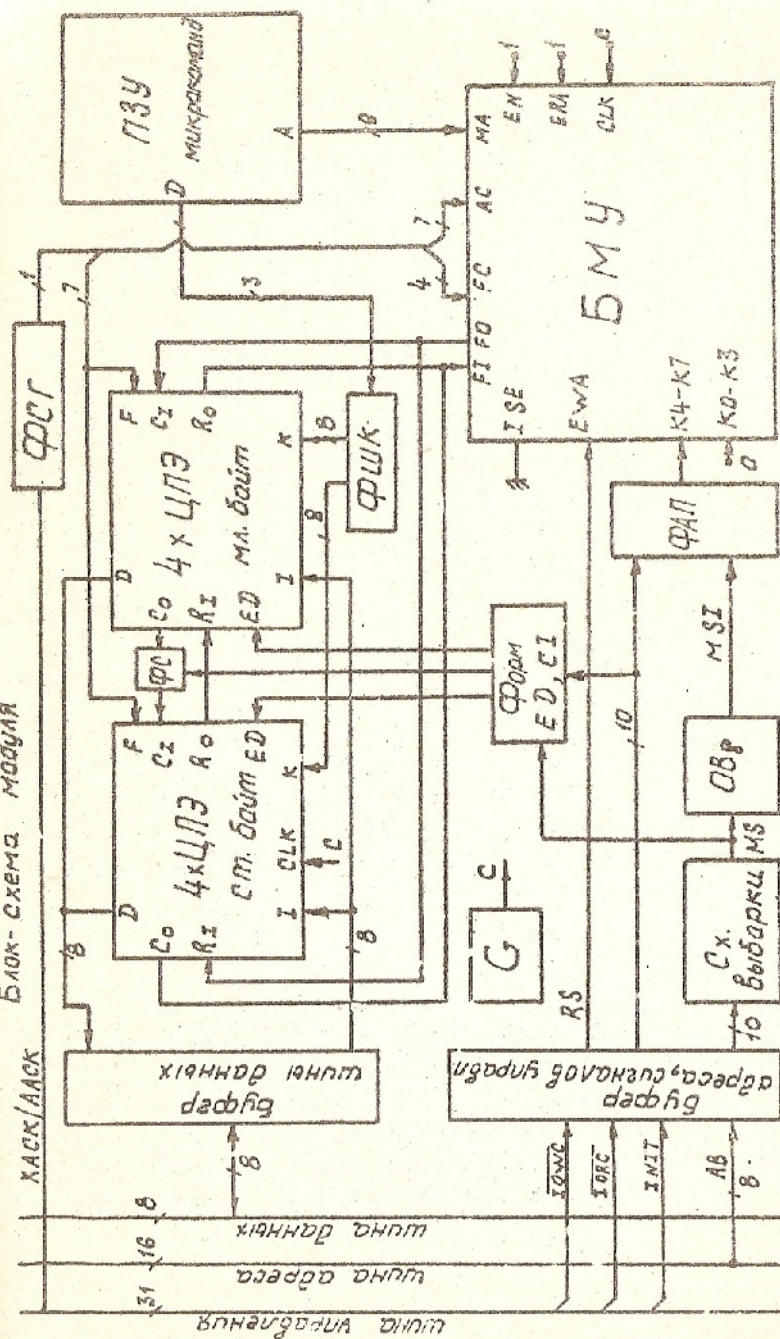
Метка	Операция (F)	Управляющий F0	Шина К	Управление адресом БУ	Принем флага	Примечание
1		3	4	5	6	7
B1	NOP	FF0	0000	JPX	HCE	
B12B13	ILR (R1)	FF0	0000	KB1		Выдача R1 BA (R1, R2)
B14B15	ILR (R2)	FF0	0000	KB1		Выдача R2 BA (R3, R4)
B17	ILR (R5)	FF0	0000	KB1		Выдача R5 BA (байт состояния)
B8	LDI	FF1	00FF	KB1		Приним мл. байта X; CI для старшего байта := 0
B9	ORI	FF0	FF00			Приним ст. байта X;
B9+1	SDR (R1)	FF1	FFFF	KB1		Запись X в R1
B10	LDI	FF1	00FF	KB1		Приним мл. байта Y; CI для старшего байта := 0
B11	ORI	FF0	FF00			Приним старшего байта Y
B11+1	SDR (R2)	FF1	FFFF	KB1		Запись Y в R2
B15	NOP	FF0	0000	JMP (умножен)		Переход к микропрограмме умножения

(рассмотрим на примере X). Сначала центральный процессор микро-ЭВМ выполняет команду $OUT F0$ - запись младшего байта X . Микрофункция перехода JRX воспринимает соответствующие сигналы на входах $K7 - K4$ БМУ и передает управление в блок В8. Здесь в ЦПЭ подается микрофункция LDT , по которой информация с входных линий I принимается в аккумулятор. Обратите внимание на маску шины K ЦПЭ - $00FF$, которая выделяет только младший байт. Таким образом, младший байт X будет принят в младшие 8 разрядов аккумулятора, а старшие 8 разрядов аккумулятора обнулятся. Управление в микропрограмме снова передается в JRX . Затем центральный процессор микро-ЭВМ выполнит команду $OUT F1$ - запись старшего байта X . Микрофункция перехода JRX передаст управление в блок В9. Старший байт принимается здесь в старшие восемь разрядов аккумулятора (микрофункция $OK1$, шина $K - FF00$). После этого содержимое аккумулятора (все 16 разрядов X) переписывается в регистр $R1$ (микрофункция $STR(R1)$). Управление снова возвращается в В1. Если процессор выполняет команду $OUT F4$, функция перехода передает управление в блок В16, где выполняется команда безусловной передачи управления в микропрограмму умножения.

4.6. Блок-схема модуля

Модуль аппаратного умножения с учетом проработки микропрограммы и требований к сопряжению с шиной микро-ЭВМ можно построить из следующих блоков и элементов (рис.4.6):

1. Блок микропрограммного управления (БМУ) - формирует адрес следующей микрокоманды;
2. ПЗУ микрокоманд - хранит микропрограмму работы модуля;
3. Массив ЦПЭ - состоит из 8 элементов (ЦПЭ) и реализует микрооперации над шестнадцатиразрядными данными в соответствии с разработанной микропрограммой. Массив ЦПЭ разделен на две части: четыре ЦПЭ младшего байта ($M0, 1, 2, 3$) и четыре ЦПЭ старшего байта ($M4, 5, 6, 7$). Нумерация ЦПЭ справа-налево;
4. Схема формирования переноса (SC) - предназначена для формирования нулевого сигнала переноса при приеме младшего байта X или Y в регистры ЦПЭ;
5. Формирователь шины K (ФМК) - формирует сигналы на входных линиях K массива ЦПЭ в соответствии с кодом в поле K текущей микрокоманды;
6. Формирователь сигналов ED, CI - предназначен для управления выходными буферами данных младшего и старшего байтов массива



Dec. 4. 6

ЦПЗ при выполнении операций вывода информации из модуля, формирует также сигнал $G1$ управления ФС;

7. Буфер шины данных - обеспечивает сопряжение двунаправленных линий данных шины микро-ЭВМ с входными линиями I и выходными линиями O массива ЦПЗ;

8. Буфер адреса и сигналов управления - обеспечивает развязку и согласование линий адреса и управления шины микро-ЭВМ и внутренних схем модуля;

9. Схема выборки - формирует сигнал выборки модуля MS , когда центральный процессор микро-ЭВМ обращается к модулю (на шине адреса комбинации $P0 - P4$);

10. Формирователь адреса перехода (ФАП) - формирует код на входе команды БУ ($K7 - K4$) в соответствии с адресом на шине микро-ЭВМ;

11. Одновибратор - вырабатывает импульсный сигнал выборки модуля MSI определенной длительности для отсраивания выдачи кода на выходе ФАП;

12. Формирователь сигнала готовности (ФСГ) - генерирует сигнал готовности модуля при выполнении операции умножения и выдает его на линии ХАСК, шины микро-ЭВМ.

Рассмотрим основные элементы работы модуля.

Умножитель приводится в исходное положение высоким уровнем на линии $INIT$ шины микро-ЭВМ. По этому сигналу в буфере адреса формируется сигнал сброса RS , который подается на вход загрузки команды БУ EWA . При этом на входах команды БУ ($K7 - K0$) должен стоять адрес первой команды микропрограммы $B1$ (см. табл. 4.2).

После снятия сигнала $INIT$ начинается выполнение этой микрокоманды. В поле функции перехода записан код IPX , таким образом она будет циклически выполняться до тех пор, пока на входах $K7 - K0$ сохраняется ее адрес. Отметим, что эта же микрокоманда должна формировать истинный сигнал готовности - модуль готов к обмену с процессором и выполнению операции умножения.

Прием информации рассмотрим на примере записи в модуль множимого Y . Сначала на линиях адреса шины микро-ЭВМ устанавливается адрес $F2$, затем на линиях данных появляется младший байт Y , сопровождаемый низким уровнем сигнала $TOWC$ (процессор выполняет команду OUT - см. /3, 19/). При совпадении адреса $F2$ и сигнала $TOWC$ срабатывает схема выборки и формирует сигнал выборки модуля MS . Одновременно буфер шины данных настраивается на прием информации и пропускает младший байт Y на входы I массива ЦПЗ.

(обратите внимание — одновременно на младший и старший байт массива ЦПЭ). Формирователь ЕД, СХ вырабатывает сигнал СТ, который запрещает передачу переноса из младшего байта массива ЦПЭ в старший. Адрес P2 подается также на входы схемы ФАП, которая формирует код перехода к микрокоманде B40 (табл.4.2). По переднему фронту сигнала MS одновибратор формирует импульсный сигнал MSI, который стробирует выдачу кода с ФАП на входы команды БМУ. Функция перехода ВРХ анализирует этот код и формирует адрес микрокоманды ВЮ. По этой микрокоманде младший байт Y принимается в аккумулятор ЦПЭ. Причем прием происходит только в младшей половине массива ЦПЭ, так как на входах К ЦПЭ согласно микрокоманде устанавливается код 00FF. В старший байт аккумулятора записывается код нуля, потому что перенос из младшего байта запрещен. После микрокоманды ВЮ управление снова возвращается к В1 — модуль ожидает обращение. Подобно выполняется прием старшего байта Y. Основное отличие заключается в том, что выбирается микрокоманда ВП, по которой старший байт Y принимается в старший байт аккумулятора массива ЦПЭ (шина К — FF001). После микрокоманды ВП следует микрокоманда ВП + 1. Здесь содержимое аккумулятора (шестнадцатый разряд двоичного Y) переписывается в регистр R3, а управление передается к микрокоманде В1 — ожидание обращения к модулю. Аналогично выполняется прием множителя X и команды нулевого оператора.

