«Числом гармонию измерить»

ОЦЕНКА И АНАЛИЗ ЭФФЕКТИВНОСТИ АРХИТЕКТУРЫ МИКРОКОНТРОЛЛЕРОВ

Введение

Микроконтроллером (МК) называется интегральная схема, объединяющая в одном кристалле процессор, память и устройства ввода-вывода.

МК обладает тремя свойствами, за счет которых он стал массовым прибором для обработки данных:

– первое, это программное управление, которое позволяет использовать одну и ту же схему для выполнения различных функций;

- второе, это объединение процессора с памятью и устройствами ввода-вывода в одном кристалле, что позволяет строить аппаратуру с минимальным использованием дополнительных схем;

- третье, это возможность программирования схемы самим пользователем под собственные нужды.

Настоящая работа посвящена анализу эффективности архитектуры микроконтроллера на уровне системы команд процессора.

Эффективная архитектура процессора МК должна обеспечивать минимизацию объема кода программы и тактов, затрачиваемых на ее выполнение.

Кроме этого архитектура процессора должна:

а) строиться на основе единообразного использования набора концептуальных принципов построения системы команд, которые обеспечивают расширение разрядности данных и адресного пространства, а также минимальное наличие особых ситуаций;

б) обеспечивать сокращение семантического разрыва с языком программирования;

в) способствовать достижению наилучших характеристик по потреблению и стоимости.

Разные производители микроконтроллеров по разному решают эти задачи. Поэтому на рынке МК одновременно сосуществует множество процессоров с различной архитектурой.

Целью работы является оценка эффективности архитектуры процессора МК и анализ факторов, влияющих на его эффективность.

Первый и второй разделы содержат общую информацию по техническим требованиям к процессору МК и рынку МК.

Третий раздел содержит описание основных архитектур процессора МК.

В четвертом разделе приведена методика оценки эффективности процессора МК на основе статистических характеристик использования операторов языка высокого уровня в типовых программах МК и затрат на их реализацию в машинном коде.

В пятом разделе проведена оценка влияния архитектурных особенностей процессора МК на объем программного кода и тактов на его выполнение.

В шестом разделе предложена концепция архитектуры процессора для микроконтроллера, обеспечивающего существенное снижение объема программного кода и тактов на его выполнение по сравнению с существующими МК.

В заключении даны оценки стоимости аппаратуры и трудоемкости разработки программ МК.

Работа представляет интерес для программистов и разработчиков микроконтроллеров.

1. Технические требования к процессору МК

Областью применения МК являются встроенные системы цифровой обработки данных. Задачи, которые решает МК в этих системах, можно разделить на 3 группы:

- дискретное (релейное) управление;
- аналоговое управление;
- цифровая обработка сигналов.

Задачи дискретного управления охватывают класс цифровых автоматов, счетных устройств и программаторов. Эти задачи характеризуются преобладанием логических операций обработки данных. Объем вычислений незначителен и решается в рамках целочисленной арифметики. Входные и выходные данные поступают и выдаются на устройства релейного типа. Управление осуществляется по событиям, а также по временной диаграмме со временем реакции в доли секунд.

В задачах аналогового управления осуществляется обработка информации от аналоговых датчиков, расчет и выдача управляющих воздействий на исполнительные аналоговые устройства. Требования к точности вычислений определяются точностью датчиков и исполнительных устройств, которая составляет, как правило, 8-12 разрядов. Управление осуществляется с частотой, обеспечивающей необходимое время реакции системы на внешние воздействия в единицы миллисекунд.

В задачах цифровой обработки сигналов над данными выполняется значительный объем вычислений в реальном времени с точностью электронных преобразователей сигналов, которая составляет, как правило, 12-24 разряда. Промежуточные вычисления, в целях предупреждения накопления ошибок, выполняются с более высокой точностью. Обработка данных выполняется на частоте дискретизации сигнала от 10 кГц до 10 мГц и более.

В таблице 1 приведены требования к МК для различных областей применения.

Таблица 1. Требования к МК

Параметр	Логическое управление	Цифровое управление	Цифровая обработка сигналов
Разрядность, бит	4-8	8-16	16-32
Время реакции, мс	более 10	1-10	менее 1
Частота дискретизации	10-100 Гц	0.1- 10 КГц	0.01-10 МГц
Производительность, MIPS	До 1	1-10	10-100 и более
Объем программ (данных)	От 512 байт (32 байт)	32 КБ (8 КБ)	64 КБ (16 КБ) и более
Стоимость МК, уе	1	до 10	более 10

2. Классификация МК

Основным классификационным признаком МК является разрядность процессора. Различают 4, 8, 16 и 32 разрядные МК.

От разрядности процессора зависят стоимость и производительность МК, приблизительно в два раза на каждое удвоение разрядности. На рисунке 1 приведены графики стоимости МК различной разрядности в зависимости от объема встроенной памяти. При увеличении объема встроенной памяти доля затрат на процессор падает. Поэтому при определенных значениях объема встроенной памяти переход на процессор более высокой разрядности становится экономически оправданным.

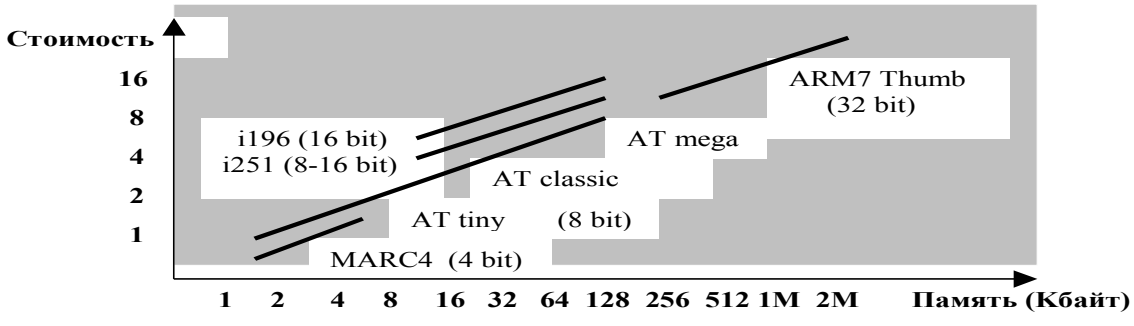

Рисунок 1

4-разрядные МК занимают 10% от объема выпуска микроконтроллеров (четверть в штучном выражении). В основном это массовые изделия с масочным ПЗУ для применения в электронных часах, калькуляторах, а также контроллерах бытовой аппаратуры и автомобильной электроники.

16-разрядные МК занимают на рынке около трети объема выпуска микроконтроллеров. Однако, в последнее время, в связи с распространением устройств цифровой обработки сигналов, они стали уступать в объеме выпуска 32-разрядным МК, которые обеспечивают более высокую производительность.