Передача данных из модуля в процессор микро-ЭВМ происходит в следующей последовательности (на примере чтения байта состояния). Процессор микро-ЭВМ устанавливает на линиях адреса шину код P4 и сигнал на линии IORC переходит в нижний уровень (команда IW MP 8080). По комбинации P4 на линиях адреса и сигналу IORC схема сборки формирует сигнал MS. Буфер шины данных настраивается на выдачу информации, подключая выходы D старшего и младшего байтов массива ЦПЭ к линиям данных шин микро-ЭВМ. При этом выходы D переходят из третьего состояния только в одной из половин массива ЦПЭ, что обеспечивается формированием соответствующего сигнала ЕД управления входным буфером данных ЦПЭ. Для адреса P4, схема формирования ЕД вырабатывает сигнал разрешения для младшего байта массива ЦПЭ, таким образом на линии данных микро-ЭВМ подается содержимое младшего байта аккумулятора ЦПЭ. По сигналу MS срабатывает одновибратор и схема ФАП, формируя на входах команды БМУ код перехода к микрокоманде ВП7. Микрофункция TLR(25) этой микрокоманды заносит содержимое регистра R3 в аккумулятор.

следовательно младший байт 85 (байт состояния) будет выдан на линии данных шины микро-ЭВМ и принят в микропроцессор. После микрокоманды В17 управление в микропрограмме снова передается к В1 - микрокоманде ожидания обращения к модулю.

Работа модуля при выполнении собственно микропрограммы умножения аналогична описанной в п.3. Важно только помнить, что в процессе выполнения собственно операции сигнал готовности отсутствует, сообщая тем самым центральному процессору, что модуль занят.

4.7. Полная микропрограмма работы модуля

Перейдем к разработке полной микропрограммы умножителя в мнемонических обозначениях. Ориентируясь на выбранную структуру модуля и предложенную организацию его работы, выберем формат микрокоманд.

Микрокоманда содержит следующие поля (рис.4.7)

1. Семirazрядное поле управления операцией (микрофункцией) ЦПЭ F_6-F_0 - задает микрофункцию, которая должна выполняться массивом ЦПЭ в данном такте.
 2. Трехразрядное поле управления шиной К ЦПЭ - задает 5 различных комбинаций из входов К массива ЦПЭ: 0000_{16} , $F7FF_{16}$, $00FF_{16}$, $FF00_{16}$, 8000_{16} .
 3. Семirazрядное поле управления адресом БМУ AC_6-AC_0 - определяет функцию перехода БМУ, с помощью которой формируется адрес следующей микрокоманды.
 4. Двухразрядное поле управления приемом флагов FC_3, FC_2 - определяет состояние флагов FC, FD БМУ.
 5. Двухразрядное поле управления выключением флагов FC_1, FC_0 - определяет состояние выхода флага FO БМУ (состояние входа SX массива ЦПЭ).
 6. Одноразрядное поле управления сигналом готовности модуля.
- Таким образом микрокоманда модуля должна содержать 22 разряда. Как указывалось ранее (п.3), при составлении микропрограмм удобно пользоваться таблицей распределения ячеек ПЗУ микрокоманд и таблицей переходов между микрокомандами (для разрабатываемой микропрограммы эти таблицы приведены на рис. 4.8, 4.9 соответственно). Собственно микрокоманды последовательно записываются в таблице, формат которой расширен по сравнению с табл.4.1 за счет колонок "Адрес микрокоманды" и "Адрес следующей микрокоманды". В этих колонках записывается адрес ячейки ПЗУ, в которой хранится данная микрокоманда, и адрес ячейки ПЗУ, из которой будет извлекаться следующая микрокоманда.

Формат микрокоманды модуля умножения

Рядов	Обознач.	Куда подключен
0	МК ₀	F ₆
1	МК ₁	F ₅
2	МК ₂	F ₄
3	МК ₃	F ₃
4	МК ₄	F ₂
5	МК ₅	F ₁
6	МК ₆	F ₀
7	МК ₇	
8	МК ₈	
9	МК ₉	
10	МК ₁₀	УА ₆
11	МК ₁₁	УА ₅
12	МК ₁₂	УА ₄
13	МК ₁₃	УА ₃
14	МК ₁₄	УА ₂
15	МК ₁₅	УА ₁
16	МК ₁₆	УА ₀
17	МК ₁₇	УФ ₃
18	МК ₁₈	УФ ₂
19	МК ₁₉	УФ ₁
20	МК ₂₀	УФ ₀
21	МК ₂₁	
22	МК ₂₂	
23	МК ₂₃	

Входы
управления
функций
ЦПЭ

Входы
формирования
шины К

Входы
управления
регистром
адреса
БМУ

Входы управ-
ления приг-
ном слагаеб

Входы управ-
ления выде-
чей слагаеб

сигнал готовности

Резерв

Резерв

Рис. 4.7

манда. Адреса фиксируются в виде двух шестнадцатеричных цифр, например, 7 - 14 означает ячейку памяти с адресом 00111110 (седьмая строка, четырнадцатая колонка).

Рассмотрим размещение микрокоманд для микропрограммы ожидания обращения и обмена. Микрокоманду ожидания обращения разместим в ячейке 0 - 15 (нулевая строка, пятнадцатая колонка). По этой микрокоманде никаких операций в ЦПЭ не выполняется, поэтому в поле операции ЦПЭ записываем микрофункцию NOP. На шине K должна быть комбинация 0000, на входе СТ массива ЦПЭ должна быть 0, поэтому в поле управления выходом флага записываем лемоническое обозначение FFO. Триггера ФС, ФЭ БМУ должны сохранять свое состояние - в поле управления приемом записывается HCF. В поле управления адресом БМУ записываем JRX - условный переход по входам K₇-K₄ БМУ. Эта функция определяет переход на одну из ячеек нулевой строки, номер которой задается двоичным кодом на входах K₇-K₄. В поле готовности проставляем символ R - модуль готов и работе. Так формируется первая микрокоманда микропрограммы, см. табл. 4.3.

Расположим в первых трех ячейках нулевой строки микрокоманды выдачи информации: в ячейке 0-0 микрокоманду BI2, в ячейке 0-1 микрокоманду BI4, в ячейке 0-2 микрокоманду BI7. Все эти микрокоманды одинаковы и отличаются только номером регистра, который переписывается в аккумулятор ЦПЭ. Например, для выдачи младших шестнадцати разрядов результата (байты P2, P1) необходимо в аккумулятор переписать содержимое регистра R1, т.е. в поле операции ЦПЭ микрокоманды BI2 записываем ILR(R1), причем на шине K ЦПЭ должна быть комбинация 000016. Вход СТ массива ЦПЭ (выход FO БМУ) должен быть в 0, поэтому в поле управления выходом флага БМУ записываем FFO. Состояние триггеров ФС, ФЭ БМУ в рассматриваемой микрокоманде безразлично, поэтому в поле управления приемом флагов ставим HCF. После формируемой микрокоманды должна выполняться микрокоманда BI1, т.е. необходимо передать управление в ячейку 0-15. Таким образом, следующий адрес находится в текущей (нулевой) строке и можно воспользоваться функцией перехода JCR, которая и записывается в поле управления адресом БМУ. Так составляется микрокоманда BI2, аналогично формируются микрокоманды BI4, BI7.

Рассмотрим микрокоманды приема информации (на примере приема множителя X). Ранее мы определяли, что прием младшего байта и прием старшего байта выполняется отдельными микрокомандами. Младший байт принимается в аккумулятор микрокомандой BI9. Расположим ее в ячейке 0-6. Для приема информации с входной шины I в акку-

мулятор используем микрофункцию $LDI(AC)$, причем на вход CI младшего байта ЦПЭ должна подаваться 1. Следовательно, в поле управления выходом флага FO БМУ записываем $FF4$. Так как прием происходит только в младший байт на входах K ЦПЭ должна быть комбинация $COFF_{16}$. После этой микрокоманды необходимо перейти к микрокоманде BI - в поле управления адресом БМУ записываем функцию перехода JCR в адресу $0-15$.

Прием старшего байта (BII) в старшие разряды аккумулятора ЦПЭ происходит по микрооперации $ORI(AC)$ при комбинации на входах K ЦПЭ $FF0_{16}$. На вход CI ЦПЭ необходимо подать 0, поэтому в поле управления выходом флага БМУ записываем $FF0$. После приема Y в аккумулятор, его необходимо переписать в регистр $R3$. Это выполняется микрокомандой $BII + I$, которую расположим в ячейке $I-7$. Тогда для перехода из $0-7$ в $I-7$ поле управления адресом БМУ должно содержать функцию JCC - переход в текущей колонке.

Перезапись содержимого аккумулятора (код Y) в регистр $R3$ осуществляется микрофункцией $SDR(R3)$, комбинация на шине K $FFFA$. В поле управления FO записывается операция FPI - на вход CI ЦПЭ необходимо подать 1. После микрокоманды $BII+I$ (ячейки $I-7$) управление снова должно быть передано в микрокоманде BI , т.е. в ячейку $0-15$. Такой переход (изменяется и номер строки, и номер колонки) можно осуществить только с помощью функции JZR - переход к нулевой строке.

Аналогично формируются микрокоманды приема X ($B8, B9, B9+I$). Прием команды начать операцию умножения происходит с помощью передачи управления в ячейку $0-8$. Здесь никаких операций в ЦПЭ не выполняется, а следующая микрокоманда с помощью функции перехода выбирается из ячейки $I-8$, где собственно и начинается микропрограмма умножения.

При записи микропрограммы умножения будем группировать команды в $8 - 15$ колонках. Для организации условных переходов по флагу F и триггеру QZ (см. предварительную микропрограмму, табл. 4.1) используем функции перехода JFL и JZF . Если микрокоманды переходят располагаются в колонках $8 - 15$ (строки $0 - 16$), то микрокоманды - "приемники" управления должны всегда располагаться в колонках $10, 11$. Причем, переход в колонку происходит при нулевом значении опрашиваемого триггера, а в колонку 11 - при единичном.

Рассмотрим распределение памяти для начального участка микропрограммы умножения. Микрокоманду $M2$ разместим в ячейке $I-3$, микрокоманду $M2 + I$ в ячейке $2-I$ (переход в колонку - JCC). Микроко-

манду M2+2 расположим в ячейке 2-9 (переход в текущей строке - JCR). Далее следует микрокоманда M3 - условный переход по состоянию триггера F БУ. Эту команду запишем в ячейку 1-9. Микрокоманды - "приемники" условного перехода удобно записать в ячейки 1-10 (микрокоманда M13 - переход по нулю) и 1-11 (микрокоманда M4 - переход по единице). Затем распределяется память ЦЗУ для блока M4 и так далее. Выбранные адреса микрокоманд фиксируются в колонках "Адрес МК" и "Следующий адрес МК" табл.4.3. Одновременно указывается и функция управления адресом БУ: (при переходе в колонке JCC , при переходе в строке JCR) и заполняется таблица переходов между микрокомандами (рис.4.9). При дальнейшем распределении памяти необходимо принимать во внимание два следующих факта.

Во-первых. Так как колонки 10,11 используются при организации главных переходов и занимать их для записи последовательно выполняемых микрокоманд нежелательно, то для микрокоманд - "приемников" условных переходов необходимо подыскивать строки со свободными ячейками. Например, микрокоманда M6 (размещена во второй строке, колонки двенадцать) должна проверять состояние флага Ф2 и передавать управление к M7 (Ф2=0) или к M8 (Ф2=1). Микрокоманды M7, M8 предпочтительно расположить в строке 3, колонки 10,11 так как вторая строка уже вся занята микрокомандами второго и пятого блоков.

Во-вторых. Управление к некоторым микрокомандам передается из различных источников, поэтому часто требуется вводить промежуточные микрокоманды, задающие правильные переходы по колонкам в строкам. Например, микрокоманда M11 (блок II - одновременный сдвиг регистров R1, R2 вправо) принимает управление из четырех источников: из блоков M7, M8 в результате условных переходов и из блоков M9, M10 безусловно. Начальную микрокоманду II блока разместим в ячейке 4-8. Для перехода к этой микрокоманде используются промежуточные микрокоманды M11^в в ячейках 4-10, 5-8, 6-11. Переход из блока M7 происходит через промежуточную микрокоманду 4-10. Переход из блока M8 происходит через микрокоманды в ячейках 5-11, 5-8. Переход из блока M9 промежуточных микрокоманд не использует. Переход из блока M10 осуществляется через промежуточную микрокоманду в ячейке 5-8. Никаких действий в ЦПЗ во время выполнения промежуточных микрокоманд не производится.

Выход из микропрограммы умножения возможен в блоках M20, M19, M18. В поле управления переходом этих микрокоманд записываем функцию перехода к нулевой строке JZR с адресом C-15, т.е. переход к микрокоманде ожидания обращения к модулю. Сигнал готовности во

Распределение ячеек ПЗУ микрокоманд

столбец №

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

812138415817				88	89	810	811	816							812
					89+1		811+1	816		M13	M14	M4+1	M4+2	M4+3	M13+1
								M2+1	M2+2	M13*	M6*	M6	M6+1	M5	M13+2
								M11+1	M7+1	M7	M8	M8+1	M11+2	M11+3	M14
								M11	M9+1	M11*	M9	M11+5	M11+5	M11+4	
								M11*	M10+1	M10	M11*	M12	M12+1		
									M16	M16*	M15*	M15			
									M18+1	M15*	M18	M16*			
										M17	M20				
										M19	M18*				

4
строка

6

8

9

Рис. 4.8

Пареходы между микроконтантами

Вход по сдвигу
INIT

0-0, 1, 2, 4, 5, 6, 7, 8

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1-3
0-15	0-15	0-15	0-15	0-15	0-15	0-15	0-15	0-15	0-15	1-9	1-9	1-11	1-12	1-13	1-10
0-15	0-15	0-15	0-15	0-15	1-5	0-15	0-7	0-8	2-9	1-15	1-12	1-13	1-14	2-14	2-15
					0-5	0-15		1-8	2-8	3-13	5-13	2-13	2-14	1-15	3-15
										1-10	2-12	3-11	2-12	3-13	2-15
								2-9	1-9	2-12	3-12	4-13	3-14	4-14	5-15
								4-8	7-10	2-12	3-12	4-13	4-14	3-14	5-15
								5-13	4-11	3-9	3-1	4-13	4-14	3-14	5-15
								4-10	4-11	3-9	4-15	5-12	4-12	4-13	
								5-8	4-8	4-15	4-9	5-12	4-12	4-13	
								5-11	5-10	3-12	5-12	4-12	5-12		
								5-8	5-8	5-9	5-8	5-13	5-11		
								6-10	6-10	4-15	4-15	5-11	5-12		
								7-11	7-11	6-9	6-12	6-12	6-11		
								8-11	8-11	7-12	7-9	7-10	8-10		
								9-12	9-12	8-12	8-12	8-12	8-10		
								10-13	10-13	11-13	11-13	11-13	11-13		
								11-14	11-14	12-14	12-14	12-14	12-14		
								12-15	12-15	13-15	13-15	13-15	13-15		
								13-16	13-16	14-16	14-16	14-16	14-16		
								14-17	14-17	15-17	15-17	15-17	15-17		
								15-18	15-18	16-18	16-18	16-18	16-18		
								16-19	16-19	17-19	17-19	17-19	17-19		
								17-20	17-20	18-20	18-20	18-20	18-20		
								18-21	18-21	19-21	19-21	19-21	19-21		
								19-22	19-22	20-22	20-22	20-22	20-22		
								20-23	20-23	21-23	21-23	21-23	21-23		
								21-24	21-24	22-24	22-24	22-24	22-24		
								22-25	22-25	23-25	23-25	23-25	23-25		
								23-26	23-26	24-26	24-26	24-26	24-26		
								24-27	24-27	25-27	25-27	25-27	25-27		
								25-28	25-28	26-28	26-28	26-28	26-28		
								26-29	26-29	27-29	27-29	27-29	27-29		
								27-30	27-30	28-30	28-30	28-30	28-30		
								28-31	28-31	29-31	29-31	29-31	29-31		
								29-32	29-32	30-32	30-32	30-32	30-32		
								30-33	30-33	31-33	31-33	31-33	31-33		
								31-34	31-34	32-34	32-34	32-34	32-34		
								32-35	32-35	33-35	33-35	33-35	33-35		
								33-36	33-36	34-36	34-36	34-36	34-36		
								34-37	34-37	35-37	35-37	35-37	35-37		
								35-38	35-38	36-38	36-38	36-38	36-38		
								36-39	36-39	37-39	37-39	37-39	37-39		
								37-40	37-40	38-40	38-40	38-40	38-40		
								38-41	38-41	39-41	39-41	39-41	39-41		
								39-42	39-42	40-42	40-42	40-42	40-42		
								40-43	40-43	41-43	41-43	41-43	41-43		
								41-44	41-44	42-44	42-44	42-44	42-44		
								42-45	42-45	43-45	43-45	43-45	43-45		
								43-46	43-46	44-46	44-46	44-46	44-46		
								44-47	44-47	45-47	45-47	45-47	45-47		
								45-48	45-48	46-48	46-48	46-48	46-48		
								46-49	46-49	47-49	47-49	47-49	47-49		
								47-50	47-50	48-50	48-50	48-50	48-50		
								48-51	48-51	49-51	49-51	49-51	49-51		
								49-52	49-52	50-52	50-52	50-52	50-52		
								50-53	50-53	51-53	51-53	51-53	51-53		
								51-54	51-54	52-54	52-54	52-54	52-54		
								52-55	52-55	53-55	53-55	53-55	53-55		
								53-56	53-56	54-56	54-56	54-56	54-56		
								54-57	54-57	55-57	55-57	55-57	55-57		
								55-58	55-58	56-58	56-58	56-58	56-58		
								56-59	56-59	57-59	57-59	57-59	57-59		
								57-60	57-60	58-60	58-60	58-60	58-60		
								58-61	58-61	59-61	59-61	59-61	59-61		
								59-62	59-62	60-62	60-62	60-62	60-62		
								60-63	60-63	61-63	61-63	61-63	61-63		
								61-64	61-64	62-64	62-64	62-64	62-64		
								62-65	62-65	63-65	63-65	63-65	63-65		
								63-66	63-66	64-66	64-66	64-66	64-66		
								64-67	64-67	65-67	65-67	65-67	65-67		
								65-68	65-68	66-68	66-68	66-68	66-68		
								66-69	66-69	67-69	67-69	67-69	67-69		
								67-70	67-70	68-70	68-70	68-70	68-70		
								68-71	68-71	69-71	69-71	69-71	69-71		
								69-72	69-72	70-72	70-72	70-72	70-72		
								70-73	70-73	71-73	71-73	71-73	71-73		
								71-74	71-74	72-74	72-74	72-74	72-74		
								72-75	72-75	73-75	73-75	73-75	73-75		
								73-76	73-76	74-76	74-76	74-76	74-76		
								74-77	74-77	75-77	75-77	75-77	75-77		
								75-78	75-78	76-78	76-78	76-78	76-78		
								76-79	76-79	77-79	77-79	77-79	77-79		
								77-80	77-80	78-80	78-80	78-80	78-80		
								78-81	78-81	79-81	79-81	79-81	79-81		
								79-82	79-82	80-82	80-82	80-82	80-82		
								80-83	80-83	81-83	81-83	81-83	81-83		
								81-84	81-84	82-84	82-84	82-84	82-84		
								82-85	82-85	83-85	83-85	83-85	83-85		
								83-86	83-86	84-86	84-86	84-86	84-86		
								84-87	84-87	85-87	85-87	85-87	85-87		
								85-88	85-88	86-88	86-88	86-88	86-88		
								86-89	86-89	87-89	87-89	87-89	87-89		
								87-90	87-90	88-90	88-90	88-90	88-90		
								88-91	88-91	89-91	89-91	89-91	89-91		
								89-92	89-92	90-92	90-92	90-92	90-92		
								90-93	90-93	91-93	91-93	91-93	91-93		
								91-94	91-94	92-94	92-94	92-94	92-94		
								92-95	92-95	93-95	93-95	93-95	93-95		
								93-96	93-96	94-96	94-96	94-96	94-96		
								94-97	94-97	95-97	95-97	95-97	95-97		
								95-98	95-98	96-98	96-98	96-98	96-98		
								96-99	96-99	97-99	97-99	97-99	97-99		
								97-100	97-100	98-100	98-100	98-100	98-100		
								98-101	98-101	99-101	99-101	99-101	99-101		
								99-102	99-102	100-102	100-102	100-102	100-102		
								100-103	100-103	101-103	101-103	101-103	101-103		
								101-104	101-104	102-104	102-104	102-104	102-104		
								102-105	102-105	103-105	103-105	103-105	103-105		
								103-106	103-106	104-106	104-106	104-106	104-106		
								104-107	104-107	105-107	105-107	105-107	105-107		
								105-108	105-108	106-108	106-108	106-108	106-108		
								106-109	106-109	107-109	107-109	107-109	107-109		
								107-110	107-110	108-110	108-110	108-110	108-110		
								108-111	108-111</						