Основную долю объема выпуска микроконтроллеров занимают 8-разрядные МК за счет наилучшего соотношения цены и технических характеристик. Современные 8-разрядные МК могут работать на частоте до 100 МГц, что позволяет им решать не только задачи логического и цифрового управления, но и задачи цифровой обработки сигналов.

3. Обзор архитектуры процессоров МК

3.1. Intel 8051

МК 8051 был выпущен фирмой Intel в 1980 году. Архитектура МК 8051 в настоящее время воспроизводится и развивается такими фирмами как Phillips, Atmel, STM, Analog Device и др.

МК 8051 имеет CISC архитектуру с одноадресной системой команд переменной длины от 1 до 3 байт обработки данных типа аккумулятор-регистр, аккумулятор-память, константа-память и пересылки память-память. Памяти команд и данных разделены. Память данных содержит до 512 байт внутренней памяти, 128 байтов регистров периферийных устройств и до 64 Кбайт внешнего ОЗУ. РОН образуют 4 банка и размещены во внутренней памяти. Аккумулятор, РОН и данные при косвенной адресации обрабатываются байтовыми командами, прямая адресация памяти и регистров периферии, а также константы задаются в команде дополнительными байтами. Внешнее ОЗУ и память команд доступны в командах загрузки/выгрузки аккумулятора.

Машинный цикл выполнения команды МК 8051 содержит 12 тактов, однако система команд позволяет выполнять команду за 1-2 такта, что реализовано в последних изделиях фирмы Atmel. Дополнительный такт требуется для команд управления и модификации данных.

В таблице 3 приведены основные способы задания операндов в командах.

Таблица 3

Обозначение	Наименование	Примечания
ACC	Аккумулятор	доступен по адресу E0
Rn	Регистровая	4 банка по 8 регистров
Direct	Прямая	128 байт ОЗУ и 128 байт регистров периферии
#	Непосредственная	байт в команде
@Ri	косвенная через R0,R1	256 байт внутреннего и 256 байт внешнего ОЗУ
@DPTR	косвенная через DPTR	64 Кбайт внешнего ОЗУ
@A+DPTR	относительная через DPTR	256 байт памяти команд относительно DPTR
Rel	относительная через PC	256 байт памяти команд относительно PC

В таблице 4 приведены основные форматы команд работы с данными.

Таблица 4

Операнд	A	Rn	Direct	@Ri	DPTR	#

A	un A	bin A,Rn
		mov A,Rn
		xch A,Rn
			bin A,direct
			mov A,direct
			xch A,direct
			cjne A,direct,rel
				bin A,@Ri
				mov A,@Ri
				movx A,@Ri
				xch A,@Ri
					movx A,@DPTR
					movc @A+DPTR
					movc @A+PC
						bin A,#
						mov A,#
						cjne A,#,rel

Rn	mov Rn,A	un Rn djnz Rn,rel	mov Rn,direct	-	-	mov Rn,# cjne Rn,#,rel
Direct	log direct,A	mov direct,Rn	un direct mov direct,direct djnz direct,rel	Mov direct,@Ri	-	log direct,# mov direct,#
@Ri	mov @Ri,A movx @Ri,A	-	mov @Ri,direct	un @Ri	-	mov @Ri,# cjne @Ri,#,rel
DPTR	movx @DPTR,A	-	-	-	inc DPTR	mov DPTR,#

В таблице 5 приведены основные форматы команд управления.

Таблица 5

Мнемоника	Наименование
Jxx rel	ветвление по условию
Jbc bit,rel	ветвление при установленном бите
Cjne A,arg,rel	ветвление при неравенстве
djnz arg,rel	декрементировать и перейти по нулю
Xjmp addr	передача управления по короткому (11 бит) или длинному (16 бит) адресу
jmp @A+DPTR	Передача управления по косвенному адресу относительно DPTR
xcall addr	вызов подпрограммы по короткому (11 бит) или длинному (16 бит) адресу

3.2. МК PIC

МК PIC выпускаются фирмой Microchip с конца 80-х годов. За счет разнообразия моделей и хороших технических характеристик при доступной цене МК PIC получили широкое распространение в отечественных разработках.

МК PIC имеют архитектуру с раздельными областями памяти команд и данных, одноадресной аккумуляторной системой команд, выполняющихся за один машинный цикл, состоящий из 4 тактов. Доступ к данным осуществляется через регистровое окно размером до 256 байт. Переключение регистрового окна осуществляется программно.

МК PIC делятся на три семейства - младшее, среднее и старшее, отличающихся длиной командного слова, объемами адресуемой памяти и составом команд (таблица 7).

Таблица 7

Параметр	PIC12XXX	PIC16XXX	PIC17XXX
разрядность команды, бит	12	14	16
размер регистрового окна, байт	32	128	256
размер страницы ПЗУ, слов	512	2048	16384
ПЗУ, слов	384-2048	512-8192	2048-16384
ОЗУ, байт	25-73	80-368	232-1536
число команд	33	35	55

В таблице 8 приведены основные способы задания операндов команды. Косвенная адресация в МК PIC осуществляется с использованием регистра адреса FSR и псевдорегистра INDF.

Таблица 8

Обозначение	Наименование
W	Аккумулятор
F	регистровая (прямая)
indf	Косвенная
#	непосредственная

В таблице 9 приведены форматы команд обработки данных. Результат операции может сохраняться в аккумуляторе или регистре.

Таблица 9

Разрядность команды	Поля
12	11 КОП 5 | 4 f 0
14	13 КОП 7 | 6 f 0
16	15 КОП 8 | 7 f 0

В таблице 10 приведены форматы команд с константами, в т.ч. ветвления.

Таблица 10

Разрядность команды	Поля
12	11 КОП 8 | 7 k (символ) 0
14	13 КОП 8 | 7 k (символ) 0
16	15 КОП 8 | 7 k (символ) 0

В таблице 11 приведены форматы передач управления и вызова процедур.

Таблица 11

Разрядность команды	Поля
12	11 КОП 9 | 8 k (символ) 0
14	13 КОП 11 | 10 k (символ) 0
16	15 КОП 8 | 12 k (символ) 0

Для управления выполнением программы используются команды пропуска по флагам регистра состояния. Безусловные переходы и вызов процедуры выполняются в пределах страницы ПЗУ. Страницы ПЗУ переключаются через специальный регистр PCLATH. Размер страницы составляет 512, 2048 и 8192 байт для разрядности команд 12, 14 и 16 бит соответственно.

Машинный цикл PIC занимает 4 такта, однако оптимизация аппаратуры позволяет выполнять команду за два такта.

3.3. МК AVR

Фирма Atmel начала выпускать 8-разрядный микроконтроллер AVR со второй половины 90-х годов. AVR имеет RISC-архитектуру с 32 регистрами общего назначения, раздельными шинами памяти команд и данных, 2-адресными командами обработки данных на регистрах, доступом к памяти и регистрам ввода-вывода через команды загрузки-выгрузки. Семейство МК AVR включает модели Tiny, Classic и Mega. Параметры моделей приведены в таблице 12.

Таблица 12

Параметр	Tiny	Classic	Mega
ПЗУ, Кбайт	1 – 2	2-8	8-128
ЭСППЗУ, байт	Нет, 64 (128)	64-256	512 - 4 К
индексные регистры	Z	X, Y, Z	X, Y, Z
ОЗУ, байт	32РОН, нет (128)	до 1 К	до 4 К
Стек	3 уровня	в ОЗУ	в ОЗУ

Команды имеют фиксированную длину 16 бит и выполняются за 1 такт. Исключение составляют некоторые команды доступа к данным и управления, которые имеют длину 32 бита и выполняются за два такта.

Старшие 6 регистров образуют регистровые пары X, Y (Classic и Mega) и Z, которые используются при косвенной адресации.

В таблице 13 приведены способы задания операндов в командах обработки данных. С константами работает только старшая половина регистров.

Таблица 13

Обозначение	Адресация	Модель
R	Прямая РОН	все
P	Прямая регистра ввода-вывода	
@Z	косвенная	
Addr	прямая	
#	непосредственная	
X+n	относительная косвенная	только Classic и Mega
-X	косвенная с предекрементом	
X+	косвенная с постинкрементом	

Состав команд включает в себя команды передачи данных между регистрами и памятью, в т.ч. чтение и запись программной памяти, чтение и запись внешних регистров, команды обработки данных, в т.ч. работы с битами, команды передачи управления и ветвления. Ветвление выполняется относительно счетчика команд со смещением 6 бит, передача управления и вызов процедуры выполняются по абсолютному адресу 12 бит.

В старших моделях AVR реализованы абсолютная (21 бит) и косвенная передача управления и вызова процедуры. В модели Mega также реализовано двухтактное аппаратное умножение и 16-разрядные операции с регистровыми парами.

3.4. МК MARC Atmel

MARC - это 4-разрядный микроконтроллер со стековой архитектурой. MARC разработан фирмой Temic, после приобретения которой, фирма Atmel начала выпускать его под собственной торговой маркой.

МК имеет стек данных и стек возвратов, а также два индексных регистра X, Y для косвенной адресации данных. Команды имеют длину 1 байт, для прямой адресации данных и длинных переходов или вызовов процедур используются двухбайтовые команды. Один байт команды выполняется за один такт.

Обработка данных осуществляется на стеке. Верхушка стека данных реализована в виде отдельного регистра-аккумулятора. Обмен данными из памяти осуществляется командами загрузки-выгрузки стека с использованием непосредственной, прямой, а также косвенной адресации через регистры X, Y, включая инкрементную и декрементную моды. 4-байтовые константы загружаются в стек одной командой.

3.5. Микроконтроллер UNC80

8-разрядный микроконтроллер UNC80 является развитием серийного микроконтроллера 1878ВЕ1 з-да «Ангстрем». МК имеет гарвардскую архитектуру с раздельными шинами доступа к памяти программ и данных.

Память программ организована в виде последовательности 16-разрядных слов, разделенных на 256 секций по 32 Кслова в каждой. Номер секции задается в служебном регистре.

Память данных организована в виде байтового двухпортового ОЗУ, разделенного на 256 секций по 64 Кбайта в каждой. Номер секции задается в служебном регистре.

Основным способом доступа к памяти данных является прямая адресация с использованием четырех регистровых окон A, B, C, D (служебные регистры). В каждом окне доступно 8 байт. Номер окна и адресуемого байта задаются непосредственно в команде. Места расположения окон (старшие 8 разрядов) задаются в 3 служебных регистрах. Это позволяет адресовать 2 Кбайт адресного пространства памяти данных. Служебные регистры d0-d7 могут использоваться в качестве операндов команды для косвенной адресации памяти, стека и задания числа циклов. Обращение к регистрам d6, d7 обеспечивает косвенную, авто-инкрементную и авто-декрементную адресацию памяти данных до 64 Кбайт. Регистр-указатель SP (d5) используется для организации стека в памяти данных с использованием операций загрузки и выгрузки стека, косвенной и индексной адресации относительно верхушки стека.

Чтение команд из памяти команд осуществляется через счетчик команд.

Содержимое памяти команд может быть прочитано побайтно, используя специальный режим адресации через регистр d6.

Регистры внешних устройств находятся в отдельном адресном пространстве размером 2 Кб, доступ к которому осуществляется через регистровое окно C.

Микроконтроллер имеет 19 служебных регистров и 2 внутренних регистра. Доступ к служебным регистрам осуществляется с помощью специальных команд загрузки/выгрузки.

В микроконтроллере имеется внутренний стек для сохранения контекста при прерывании и вызове процедур. Счетчик команд и регистр состояния сохраняются в стеке аппаратно - регистры процессора с помощью специальных команд.

Микроконтроллер имеет 5 групп команд:

- пересылки данных;
- загрузки и выгрузки регистров процессора;
- арифметической и логической обработки (в т.ч. сдвиговые операции);
- управление выполнением программы;
- управление режимом работы процессора.

Система команд микроконтроллера 2-адресная. В команде задаются операнды источник и приемник операции. В командах можно непосредственно задавать байтовую или 4-разрядную константу. Имеются команды пересылки цепочки байт (до 256) и одного 16-разрядного слова. Команды имеют фиксированную длину 16 бит, исключение составляют команды вызова процедуры и передачи управления, которые занимают 2 слова.

Микроконтроллер имеет 2 ступенчатый конвейер. Машинный цикл выполнения команды занимает два такта. Данные из памяти выбираются одновременно по двум шинам для операнда источника и приемника.

3.6. Микроконтроллер ARM7 Thumb

ARM7 Thumb построен на базе 32-разрядного процессора ARM7. Процессор ARM7 имеет регистровую архитектуру с 16 регистрами и 32-разрядную трехадресную систему команд. Особенностью работы процессора ARM7 в режиме Thumb является использование 16-разрядных команд, которые являются сокращенным вариантом набора 32-разрядных команд процессора ARM7, а также наличие только 8 регистров общего назначения.

Система команд процессора ARM7 в режиме Thumb содержит 19 форматов, включающих команды сдвиговых, арифметических и логических операций на регистрах, команды загрузки и выгрузки регистров в память и стек, команды передачи управления, программного прерывания и специальные команды.

Сдвиговые команды обеспечивают параметрический сдвиг содержимого регистра с сохранением результата операции в другом регистре.

Арифметические операции имеют 3-адресный и 2-адресный форматы. Операндом операции может быть короткая константа длиной 3 или 8 разрядов.

Логические операции и операции сравнения имеют 2-адресный формат. Операции с константами требуют предварительной загрузки константы в регистр.