Метка	Адрес	Операция ЦП (F)	Управл. FD	Шина K	Управление адресом ВМУ	След. адрес	Прием определ.	Голосность	Примечание
1	2	3	4	5	6	7	8	9	10
B1	0-15	NOP	FF0	0000	JPX		HCE	R	выдача R1 в A (R1, R2)
B12, B13	0-0	ILR (R1)	FF0	0000	JCR	0-15	HCE	NP	выдача R2 в A (R3, R4)
B14, B15	0-1	ILR (R2)	FF0	0000	JCR	0-15	HCE	NP	выдача R3 в A (байт сетевой)
B17	0-2	ILR (R5)	FF0	0000	JCR	0-15	HCE	NP	прием M1, B, X
B8	0-4	LDI (A)	FF0	00FF	JCR	0-15	HCE	NP	прием M1, B, A
B9	0-5	ORI (A)	FF0	FF00	JCC	1-3	HCE	NP	запись X в R1
	1-5	SDR (R1)	FF1	FFFF	JER	0-15	HCE	NP	прием M1, B, A
B10	0-6	LDI (A)	FF1	00FF	JCR	0-15	HCE	NP	прием M1, B, Y
B11	0-7	ORI (A)	FF0	FF00	JCC	1-7	HCE	NP	прием M1, B, Y
	1-7	SDR (R3)	FF1	FFFF	JER	0-15	HCE	NP	запись Y в R3
B16	0-8	NOP	FF0	0000	JCC	1-8	HCE	NP	переход к умножению
M2	1-8	CLA	FF0	0000	JCC	2-8	STZ	NP	A := 0; T2 := 0
	2-8	SDR (R2)	FF1	FFFF	JCR	2-9	HCE	NP	R2 := 0
	2-9	SDR (R5)	FF1	FFFF	JCC	1-9	HCE	NP	R5 := 0
M3	1-9	TER (R1)	FF0	FFFF	JFL	1-10, 11	HCE	NP	R = 0000
M4	1-11	ILR (R3)	FF0	0000	JCR	1-12	HCE	NP	A := R3
	1-12	SDR (R4)	FF1	FFFF	JCR	1-13	HCE	NP	R4 := A
	1-13	CMR (R4)	FF1	0000	JCR	1-14	HCE	NP	R4 := R4 + 1
	1-14	JLR (R4)	FF1	0000	JCC	2-14	HCE	NP	A := -16 (прием M1, M)
M5	2-14	LLM (A)	FF0	FFFF	JCR	2-13	HCE	NP	R5 := A
M6	2-13	SDR (R5)	FF1	FFFF	JCR	3-12, 11	HCE	NP	проверка: FI = 0
	2-12	NOP	FF0	0000	JEF	3-9	HCE	NP	A := R1
M7	3-10	JLR (R1)	FF0	0000	JCR	4-10, 11	STZ	NP	проверка M1, R, X, Y, запись в FI
	3-9	SRA	FF0	0000	JFL				

Таблица 4.3 (продолжение)									
1	2	3	4	5	6	7	8	9	10
M9	4-10	NOP	FF0	0000	JCR	4-8	HCE	NR	Переход к M11 (M11)
M11	4-11	JLR(R4)	FF0	0000	JCR	4-9	HCE	NR	A:=R4
	4-9	ALR(R2)	FF0	FFFF	JCR	4-8	HCE	NR	R2:=A+R2
	4-8	TLR(A)	FF0	0000	JLC	3-8	STC	NR	Замещение знака в FC
	3-8	JLR(R2)	FF0	0000	JCR	3-13	HCE	NR	A:=R2
	3-13	SRA	FFC	0000	JCC	3-14	STC	NR	Сдвиг
M12	3-14	JLR(R2)	FF1	FFFF	JCR	4-14	HCE	NR	R2:=A
	4-14	TLR(R1)	FF0	0000	JCR	4-13	HCE	NR	A:=R1
	4-13	SRA	FFC	0000	JCR	4-12	HCE	NR	Сдвиг
	4-12	SDR(R1)	FF1	FFFF	JCC	5-12	HCE	NR	R1:=A
	5-12	JLR(R5)	FF1	0000	JCR	5-13	HCE	NR	R5:=R5+1
M13	5-13	TER(R5)	FF0	FFFF	JCR	2-10,11	HCE	NR	R5:=0000
	2-10	NOP	FF0	0000	JFL	1-10	HCE	NR	переход к M13 (M13)
	2-11	NOP	FF0	0000	JCC	3-12	HCE	NR	переход к M6 (M6)
	3-11	ILR(R1)	FF0	0000	JCR	5-10,11	STC	NR	A:=R1
	5-11	SRA	FF0	0000	JF1	5-8	HCE	NR	Проверка на нх, запись в FC
M10	5-10	NOP	FF0	0000	JCR	4-8	HCE	NR	переход к M11 (M11)
	5-9	JLR(R5)	FF0	0000	JCR	5-1	HCE	NR	A:=R5
	4-10	ALR(R2)	FF1	FFFF	JCR	5-8	HCE	NR	R2:=R2+A
	1-15	JLR(R2)	FF0	0000	JCR	1-15	HCE	NR	A:=R2
	2-15	SDR(R6)	FF1	FFFF	JCR	2-15	HCE	NR	R6:=A
M14	2-15	CLR(R6)	FF0	0000	JCC	3-15	HCE	NR	R6:=R6 (=R2)
	3-15	TER(R6)	FFC	FFFF	JFL	5-10,11	HCE	NR	R6:=0000
	6-11	NOP	FF0	0000	JCR	5-9	HCE	NR	переход к M16 (M16)
	6-12	TER(R2)	FF0	0000	JCR	5-12	HCE	NR	переход к M15 (M15)
	6-9	TLR(R1)	FF0	FFFF	JFL	8-10,11	HCE	NR	R2:=0000
M17	7-10	NOP	FF0	0000	JFL	7-10,11	HCE	NR	R1:=0000
	7-12	NOP	FF0	0000	JCR	7-12	HCE	NR	переход к M15 (M15)
	8-10	TER(R1)	FF0	FFFF	JCC	6-12	HCE	NR	R1:=0000
	9-11	NOP	FF0	0000	JFL	9-10,11	HCE	NR	переход к M15 (M15)
	7-8	TLR(R5)	FF1	0000	JCR	7-11	HCE	NR	R1:=0000
M18	9-10	JLR(R5)	FF1	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
	9-11	TLR(R5)	FF1	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
	9-10	JLR(R5)	FF1	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
	9-11	NOP	FF0	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
	9-11	NOP	FF0	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
M19	9-10	JLR(R5)	FF1	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
	9-11	TLR(R5)	FF1	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
	9-10	JLR(R5)	FF1	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
	9-11	NOP	FF0	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
	9-11	NOP	FF0	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
M20	9-10	JLR(R5)	FF1	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
	9-11	TLR(R5)	FF1	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
	9-10	JLR(R5)	FF1	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
	9-11	NOP	FF0	0000	JCR	0-15	HCE	NR	переход к M18 (M18)
	9-11	NOP	FF0	0000	JCR	0-15	HCE	NR	переход к M18 (M18)

для выполнения операции умножения должен отсутствовать - в поле готовности записываем NR. Заполнение других полей выполняется в соответствии с ранее составленной предварительной микропрограммой.

Учитывая введение промежуточные микрокоманды, можно рассчитать окончательно суммарное количество микрокоманд, необходимых для умножения. Оно составит $10 + 16 \times 16 + 14 = 280$ микрокоманд, следовательно, время выполнения операции будет 140 мкс при тактовой частоте 2 МГц.

4.8. Синтез схем модуля

Перейдем к разработке функциональных схем блоков и узлов модуля.

4.8.1. Постоянное ЗУ микрокоманд

Для построения ЗУ микрокоманд удобно использовать микросхемы K556PT5, представляющие собой ПЗУ с организацией 512×8 разрядов. Так как микрокоманда модуля содержит 22 разряда, необходимо поставить 3 микросхемы. Схема ПЗУ приведена на рис. 4.10. Десятиразрядный адрес $MA_0 - MA_8$ с БМУ параллельно подается на адресные входы микросхем. На входы выборки кристалла поданы разрешающие уровни, поэтому 22-разрядное слово микрокоманды появится на выходах всех трех микросхем через 70 нс после поступления адреса. Выходы микросхем ПЗУ серии K556 выполнены с открытым коллектором. Для согласования по уровням с нагрузочными микросхемами необходимо определить величину резистора R_1 . Сопротивление R_1 зависит от минимального и максимального значений допустимых уровней согласования на выходах БИС ПЗУ. Максимальное значение R_1 определяется из режима, когда все выходы микросхемы имеют высокий уровень /16/:

$$R_{1\max} = \frac{E_{\min} - U_{1\max}^1}{I_{1\max}^1 + n \cdot I_{1\max}^1}$$

где: $E_{\min}$ - минимальное напряжение питания (4,75 В);

n - количество нагружаемых входов, примем $n = 8$ (массив ЦПЗ имеет 8 входов управления функцией);

$U_{1\min}^1$ - минимальный уровень логической единицы схемы нагрузки (2,4 В);

$I_{1\max}^1$ - выходной ток логической единицы БИС ПЗУ (0,1 мА);

$I_{1\max}^1$ - входной ток логической единицы схемы нагрузки (0,04 мА).

Следовательно:

$$R_{1\max} = \frac{4,75 - 2,4}{0,1 - 0,04} = \frac{2,35}{0,42} = 5,6 \text{ кОм},$$

Схема ПЗУ микрокоманд

MA0 1

MA1 2

MA2 3

MA3 4

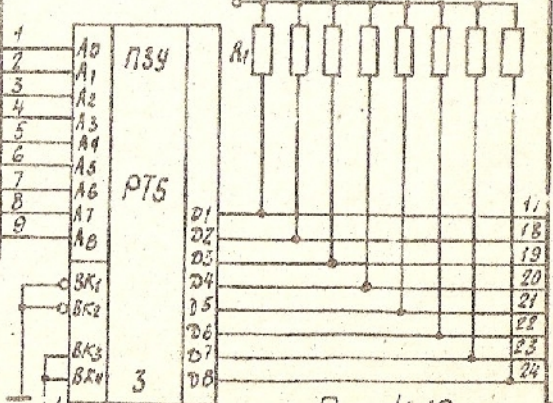
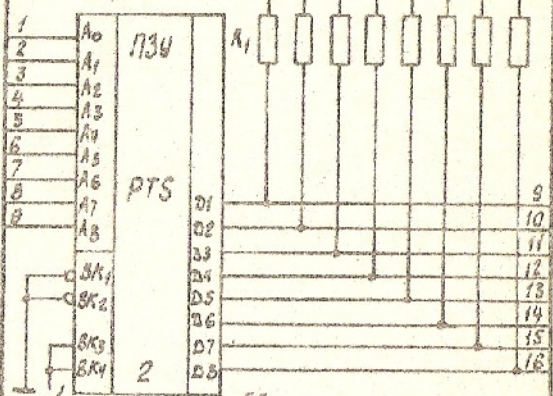
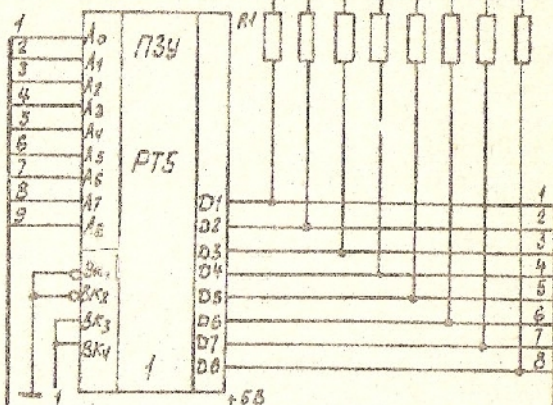
MA4 5

MA5 6

MA6 7

MA7 8

MA8 9



1 МК0

2 МК1

3 МК2

4 МК3

5 МК4

6 МК5

7 МК6

8 МК7

9 МК8

10 МК9

11 МК10

12 МК11

13 МК12

14 МК13

15 МК14

16 МК15

17 МК16

18 МК17

19 МК18

20 МК19

21 МК20

22 МК21

23 МК22

24 МК23

Рис. 4.10

Минимальное значение R_1 находится из условия, когда выход ИЭУ переходит в нижний уровень:

$$R_{1\min} = \frac{E_{п\max} - U^0_{\max}}{I_{\text{вых}}^0 - n \cdot I_{\text{вх}}^0},$$

где: $U_{п\max}$ — максимальное напряжение питания (5,25 В);

$U^0_{\max}$ — максимальный уровень логического нуля схемы нагрузки (0,4 В);

$I_{\text{вых}}^0$ — выходной ток логического нуля ИС ИЭУ (15 мА);

$I_{\text{вх}}^0$ — входной ток логического нуля схемы нагрузки (1,6 мА).

Расчет дает

$$R_{1\min} = \frac{5,25}{15} - \frac{0,4}{8 \cdot 1,6} = \frac{4,85}{2,2} = 2,2 \text{ кОм}.$$

Выбираем окончательно $R_1 = 3,3 \text{ кОм}$.

Если при расчете $R_1 < 140 \Omega$, то необходимо включить дополнительное сопротивление, величина которого определяется из неравенства:

$$\frac{U^1_{\min}}{E_{п\min} - U^1_{\min}} R_1 \leq R_2 \leq \frac{U^1_{\max}}{E_{п\max} - U^1_{\max}} R_1.$$

Емкость разработанного ИЭУ (512 микрокоманд) значительно больше, чем объем микропрограммы работы умножителя (60 микрокоманд). Это позволяет в дальнейшем достаточно просто расширять функции модуля.

4.8.2. Буфер адреса и сигналов управления

На вход схемы с шины микро-ЭВМ поступает три сигнала управления $\overline{IOWC}$, $\overline{IORC}$ и $\overline{INIT}$ и младший байт адреса. Схема необходимая для согласования сигналов шины с нагрузкой — не более одного стандартного TTL входа. Построим схему на базе элементов К155ЛН1, в которой содержится шесть инверторов. Схема буфера адреса и сигналов управления приведена на рис. 4.11. Здесь

- сигнал записи $WR = \overline{\overline{IOWC}}$;
- сигнал чтения $RD = \overline{\overline{IORC}}$; $\overline{RD} = \overline{RD}$;
- сигнал сброса $RS = \overline{\overline{INIT}}$.

4.8.3. Схема выборки

Сигнал выборки модуля MS , генерируемый схемой, должен принимать единичное значение при обращении к модулю. Следовательно, сигнал должен быть активным, когда на адресных линиях шины микро-ЭВМ установлена одна из комбинаций

$$IO = IIII \ 0000$$

$$PI = IIII \ 0001$$

Буфер адреса и сигналов управления,
Формирователь сигнала выборки

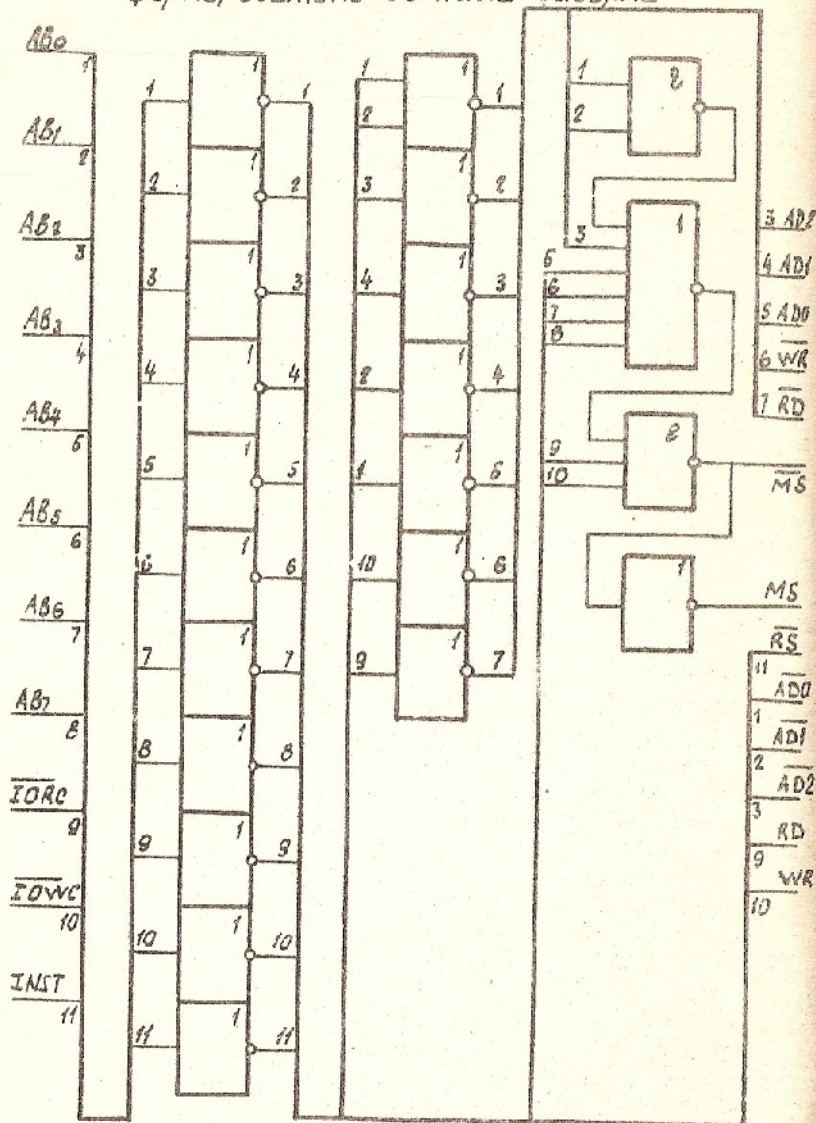


Рис. 4.11

F2 = IIII 0010

F3 = IIII 0011

F4 = IIII 0100

и присутствует либо сигнал чтения ($RD=1$), либо сигнал записи ($WR=1$). Следовательно,

$$MS = RD \cdot WR \cdot AD_7 \cdot AD_6 \cdot AD_5 \cdot AD_4 \cdot \overline{AD_3} (\overline{AD_2} + AD_2 \cdot \overline{AD_1} \cdot \overline{AD_0}).$$

Это выражение легко преобразуется к виду

$$MS = \overline{WR \cdot RD \cdot S_1},$$

$$S_1 = \overline{AD_7 + AD_6 + AD_5 + AD_4 + AD_3 + S_2},$$

$$S_2 = \overline{AD_2 \cdot \overline{AD_1} \cdot \overline{AD_0}}.$$

Схема, построенная по этим соотношениям на элементах серии К155, приведена на рис.4.11.

4.8.4. Буфер шины данных

Буфер шины данных, связывающий двунаправленные линии данных шины микро-ЭВМ с входными I и выходными D линиями массива ЦПЭ удобно строить на элементах К589АП6 или К589АП26 /4,12/. Так как входы I и выходы D ЦПЭ инверсные, в сигналах данных на шине микро-ЭВМ прямые /3/, лучше использовать микросхемы К589АП26, которые обеспечивают инвертирование сигналов. Для передачи байта данных необходимо применять два параллельно включенных четырехразрядных шинных формирователя. Схема буфера приведена на рис.4.12.

Шинный формирователь К589АП26 управляется с помощью двух входов. Вход ВК - выборка кристалла - определяет состояние формирователя, причем высокий уровень переводит его в третье состояние. В нашей схеме подключение модуля к шине данных микро-ЭВМ должно происходить при $MS=1$, следовательно, $BK = \overline{MS}$ (вход ВК - инверсный). Направление передачи сигналов выбирается уровнем на входе УВ. Если на входе УВ низкий уровень, информация о входов А передается на входы/выходы В формирователя, если уровень высокий, информация передается о входов/выходов В на выходы С. Для управления входами УВ используем сигнал $\overline{RD}$. Тогда при выборке модуля формирователь настроится на передачу данных с линий $\overline{D_B}$ шины микро-ЭВМ на входы I ЦПЭ. И только при чтении данных ($\overline{RD}=0$) - формирователь подключит выходы D ЦПЭ к линиям $\overline{D_B}$ шины микро-ЭВМ.