Команды загрузки-выгрузки регистров обеспечивают пересылку данных между регистрами, загрузку константы в регистр и доступ к данным в памяти с использованием индексной адресации относительно регистра, 5-разрядного смещения относительно регистра или 8-разрядного смещения относительно указателя стека (локальные данные) или счетчика команд (константы).

Команды ветвления содержат код условия и 8-разрядное смещение адреса перехода.

Безусловные передачи управления задают 11 или 22-разрядное смещение, либо регистр, который содержит адрес передачи управления. Особенностью системы команд процессора ARM является отсутствие команды вызова программ, которую заменяет команда передачи управления с 22-разрядным смещением, сохраняющая адрес возврата в специальном регистре LR (Link Register). Сохранение и восстановление регистра LR осуществляется программно.

ARM7 Thumb имеет 3-ступенчатый конвейер. При выполнении перехода выбранная команда аннулируется. Регистровые команды выполняются за один такт. Однако из-за того, что процессор имеет Фон-неймановскую архитектуру, команды доступа к памяти, включая константы из памяти команд, требуют дополнительных тактов.

4. Методика оценки эффективности МК

Традиционный подход к оценке эффективности процессора МК заключается в использовании специальных bench-mark программ с последующим измерением объема кода и времени выполнения программы. Этот подход не позволяет оценивать эффективность процессора МК на этапе проектирования его архитектуры. Кроме этого, на результаты оценки bench-mark программ оказывает определенное влияние эффективность реализации самого компилятора, что снижает объективность оценки собственно архитектуры процессора.

В качестве альтернативы предлагается использовать для оценки эффективности процессора МК затраты машинного кода и числа тактов, необходимых для реализации операторов языка высокого уровня на основе частоты их употребления (веса) в типовых программах логического и цифрового управления, где МК применяются наиболее широко.

Ниже приведены статистические характеристики программ обработки данных для задач управления (где доля вычислений достаточно мала), измеренные по различным параметрам (Электроника СБИС. Проектирование микроструктур. под ред.Н.Айнспрук, М.Мир 1989).

Таблица 14. Доля операторов

Присваивание	Условные операторы	Циклы	Вызов процедур
49%	16%	17%	18%

Таблица 15. Доля операндов и операций в операторах присваивания

Один операнд	Один операнд и знак операции	Более
66%	20%	14%

Таблица 16. Доля операций в операторах присваивания

Логические	сложение и вычитание	Умножение
55%	42%	3%

Таблица 17 Распределение типов данных

Константы	Скаляры	Массивы и пр.
33%	42%	25%

Таблица 18 Распределение констант различной разрядности

Константа, бит	1	4	8	16 и более
Доля, %	20	30	40	10

Для оценки эффективности архитектуры процессора МК операторы языка высокого уровня сводятся к типовым операциям машинного языка - пересылкам, унарным и бинарным операциям обработки и сравнения данных, условным и безусловным переходам и вызовам процедур.

В таблице 19 приведены доли типовых машинных операций, необходимых для реализации операторов языка высокого уровня.

Оператор присваивания реализуетя операциями пересылки, унарными и бинарными операциями в соответствии с данными таблицы 14.

Условный оператор реализуется операциями тестирования, бинарными операциями, операциями ветвления и перехода.

Реализация оператора цикла включает в себя операции установки начального значение параметра цикла, его модификации, проверки, ветвления и перехода.

Для операторов вызова процедур принято среднее количество операций пересылки передачи параметров и результата процедуры равное двум. Вызов процедуры и выход из нее требует выполнения еще двух операций, для оценки принято соотношение 5:1.

Косвенная адресация требует операций установки адреса, а также его модификации.

В правой колонке таблицы указаны доли типовых машинных операций, необходимых для реализации операторов языка высокого уровня, в последней строке – их общая доля в реализации конкретного оператора.

Таблица 19 Доли машинных операций

Оператор / Операция	Код	Присваивание (0.49)	Условный (0.16)	Цикл (0.17)	Вызов (0.18)	Адресация (0.10)	Всего доля
Пересылка	mov d,s	0.32	-	0.13	0.36	0.10	**0.91**
Унарная	un d,s	0.05	-	-	-		**0.05**
	un d	0.05	-	0.10	-	0.04	**0.19**
Бинарная	bin d,s,s	0.02	-	-	-		**0.02**
	bin d,s	0.05	0.06	0.02	-	0.01	**0.14**
Проверка	tst s,s	-	0.10	0.05	-	-	**0.15**
Ветвление	Bxx	-	0.16	0.17	-	-	**0.33**
Переход	Jmp	-	0.05	0.04	-	-	**0.09**
Вызов	Call+Ret	-	-	-	0.18+0.04	-	**0.22**
	Итого	**0.49**	**0.37**	**0.51**	**0.58**	**0.15**	**2.10**

В приложении приведены оценки в байтах и тактах реализации типовых машинных операций для МК 8051, PIC, AVR, MARC, UNC и ARM с учетом особенностей их архитектуры и системы команд.

Оценки получены путем учета вклада команд конкретного МК в реализацию типовой операции в соответствии с ее долей.

Результаты оценки приведены в таблице 20.

Таблица 20 Оценки объема программного кода и тактов

МК	**8051**	**PIC**	**AVR**	**MARC**	**UNC**	**ARM**
Пересылка	2.42	3.64	3.06	2.96	1.82	3.31
Унарная	0.40	0.48	1.12	1.20	0.58	1.20
Бинарная	0.62	0.84	0.72	1.12	0.36	0.96
Проверка	0.14	0.40	0.57	0.75	0.34	0.95
Ветвление	0.76	0.66	0.66	0.41	0.66	0.66
Переход	0.18	0.18	0.18	0.20	0.22	0.18
Вызов процедуры	0.43	0.44	0.34	0.32	0.80	0.88
Прочие	0.20	0.20	0.36	-	0.42	0.36
Итого байт	5.15	7.50	7.01	6.96	5.20	8.85
Итого тактов	6.86	8.45	4.16	7.44	6.15	7.09
Относительный объем кода	**1.00**	**1.46**	**1.36**	**1.35**	**1.01**	**1.72**
Относительное число тактов	**1.00**	**1.23**	**0.61**	**1.08**	**0.90**	**1.03**

В определении реализации типовых машинных операций и их долей для конкретного МК могут содержаться ошибки. Однако их влияние на оценку эффективности не столь значительно:

во первых, такая ошибка оказывает влияние на оценку в размере доли типовой операции, а значит ее влияние сокращается в несколько раз;

во вторых, ошибки имеют случайный характер с нормальным законом распределения, поэтому ошибка оценки, за счет сложения, также сокращается в несколько раз;

и наконец, последнее, все МК оцениваются с равными допущениями, поэтому ошибки для каждого МК имеют одинаковое смещение и на погрешность сравнительной оценки практически не влияют.

5. Факторы эффективности МК

Эффективность процессора микроконтроллера определяется его архитектурными особенностями.

Основными факторами, влияющими на эффективность архитектуры процессора МК, являются:

- адресность команд;
- длина команды;
- организация обработки данных;
- состав команд;
- кодировка команд;
- организация выполнения команд;
- длина обрабатываемых данных.

5.1. Адресность команды

Адресность команды определяет количество операндов команды. Различают безадресные команды (стековые), одноадресные (аккумуляторные) и двухадресные. Одноадресная система команд может содержать двухадресные команды пересылки. Двухадресная система команд имеет длину 16 бит, остальные, как правило, имеют длину 8 бит

В таблице 21 приведены расширения типовых операций обработки данных с различным числом операндов.

Таблица 21 Оценка адресности системы команд

Операция	Вес	2-адресная			1-2 адресная			1-адресная			0-адресная		
		Код	Байт	Вес	Код	Байт	Вес	Код	Байт	Вес	Код	Байт	Вес
mov d,s/#	0.91	mov d,s/#	2	1.82	mov d,s/#	3	2.73	ld s/# st d	4	3.64	ld s/# st d	4	3.64
un d,s	0.05	mov d,s un d	4	0.20	mov d,s Un d	5	0.25	ld s un d	4	0.20	ld s un st d	5	0.25
un d	0.19	un d	2	0.38	Un d	2	0.38	un d	2	0.38	ld s un st d	5	0.95
Bin d,s,s/#	0.02	mov d,,s bin d,s/#	4	0.08	ld s bin s/# st d	6	0.12	ld s bin s/# st d	6	0.12	ld s ld s Bin st d	7	0.14
Bin d,s/#	0.14	bin d,s/#	2	0.28	bin d/# st d	4	0.56	ld s bin d	4	0.56	ld s ld s Bin st d	7	0.98
Tst s,s/#	0.15	tst s,s/#	2	0.30	ld s tst s/#	4	0.60	ld s tst s,#	4	0.60	ld s ld s Tst	5	0.75
прочие	0.64		2	1.28		2	1.28		2	1.28		2	1.28
Итого байт				4.34			5.92			6.78			7.99
Оценка				**0.73**			**1.00**			**1.15**			**1.35**

5.2. Длина команды

МК имеют систему команд с переменной и фиксированной длиной. В современных микроконтроллерах двухадресная система команд имеет фиксированную длину 16 бит, одно и нуль-адресная – переменную или фиксированную длину.

Команды переменной длины могут занимать 1, 2 и более байта, содержащие байт командного слова и дополнительные байты констант, адресов данных или меток.

Байтовая длина команд обеспечивает наиболее компактный код, однако увеличивают количество тактов, затрачиваемых на выборку команды.

Команды фиксированной длины 16 бит содержат в командном слове все необходимые данные для ее выполнения. При этом увеличивается объем программного кода, но число тактов на выборку команды сокращается до одного.

Промежуточное положение занимают команды фиксированной длины с переменной разрядностью. В этом случае младшие модели МК имеют команды минимальной разрядности, а по мере увеличения ресурсов в старших моделях увеличивается разрядность командного слова, как правило, от 12 до 16 разрядов. При этом увеличивается разрядность адресных полей команды, а также вводятся новые команды и даже форматы команд.

В таблице 22 приведена оценка влияния разрядности команды на объем программного кода и производительность микроконтроллера в зависимости от доли команд переменной длины 1, 2 и 3 байта. Для оценки принято, что байтовые и 2-байтовые команды переменной длины эквивалентны, соответственно, одной команде фиксированной длины 16 бит, 3-байтовые – двум командам. Последняя колонка таблицы содержит оценки для системы команд длиной 32 разряда.

Таблица 22 Оценка длины команд

Байт	Доля	Переменная байтовая	Фиксированная 16-разрядная	Фиксированная 32-разрядная
1	0.1	0.1	0.2	0.4
2	0.5	1.0	1.0	2.0
3	0.4	1.2	1.6	1.6
Байт / Такт		2.3 / 2.3	2.8 / 1.4	4.0 / 1.0
Оценка памяти / производительности		**1.0 / 1.0**	**1.22 / 0.61**	**1.75 / 0.44**

5.3. Организация обработки данных

Обработка данных является основной задачей МК.

Модель памяти МК для программиста, в общем случае, содержит память системы, входные, выходные и промежуточные данные.

Адресное пространство программы делится на 4 сегмента:

1) программный код;

2) константы;

3) данные;

4) стек.

Программный код и константы размещаются в ПЗУ, данные и стек - в ОЗУ.

Программы делятся на модули. Каждый модуль имеет свой программный код, константы и данные. Взаимодействие модулей осуществляется за счет передачи управления и данных друг другу.

Для размещения данных модулей используются способы статического и динамического распределение памяти. Статическое распределение памяти данных осуществляется во время трансляции программы. При динамическом распределении память для данных выделяется во время работы программы при активизации модуля и освобождается после его завершения.

Для распределения данных применяется стековый механизм, обеспечивающий размещение данных модулей в памяти с учетом правил их видимости. Локальные данные модуля образуют фрейм. Для статических данных фрейм данных модуля имеет фиксированное размещение в памяти, для динамических - плавающее. Доступ к данным фрейма осуществляется по смещению относительно указателя фрейма.

Данные, с которыми работает программа, разделяются на скалярные и структурированные. Скалярные данные адресуются прямым указанием адреса или смещения данного относительно указателя фрейма, структурированные - по ссылке. Ссылка является скалярным данным, содержащим адрес структурного элемента.

Таким образом, модель памяти МК для программиста включает в себя: ПЗУ, ОЗУ, регистры ВУ, указатель фрейма, а также способы адресации памяти - абсолютную, относительную, косвенную по ссылке абсолютной или относительной.

Реализация такой модели для 8-разрядных МК связана со следующими проблемами:

- ограниченная разрядность команд не позволяет указывать в командах обработки данных полный адрес данного;

- малая разрядность данных не позволяет использовать их в качестве ссылки.

В современных микроконтроллерах используется три подхода к организации обработки данных:

а) с использованием регистров;

б) непосредственно в памяти;

в) с использованием стека.

Использование регистров позволяет сократить поле операндов в команде и время доступа к данным.

В тоже время регистры:

- требуют вспомогательных команд загрузки-выгрузки регистров при обработке данных в памяти;

- образуют отдельное адресное пространство, представляющее собой самостоятельный аппаратный ресурс, который требует вспомогательных команд для сохранения и восстановления содержимого регистров при вызове процедур и обработке прерываний.

При обработке данных в памяти используется страничная организация памяти (регистровое окно). Номер страницы хранится в специальном регистре, адрес ячейки (регистра) внутри страницы задается в команде. Адрес ячейки памяти вычисляется путем конкатенации номера страницы и адреса ячейки (регистра) внутри страницы.