4.8.5. Формирователь адреса перехода

Формирователь адреса перехода должен по импульсному сигналу вы-

Подключение системной шины данных
к входам и выходам Д ЦПЗ

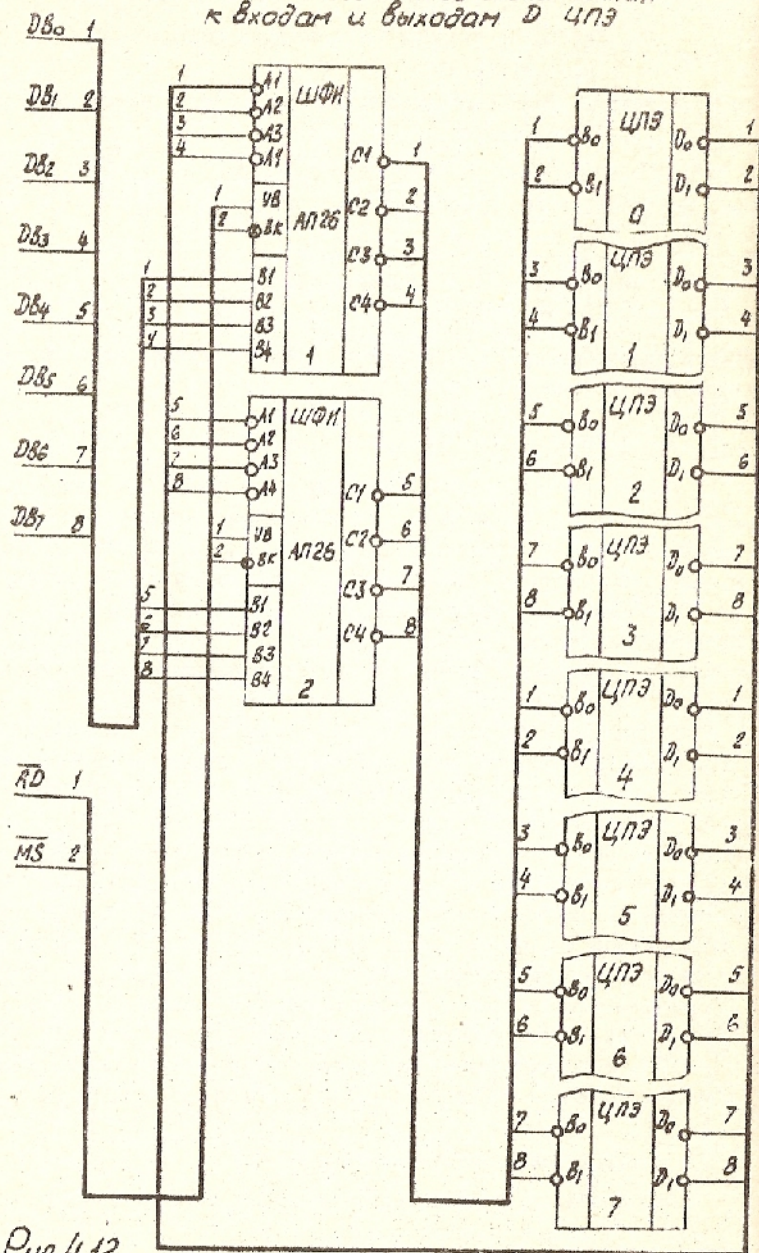


Рис 4.12

Борис MSI видать на входы команды $K_7 - K_4$ БМУ адрес ячейки в нулевой строке, из которой необходимо выбрать следующую микрокоманду. Этот адрес определяется, во-первых, адресом обращения к модулю (комбинацией сигналов на адресных линиях шины микро-ЭВМ) и, во-вторых, наличием сигнала чтения RD или записи WR . Соответствие адресов приведено в табл. 4.4.

Например, при обращении центрального процессора к модулю по адресу F2 при операции записи ($WR=1$) на выходе формирователя должна быть комбинация 0110 - переход к ячейке 0-6, где записана микрокоманда BIO приема младшего байта Y в аккумулятор ЦПЭ. Если $WR=0$, то есть $RD=1$ (операция чтения), должна быть сформирована комбинация 0001, по которой произойдет переход к ячейке 0-1, где записана микрокоманда BI4 чтения четвертого байта результата.

Если $MSI=0$ (нет выбора модуля), то на выходе формирователя должен быть код 1111, задающий переход к ячейке 0-15, что задвигает выполнение микрокоманды BI, ожидания обращения.

Анализ табл. 4.4 позволяет записать следующие выражения для выходов формирователя:

$$K_7 = MSI + MSI \cdot WR \cdot AD_2 \cdot AD_1 \cdot AD_0;$$

$$K_6 = MSI + MSI \cdot WR \cdot AD_2;$$

$$K_5 = MSI + MSI \cdot (WR \cdot AD_2 \cdot AD_1 \cdot AD_0 + WR \cdot AD_2 \cdot AD_1 \cdot AD_0 + WR \cdot AD_2 \cdot AD_1 \cdot AD_0);$$

$$K_4 = MSI + MSI \cdot (WR \cdot AD_2 \cdot AD_1 \cdot AD_0 + WR \cdot AD_2 \cdot AD_1 \cdot AD_0 + WR \cdot AD_2 \cdot AD_1 \cdot AD_0 + WR \cdot AD_1 \cdot AD_2 \cdot AD_0).$$

Приведенные выражения можно упростить, и, с учетом того, что входы $K_7 - K_0$ БМУ инверсные, они принимают вид:

$$\overline{K_7} = MSI \cdot WR \cdot AD_2;$$

$$\overline{K_6} = MSI \cdot WR \cdot AD_2;$$

$$\overline{K_5} = MSI \cdot WR \cdot AD_2 \cdot AD_1 \cdot AD_0 + \overline{WR} \cdot AD_2;$$

$$\overline{K_4} = MSI \cdot WR \cdot AD_2 \cdot AD_0 + \overline{WR} \cdot AD_2 \cdot AD_1.$$

Схема формирователя, построенная в соответствии с этими выражениями приведена на рис. 4.13. Отметим, что состояние входов $K_3 - K_0$ БМУ в схеме модуля безразлично, поэтому на них постоянно подан нулевой потенциал.

4.8.6. Формирователь шины K

Пять различных комбинаций на выходе формирователя задаются с помощью трехразрядного поля управления шиной K микрокоманды ($MK_7 - MK_9$). Кодировка комбинаций приведена в таблице на рис. 4.4, причем коды $MK_7 - MK_9$ подобраны таким образом, чтобы схема формирователя была простейшей.

Формирование номера колонки перехода

Таблица 4.4

Адрес				MSI	Номер колонки перехода				Примечание
16% ₁₀	2 система				8 система				
	AD ₂	AD ₄	AD ₆		K ₇	K ₆	K ₅	K ₄	
F0	0	0	0	0	0	0	0	0	Чтение байта 1 результата (R1)
F0	0	0	0	0	1	0	0	4	Запись младшего байта X
F1	0	0	1	0	0	0	0	0	Чтение байта 2 результата (R1)
F1	0	0	1	1	0	1	0	1	Запись старшего байта X
F2	0	1	0	0	0	0	0	1	Чтение байта 3 результата (R2)
F2	0	1	0	1	0	1	1	0	Запись младшего байта Y
F3	0	1	1	0	0	0	0	1	Чтение байта 4 результата (R3)
F3	0	1	1	1	0	1	1	1	Запись старшего байта Y
F4	1	0	0	0	0	0	1	0	Чтение байта 1 результата (R3)
F4	1	0	0	1	1	0	0	0	Пуск умножения
X	X	X	X	X	1	1	1	1	Возврат в колонку 15

Схема формирования адреса перехода

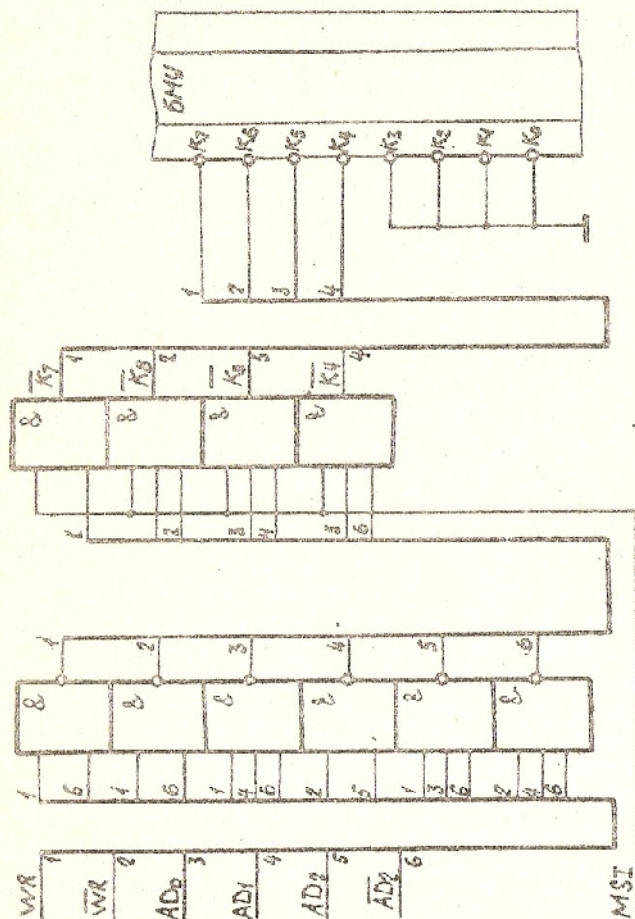


Рис. 4.13

В соответствии с таблицей, все входы N младшего байта массива ЦПЭ (ЦПЭ N 0,1,2,3) управляются разрядом $МК_9$ микрокоманды. Все входы K старшего байта массива ЦПЭ (ЦПЭ N 4,5,6,7), кроме K_1 ЦПЭ 7, управляются разрядом $МК_9$ микрокоманды. Выход K_1 ЦПЭ 7 управляется разрядом $МК_7$. Схема формирователя приведена на рис.4.14.

С входов M в массив ЦПЭ считывается константа -16 для задания количества циклов в операции умножения. В двоичной системе счисления в дополнительном коде число -16 записывается как 111111110000 . Так как входы M инверсные, на них подается комбинация 000000001111 . На рис.4.14 приведена также схема формирования входов M массива ЦПЭ.

4.8.7. Формирователь сигналов ED, CI

При чтении информации из модуля необходимо выдавать в буфер шин данных либо младший байт регистра, либо старший байт регистра. Причем, если выполняется операция чтения с адресами $F0, F2, F4$, требуется передать младший байт, если с адресами $F1, F3$ - старший байт. Выбор байта осуществляется включением выходного буфера данных в соответствующей половине массива ЦПЭ. Следовательно, выходной буфер данных младшего байта массива ЦПЭ должен быть включен только при чтении из модуля с адресами $F0, F2, F4$, то есть:

$$ED_{ml} = MS \cdot RD \cdot \overline{AD}$$

или (так как вход управления буфером инверсный):

$$ED_{ml} = MS \cdot RD \cdot \overline{AD} \cdot \overline{AD}.$$

Выходной буфер шин данных старшего байта массива ЦПЭ должен включаться при операциях чтения с адресами $F1, F3$. Следовательно,

$$ED_{st} = MS \cdot RD \cdot \overline{AD} \cdot AD = \overline{ED_{ml}}$$

Схема формирования сигналов управления выходным буфером массива ЦПЭ приведена на рис.4.15.

При операциях записи младших байтов X и Y в ЦПЭ должна разрываться связь по сигналу переноса между младшими и старшими байтами ЦПЭ. Это необходимо для обнуления старшего байта аккумулятора при приеме в младший байт (см. микрофункцию LDI). Причем на входе CI старшего байта ЦПЭ должна быть 1.

Запись в младший байт (адреса $F0, F2$) происходит когда истинная комбинация

$$MS \cdot WR \cdot \overline{AD2} \cdot \overline{AD}.$$

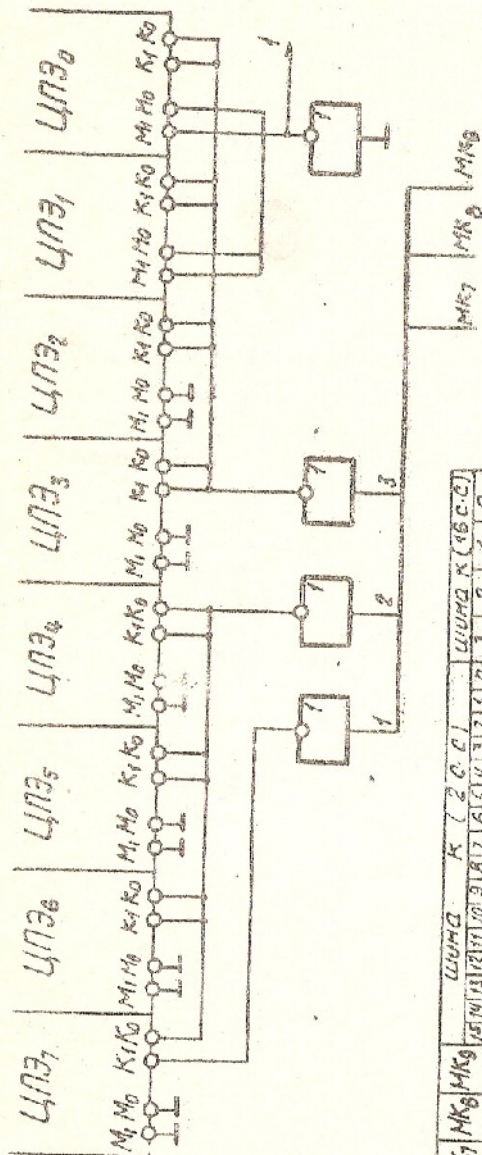
Тогда для входа CI старшего байта ЦПЭ можно записать

$$CI = C0 + MS \cdot WR \cdot \overline{AD2} \cdot \overline{AD}.$$

Так как и вход CI и выход $C0$ ЦПЭ инверсные, предыдущее выражение удобно переписать в виде

$$CI = \overline{C0} \cdot (MS \cdot WR \cdot \overline{AD2} \cdot \overline{AD}).$$

Формирование шин K и M массива ЦПЭ



MK ₇	MK ₈	MK ₉	UUMQ K (2001)										UUMQ K (4000)				
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Рис. 4.14

4.9. Оценка параметров синхронизирующего сигнала

Расчет периода и длительности тактового импульса выполним по методике из /4, 12, 21/. Для этого рассмотрим временную диаграмму выполнения микрокоманды, приведенную на рис. 4.16.

По переднему фронту синхримпульса С сформированный в БМУ соответствующим образом адрес следующей микрокоманды принимается в регистр адреса и с задержкой t_{co} появляется на выходе БМУ. Затем срабатывают элементы ЦЗУ и через время $t_{пзз}$ на информационных выходах

Д микросхем устанавливается новая микрокоманда. Семь разрядов поля управления операцией микрокоманды поступают на входы F микро-схем ЦПЭ, который начинает выполнять заданную микрофункцию. При этом наиболее важным является правильное формирование переноса. Выход переноса младшего (нулевого) ЦПЭ установится с задержкой

$$t_1 = \max \{ t_{cf}, t_{co} + t_{cc} \},$$

где: t_{cf} — задержка сигнала СО относительно входов F ЦПЭ;

t_{co} — задержка выхода F0 БМУ относительно входов F_{c1}, F_{c2} ;

t_{cc} — задержка сигнала СО относительно входов СИ ЦПЭ.

Далее происходит перенос через оставшиеся три микросхемы младшего байта массива ЦПЭ, время распространения составит $3 \times t_{cc}$.

Через схему формирования сигнала СИ (задержка t_{fc}) перенос поступит на вход СИ старшего байта массива ЦПЭ и с задержкой $4 \times t_{cc}$ появится на входе FI БМУ. Сигнал на входе FI должен быть установлен не менее чем за t_{st} до появления заднего фронта синхронизирующего импульса. Следовательно, длительность тактового импульса должна быть не менее

$$t_{co} + t_{пзз} + t_1 + 3 \cdot t_{cc} + t_{fc} + 4 \cdot t_{cc} + t_{st}.$$

В соответствии со справочными данными (типичные значения):

$$t_{co} = 30 \text{ нс};$$

$$t_{пзз} = 70 \text{ нс};$$

$$t_1 = \max(43, 30 - I_4) = 44 \text{ нс};$$

$$t_{cc} = 14 \text{ нс};$$

$$t_{fc} = 22 \text{ нс};$$

$$t_{st} = 5 \text{ нс}.$$

Тогда, длительность импульса $t_c \geq 269 \text{ нс}$. Время паузы t_{wp} должно быть не менее $33 \text{ нс} / 21/$. Таким образом, минимальный период тактовых импульсов $t_{cy} = 302 \text{ нс}$, максимальная частота $3,3 \text{ МГц}$. При такой частоте производительность модуля составит $3\,300\,000$ микроко-

Цепи обвязки и переноса, управление
выходом данных массива ЦПЗ

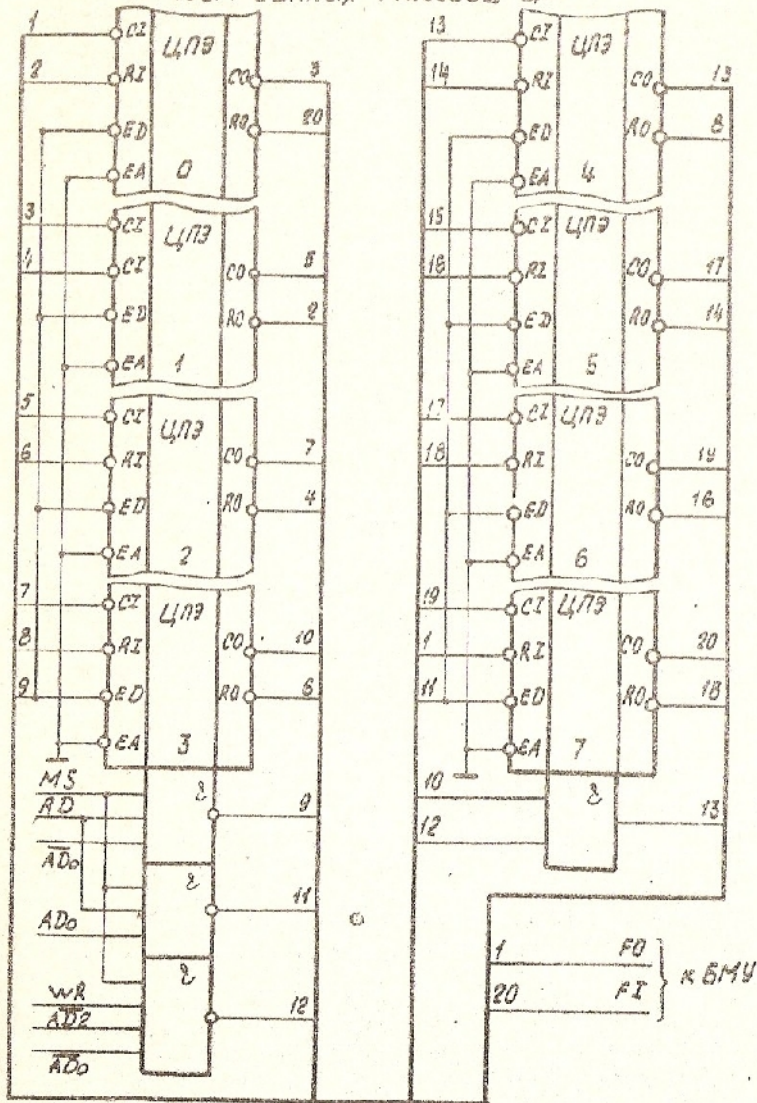


Рис. 4.15

нанд в секунду, а производительность по операциям умножения - около 11000 оп/с.

Для получения тактовой частоты 3,3 МГц необходимо строить специальный генератор. Проще ограничиться имеющимися тактовыми сигналами из шины микро-ЭВМ. Удобнее всего воспользоваться инверсной сигналом $\phi 1$. Его временные характеристики согласно /3/ составляют (см. рис. 4.17,а):

$$t_{cy} = 500 \text{ нс};$$

$$t_{wp} = 115 \text{ нс};$$

$$t_c = 385 \text{ нс}.$$

Эти параметры удовлетворяют требованиям к синхросигналу модуля умножения, однако производительность модуля снижается до 2 000 000 микроопераций в секунду. При этом операция умножения выполняется за 14. мкс, что отвечает поставленным требованиям. Собственно схема формирования синхросигнала представляет собой один мощный элемент ливерони (рис. 4.17,б).

4.10. Формирование импульсного сигнала выборки

Формирователь адреса перехода выдает код на входе команды БМУ по импульсному сигналу выборки, длительность которого должна быть достаточной для правильного формирования в БМУ адреса следующей микрокоманды. Временная диаграмма формирования сигнала MSI приведена на рис. 4.17,а. Он должен появиться при поступлении сигнала MS, но не менее, чем за время t_{sx} до переднего фронта синхроимпульса. После поступления переднего фронта синхроимпульса сигнал MSI должен сохраняться не менее t_{nx} наносекунд. Для БМУ эти времена составляют:

$$t_{sx} = 35 \text{ нс},$$

$$t_{nx} = 20 \text{ нс}.$$

С целью правильного формирования сигнала готовности потребуем, чтобы MSI завершался после поступления 1-вой микрокоманды, то есть

$$t_2 = \max(t_{nx}, t_{co} + t_{mz}) = 100 \text{ нс}.$$

Запуск одновибратора удобно осуществлять по заднему фронту синхроимпульса при наличии сигнала MS. Тогда, длительность импульса MSI составит

$$t_{mz} = t_1 + t_2 = 115 \text{ нс} + 100 \text{ нс} = 215 \text{ нс}.$$

Для формирования MSI воспользуемся микросхемой KI55AF3. Схема формирователя приведена на рис. 4.17,б. Одновибратор запускается задним фронтом сигнала

$$A = \overline{MS} \cdot C.$$

5000

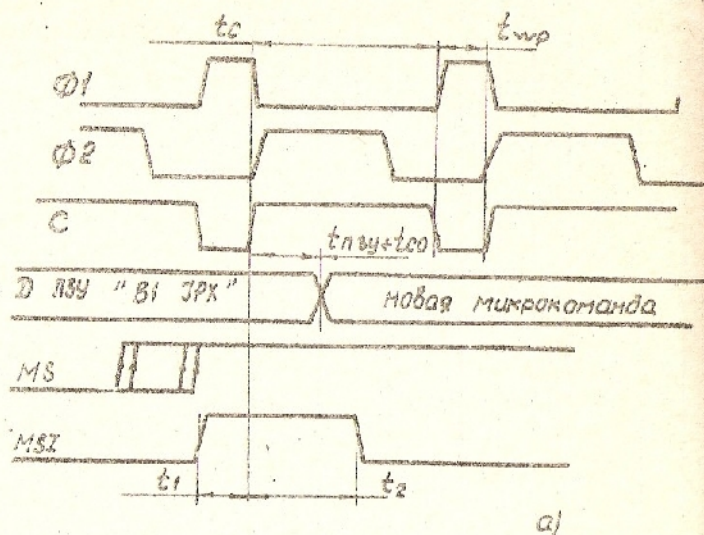
$$+ \alpha_4 > t_{wB} + t_{CO} + t_{NB} + t_I + 3t_{CC} + t_{\phi.C} + 4t_{CC} + t_{SI} = t_{wP} + t_{CB} + t_{NY} + t_{NO} + 8t_{CO} + t_{\phi.B.} + t_{SC}$$
$$I_{\text{M}} = \max_j \{x_{Dj} + c_j - z_j\}$$