Страничная организация памяти снимает проблему ограниченной разрядности поля адресации операнда в команде, однако требует использования вспомогательной команды для установки номера страницы при адресации конкретной ячейки.

Одним из способов решения этой проблемы является использование переменной длины страницы и соответственно переменной разрядности номера ячейки внутри страницы - 4, 8 и более бит. В случае необходимости доступа к данным, лежащим за пределами 4-битной страницы, используется адресация 8-битной страницы, и т.д.

Еще одним подходом к организации обработки данных в МК является использование стека. Операции на стеке выполняются командами без указания операндов. Стек, как и регистры, обеспечивает быстрый доступ к данным, однако требует дополнительных затрат на загрузку-выгрузку данных в стек и обратно. Использование стека позволяет сократить число вспомогательных команд пересылки параметров при вызове процедур, а также количество данных в памяти за счет их хранения на стеке.

В таблице 23 приведена оценка числа операций (КОП) и операндов (ОП) для МК с двухадресной системой команд с обработкой данных в памяти и регистрах. В последних колонках приведены расширения команд для регистровых ЭВМ при обработке данных только в памяти, которые являются фактически расширениями команд стековой ЭВМ.

Доля регистровой адресации установлена на основании рисунка 2.

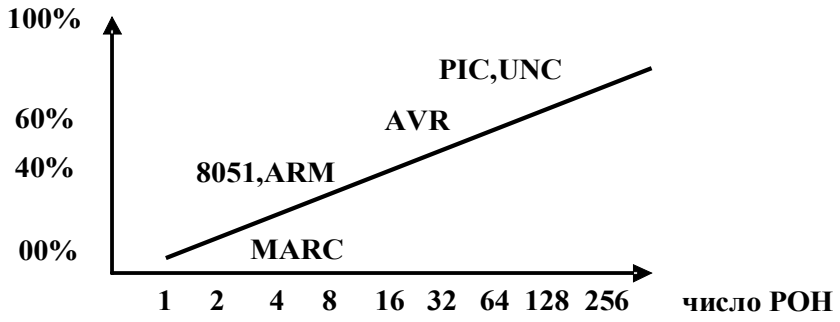

Рисунок 2

Таблица 23 Оценка обработки данных в памяти, на регистрах и стеке

Операция	Код	Доля	КОП/ОП		Вес		Код	КОП/ОП		Вес	
Пересылка	mov d,s/#	0.91	1	1-2 (1.5)	0.91	1.37	ld r,s/# st r,d	2	3-4 (3.5)	1.82	3.19
Унарная	mov d,s un d	0.05	2	3	0.10	0.55	ld r,s un r st r,d	3	5	0.15	0.25
	un d	0.19	1	1	0.19	0.19	ld r,d un r st r,d	3	5	0.57	0.95
Бинарная	mov d,s bin d,s/#	0.02	2	3-4 (3.5)	0.04	0.07	ld r,d (ld r,d) bin r,r/# st r,d	3-4 (3.5)	5-8 (6.5)	0.07	0.13
	bin d,s/#	0.14	1	1-2 (1.5)	0.14	0.21	ld r,d (ld r,d) bin r,r/# st r,d	3-4 (3.5)	5-8 (6.5)	0.49	0.91
Проверка	tst s,s/#	0.15	1	1-2 (1.5)	0.15	0.23	ld r,d (ld r,d) tst r,r/#	2-3 (2.5)	3-6 (4.5)	0.43	0.68
Прочие	-	0.64	1	1	0.64	0.64	-	1	1	0.64	0.64
Всего		2.10			2.17	3.47				4.17	6.75
Оценка	КОП	ОП			**1.00**	**1.46**				**1.92**	**2.84**
Память	**1.00**	**1.46**			**1.00**	**1.46**					
32 регистра	**1.37**	**1.81**			0.60					0.40	
					0.60	**1.10**				**0.77**	**0.71**
8 регистров	**1.55**	**2.01**			0.40					0.60	
					0.40	**0.88**				**1.15**	**1.14**
Стек	**1.92**	**1.46**			**-**					**1.00**	

5.4 Состав команд

Состав команд процессора МК включает в себя команды доступа к данным, регистрам ВУ, ПЗУ, арифметические, логические и сдвиговые команды обработки данных, команды управления ходом выполнения программы, включая условную и безусловную передачу управления, вызов подпрограммы и возврата из нее, специальные команды работы с внутренними регистрами процессора.

В командах обработки данных важно иметь формат непосредственной работы с константами, доля которых в операндах команды весьма существенна.

Логические операции составляют более половины команд обработки данных. Поэтому использование специальных команд работы с битами обеспечивает сокращение программного кода.

Для организации управления используются составные команды, которые обеспечивают проверку (и модификацию) операнда и выполнение ветвления по заданному условию, это также сокращает длину программного кода за счет исключения команд тестирования.

В таблице 24 приведены доли типовых операций анализируемых МК. Серым фоном выделены команды, которые для конкретного МК имеют значение отличное от типового.

Таблица 24 Оценка состава команд

Операция	Код	Доля	8051	PIC	AVR	MARC	UNC	ARM
Пересылка	mov d,s	0.91	0.91	0.91	0.91	0.73	0.91	0.91
Унарная	un d,s	0.05	0.05	0.05	0.05	0.05	0.05	0.05
	un d	0.19	0.09	0.19	0.15	0.19	0.19	0.19
Бинарная	bin d,s,s	0.02	0.02	0.02	0.02	0.02	0.02	0.02
	bin d,s	0.14	0.14	0.14	0.14	0.14	0.14	0.14
Проверка	tst s,s	0.15	0.05	0.10	0.15	0.15	0.15	0.15
Ветвление	Bxx	0.33	0.64	0.64	0.64	0.64	0.64	0.42
Переход	Jmp	0.09						0.44
Вызов	Call+Ret	0.22						

Вспомогательные	.	.	0.10	0.05	0.18	-	0.21	0.18
Всего байт		2.10	1.98	2.10	2.24	1.92	2.31	2.50
Оценка			**1.00**	**1.06**	**1.13**	**0.97**	**1.17**	**1.26**

5.5 Кодировка команд

Самая простая оценка эффективности кодирования команд заключается в оценке расхождения совокупного веса кода команды и операндов с частотой их появления. При эффективной кодировке это расхождение должно быть минимальным.

Эффективность кодирования обеспечивается за счет расширяемых кодов и кодов переменной длины. Для расширяемых кодов длина кода операции зависит от частоты ее использования. Для кодов переменной длины от частоты использования операции зависит также длина команды, поэтому переменная длина команд потенциально обеспечивает более эффективное кодирование.

При выборе кодировки команд разработчик сталкивается с проблемой ограниченной разрядности командного слова, поэтому должен минимизировать состав и количество форматов команд, способов адресации операндов и длину полей команды.

Уменьшить ограничения разрядности командного слова позволяют контекстное кодирование.