$t_{OC} = 43 \text{ MC}$; $t_{WC} = 33 \text{ MC}$; $t_{OC} = 22 \text{ MC}$

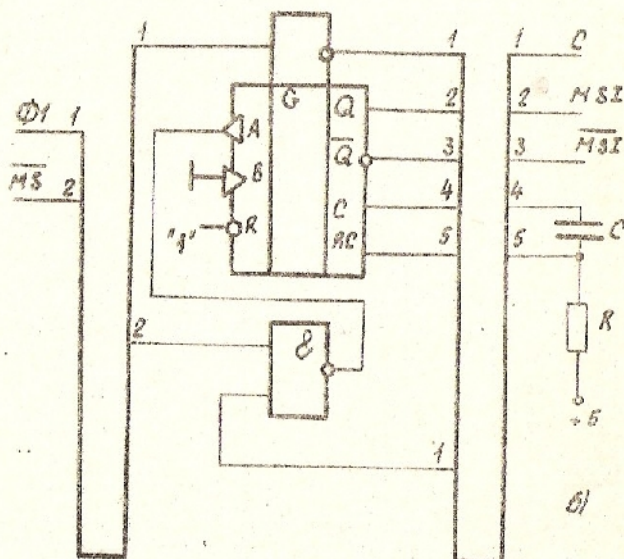
$$t_{KO} = 30 \text{ MC}; \quad t_{KO} = 30 \text{ MC}; \quad t_{KO} = 30 \text{ MC};$$
$$+ 12H_2O = 5H_2 + 7CO + 4H_2O$$

Puc. 4.16

Формирователь MSI



а)



б)

а) к выбору длительности MSI

б) схема формирователя MSI

Рис. 4.17

Длительность импульса задается элементами R и C , расчет которых можно выполнить по формуле

$$\tau_{\text{имп}} = k \cdot R \cdot C \cdot (1 + 0.7/R), \quad k = 0.28$$

(емкость в пикофарадах, сопротивление в килоомах, длительность в наносекундах).

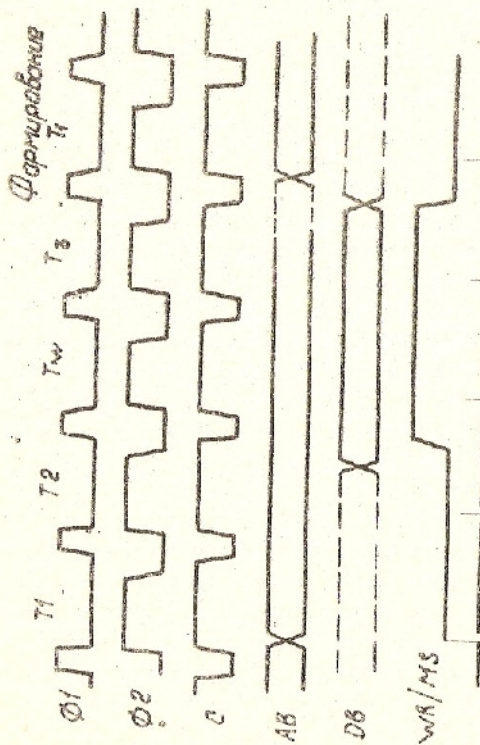
Возьмем $R = 1 \text{ кОм}$. Тогда, для $\tau = 215 \text{ нс}$, величина емкости составит 451 пф . Примем конденсатор стандартного номинала 470 пф , при этом длительность импульса будет 223 нс .

4.11. Формирователь сигнала готовности

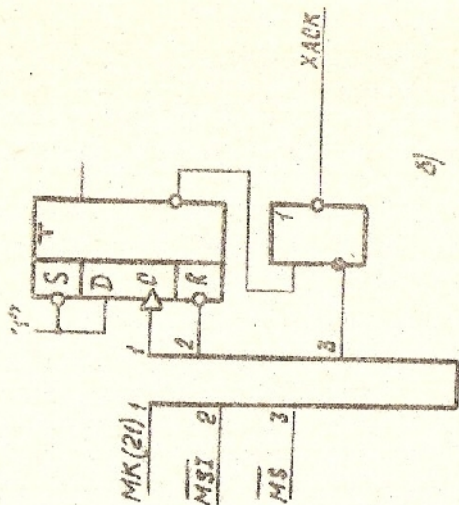
Сигнал готовности модуля подается на линию χACK шины микро-ЭВМ и сообщает центральному процессору, что очередная операция чтения или записи модулем завершена и он готов к продолжению обмена. Следовательно, сигнал готовности должен сниматься на время приема информации в модуль или организации выдачи информации из модуля и на время выполнения операции умножения.

Временная диаграмма обмена байтом данных между модулем и центральным процессором микро-ЭВМ приведена на рис. 4.18, а (на примере старшего бита χ). Микропроцессор 8080 выдает байт для записи в ячейку данных и устанавливает сигнал $\overline{\text{IOWC}}$ (запись), по которому в модуле формируется сигнал выборки $M\bar{S}$. Пока сигнала $M\bar{S}$ нет, модуль выполняет микрооперацию $B\bar{I}$ - ожидания обращения. С появлением $M\bar{S}$ формируется $M\bar{S}\bar{I}$ и код перехода, и в БМУ вырабатывается адрес микрокоманды $B\bar{I}$ приема в аккумулятор старшего бита множителя. Так как длительность C синхросигнала модуля выбрана 500 нс , ЦПЗ может не успеть принять байт в течение одного такта работы микропроцессора 8080. Следовательно, сигнал готовности на это время требуется снять. Восстановить его можно только после завершения микрокоманды $B\bar{I}$. Таким образом, сигнал готовности должен сниматься при появлении сигнала $M\bar{S}\bar{I}$ и восстанавливаться микрокомандой, следующей за микрокомандой приема или выдачи данных. Для этого в поле готовности микрокоманд $B\bar{8}, B\bar{9}, B\bar{10}, B\bar{11}, B\bar{12}, B\bar{14}, B\bar{17}$ заносится код отсутствия сигнала готовности ($N\bar{R}$, см. табл. 4.3).

Схема формирования сигнала готовности выполнена на одном $R\bar{S}$ -триггере и приведена на рис. 4.18, б. Сброс триггера осуществляется сигналом $M\bar{S}\bar{I}$, который подан на R вход триггера. Установка в единицу триггера происходит при появлении высокого уровня в $2\bar{I}$ разряде микрокоманды (поле готовности). Для подключения выхода триггера готовности к линии χACK применен буферный элемент с тремя состояниями. Управление выборкой буфера (соответственно подключение



сигнала готовности



б)

- а. временная диаграмма приема старшего байта у
- б. схема формирования сигнала готовности XACK

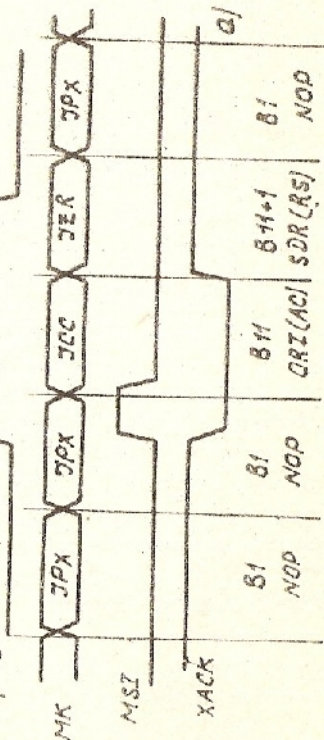


Рис. 4.16

в линии ХАСК) осуществляется сигналом $\overline{MS}$: только после модуль выбран, сигнал готовности подается на линию ХАСК.

4.12. Микропрограмма в двоичных кодах

Для прошивки ЦЗУ разработанная программа переносится в двоичных кодах (см. табл.4.5). В таблице приведен двоичный адрес каждой микрокоманды и ее двоичный 22-разрядный код.

Двоичный адрес получен простой заменой шестнадцатеричных номеров строки и столбца в табл. 4.3 их двоичными эквивалентами. Коды поля управления операцией ЦПЗ, управления адресом БМУ, управления флаговой логики записываются по соответствующим таблицам приложения и mnemonicским обозначениям в табл. 4.3. Код поля управления шиной К формируется на основании таблицы, приведенной на рис.4.14. В поле готовности записывается 1 при наличии сигнала готовности (R) и 0 - при отсутствии готовности ($\overline{MR}$).

Для завершения разработки умножителя необходимо решить следующие вопросы:

- с учетом нагрузочной способности выбрать элементы и составить принципиальную схему модуля;
- проработать конструкцию модуля и монтажные схемы;
- собрать макет модуля и выполнить его отладку;
- скорректировать микропрограмму, принципиальную и монтажную схемы.

Микропрограмма в двойных кодах (фрагмент)

№	Адрес		микрокоманда																					
			Строка	Столбец	F6	F5	F4	F3	F2	F1	F0	шина К	AC0	AC5	AC6	AC3	AC2	AC1	AC0	FC3	FC2	FC1	FC0	Г7
1	000000	00000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	000000	00000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	000000	00010	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	000000	01000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	000000	01001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	000000	01100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	000000	01101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	000000	10000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	000000	11111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	000001	01011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	000001	01111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	000001	10000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	000001	10001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	000001	10100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	000001	10101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	000001	11000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	000001	11001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	000001	11100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	000001	11111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	000010	10000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	000010	10001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	000010	10100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	000010	10101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	000010	11000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	000010	11001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	000010	11100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	000010	11111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0