При контекстном кодировании одно и тоже командное слово интерпретируется в зависимости от установленного контекста. Контекст устанавливается специальными командами. Разновидностью команд установки контекста являются непрерываемые префиксные команды, которые уточняют действие следующей команды. Преимущество использования префиксных команд заключается в том, что их выполнение осуществляется сразу после выборки, тогда как в команде переменной длины ее выполнение осуществляется только после выборки последнего слова команды.

5.6 Организация выполнения команд

Одним из широко используемых способов повышения производительности процессора МК является конвейерное выполнение команд.

В общем случае выполнение команды включает в себя следующие операции:

- выборка команды;
- дешифрация команды;
- чтение операндов;
- обработка;
- запись результата.

Время выполнения команды слагается из времен выполнения операций. В конвейере выполнение операции делится на ступени, которые выполняются отдельными устройствами. За счет совместной работы устройств производительность может быть увеличена пропорционально количеству ступеней конвейера.

Однако увеличение числа ступеней конвейера более двух ведет к появлению конфликтов в его работе. Для снижения потерь от конфликтов используются различные способы, которые требуют дополнительных аппаратных затрат и ограничений на последовательность выполнения команд. Поэтому в современных микроконтроллерах используют двухступенчатый конвейер с аннулированием выбранной команды при передаче управления.

Время выполнения команды называют машинным циклом. В зависимости от архитектуры процессора машинный цикл может содержать от 1 до 4 тактов работы процессора. Так, например, выполнение команды обработки данных в памяти требует чтения двух операндов из памяти, их обработки и записи результата в память, что требует не менее трех тактов только на обращение к памяти данных. Для выполнения команд за один такт необходимо разделять доступ к данным на отдельные операции чтения и записи, что реализовано в регистровой и стековой архитектурах МК. В МК с обработкой данных в памяти один машинный цикл выполняется за 2 такта.

Для хранения программ в МК используется Flash память, цикл обращения к которой занимает до 20 нс. За это время процессор, работая на более высокой частоте, успевает выполнить несколько операций конвейера. В случае хранения программ в памяти, для которой времена доступа к командам и данным равноценны, для достижения максимальной производительности необходимо выполнять машинный цикл за один такт работы процессора.

На работу конвейера большое влияние оказывает архитектура памяти процессора – раздельная память для программ и данных (Гарвадская) или общая (Фон-Неймановская). В первом случае выборка команд и доступ к памяти данных могут быть совмещены. Во втором – доступ к данным требует дополнительных тактов.

В таблице 25 приведены данные по числу дополнительных циклов для Гарвадской и Фон-Неймановской архитектур.

Первая колонка содержит число циклов, затрачиваемых на выборку команд. Второй колонкой приведены дополнительные циклы при организации конвейера с отложенным переходом, при котором выбранная команда после перехода всегда выполняется (в 30% случаев это NOP). Для команд ветвления принято, что в 50% случаев устанавливается новое значение счетчика команд с потерей цикла на выбранную команду. Для Гарвадской архитектуры вторая колонка содержит число дополнительных циклов для записи результата операции в память.

Таблица 25 Оценка числа машинных циклов

Операция	Код	Доля	Выборка		Отложенный переход		Раздельные шины				Общая шина	
			Цикл	Вес	Цикл	Вес	Цикл	Вес	Цикл	Вес	Цикл	Вес
Пересылка	mov d,s/#	0.91	1	0.91	0	0	0	0	1	0.91	1-2	1.37
Унарная	mov d,s un d	0.05	2	0.10	0	0	0	0	2	0.10	3	0.15
	Un d	0.19	1	0.19	0	0	0	0	1	0.19	2	0.38
Бинарная	mov d,s bin d,s/#	0.02	2	0.04	0	0	0	0	2	0.04	2-3	0.05
	bin d,s/#	0.14	1	0.14	0	0	0	0	1	0.14	1-2	0.21
Проверка	tst s,s/#	0.15	1	0.15	0	0	0	0	0	0.00	1-2	0.23
Ветвление	Bxx	0.33	1	0.33	0.5	0.17	0.5	0.17	0.5	0.17	0.5	0.17
Переход	Jmp	0.09	1	0.09	0.3	0.03	1	0.09	1	0.09	1	0.09
Вызов	Call	0.22	1	0.22	0.3	0.07	1	0.22	1	0.22	1	0.22
Итого циклов		**2.10**		**2.17**		**0.27**		**0.48**		**1.86**		**2.82**
Всего циклов						**2.44**		**2.65**		**4.03**		**4.99**
Оценка				**1.00**		**1.12**		**1.22**		**1.88**		**2.30**

5.7 Длина данных

Для 8-разрядных МК при увеличении разрядности данных увеличивается число команд, затрачиваемых на их обработку.

В таблице 26 приведены оценки числа команд 8-разрядного микроконтроллера при обработке данных длиной 8, 16 и 32 разряда. Поскольку повышенная разрядность данных характерна для вычислительных задач, доля бинарных операций обработки данных увеличена вдвое. Операция 16-разрядного умножения, как минимум, требует 4 команд 8-разрядного умножения и 8 команд 8-разрядного сложения. С учетом затрат на обработку знака и пересылки принято, что на 16-разрядное умножение затрачивается 16 команд обработки 8-разрядных данных, а для 32-разрядных данных количество команд увеличено до 48. При оценке доли 8, 16 и 32-разрядных данных приняты соответственно в размере 0.4, 0.5 и 0.1.

Таблица 26 Оценка длины данных

Операция	Код	Доля	8 бит		16 бит		32 бит	
Пересылка	mov d,s/#	0.84	1	0.84	2	1.68	4	3.36
Унарная	mov d,s un d	0.05	2	0.10	4	0.20	8	0.40
	un d	0.19	1	0.19	2	0.38	4	0.76
Бинарная	mov d,s bin d,s/#	0.04	2	0.08	4	0.16	8	0.32
	bin d,s/#	0.24	1	0.24	2	0.48	4	0.96
	mul d,s/#	0.05	1	0.05	16	0.80	48	2.40
Проверка	tst s,s/#	0.15	1	0.15	3	0.45	7	1.05
Прочие	…	0.64	1	0.64	1	0.64	1	0.64
Су мма		2.10		2.19		4.79		9.89
Оценка				**1.00**		**2.19**		**4.52**

Доля	0.4	0.5	0.1
Итого	0.40	1.10	0.45
Оценка			**2.05**

5.8. Результаты оценки эффективности МК

Оценка эффективности архитектур процессоров МК определяется отношением:

$E = Sum (a_{i=j} * K_j) / Sum (a_{i=j} * KK_j)$ (1)

где E – оценка эффективности;

a_i - доля операции;

K_j,KK_j – вклад операции j для архитектуры К и КК.

Оценка архитектуры по коэффициентам вклада архитектурных факторов, в случае их ортогональности, определяется произведением однофакторных отношений:

$E = \prod E_{kK}$ (2)

где E_k - оценка фактора j для архитектуры К относительно архитектуры КК.

В таблицах 27, 28 приведены оценки объема машинного кода и числа тактов, анализируемых МК по коэффициентам вклада архитектурных факторов.

Таблица 27 Сравнительная оценка объема машинного кода

Фактор	8051	PIC	AVR	MARC	UNC	ARM
Длина команды	1.00	1.22	1.22	1.00	1.22	1.22
Организация обработки	1.00	1.00	1.37	1.00	1.00	1.55
Формат команды	1.00	1.15	0.73	1.35	0.73	0.73
Состав команд	1.00	1.06	1.13	0.97	1.17	1.26
Итого	**1.00**	**1.49**	**1.38**	**1.31**	**1.04**	**1.74**
Итого (Таблица 20)	**1.00**	**1.46**	**1.36**	**1.35**	**1.01**	**1.72**

Таблица 28 Сравнительная оценка числа тактов

Фактор	8051	PIC	AVR	MARC	UNC	ARM
Объем кода (таблица 20)	5.15	7.50	7.01	6.96	5.20	8.85
Число тактов	5.15	7.50	3.50	6.96	5.20	4.92
Дополнительные такты	1.82	0.96	0.48	0.48	0.96	2.82
Всего тактов	6.97	8.46	3.98	7.46	6.16	6.74
Оценка тактов	1.00	1.21	0.57	1.07	0.88	0.96
Таблица 20	**1.00**	**1.23**	**0.61**	**1.08**	**0.90**	**1.03**

Достаточно близкое совпадение оценок, полученных по расширениям типовых машинных команд для анализируемых МК (таблица 20), и рассчитанных по коэффициентам вклада архитектурных факторов (таблицы 27, 28):

а) подтверждает достоверность оценок;

б) позволяет использовать однофакторную оценку эффективности архитектуры МК для выбора проектных решений при его разработке.

В качестве примера в таблице 29 приведены оценки эффективности МК при решении вычислительных задач. Для 32-разрядного процессора ARM объем кода и число тактов сокращены в 2.05 раза относительно 8-разрядных процессоров в соответствии с таблицей 26.

Таблица 29 Оценка эффективности МК при обработке 32-разрядных данных

МК	8051	PIC	AVR	UNC	ARM
Объем кода (таблица 20)	**1.00**	**1.33**	**1.36**	**1.01**	**1.72**
Число тактов (таблица 28)	**1.00**	**1.14**	**0.61**	**0.90**	**1.03**
Затраты на обработку данных	1.00	1.00	1.00	1.00	0.49
Оценка объема кода	**1.00**	**1.37**	**1.71**	**0.93**	**0.84**
Оценка числа тактов	**1.00**	**1.17**	**0.74**	**0.83**	**0.50**

Рисунок 3 демонстрирует влияние архитектурных ообенностей МК факторов на объем машинного кода.

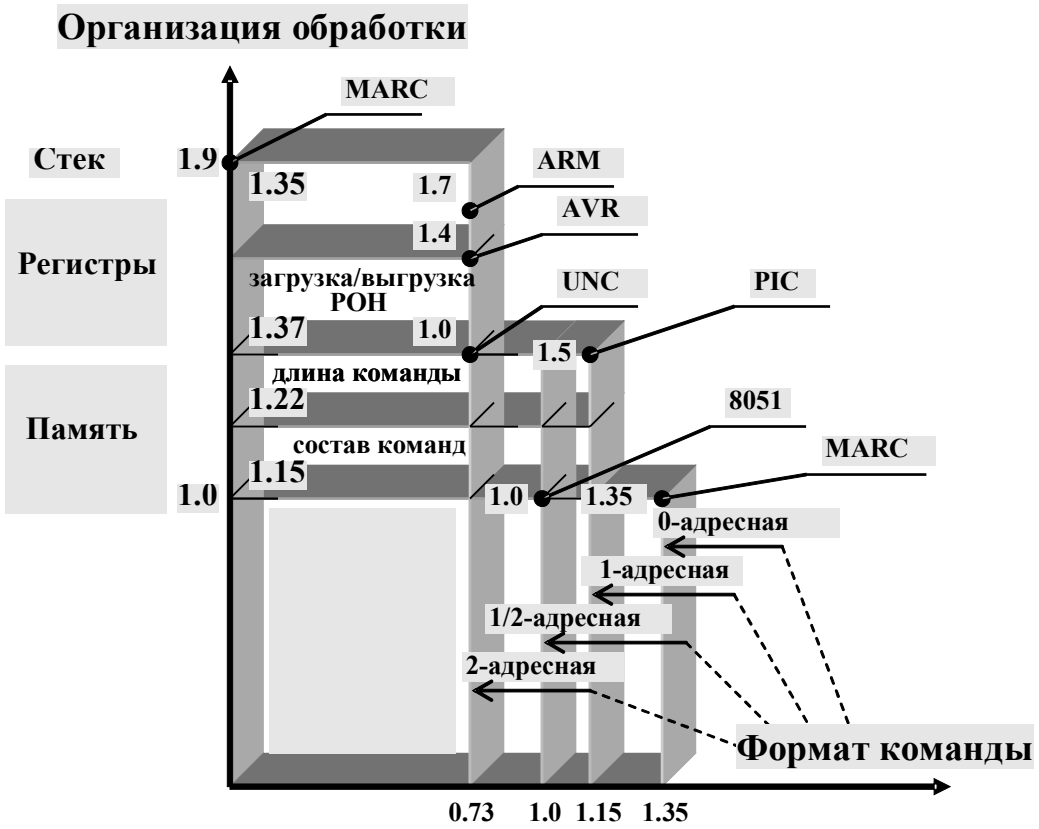

Рисунок 3

На рисунке объем программного кода представлен площадью прямоугольника, одна сторона которого является фактором организации обработки (абсцисса), другая - формата команды (ордината). На рисунке также отражено влияние состава и длины команды, вносящих свой вклад в увеличение объема программного кода.

6. Процессор M8

Процессор M8 является концептом 8-разрядного процессорного ядра микроконтроллера.

Процессор M8 отличает:

а) возможность обработки данных длиной 8, 16 и 32 разряда, данные длиной более байта обрабатываются потактно;

б) байтовый формат команд с безадресными стековыми командами обработки данных, выполняющихся за один такт. Имеются префиксные команды, которые дополняют информационное поле команды до 8 и более разрядов, а также расширяют безадресные стековые команды до одноадресных команд обработки данных в памяти и двухадресных команд пересылки данных;

в) страничная адресация памяти с возможность косвенной, индексной, инкрементной и декрементной адресации до 64 Кб ОЗУ через два индексных регистра, а также прямой адресации регистров ВУ;

г) высокая эффективность вызова программ. Процессор имеет два стека - один для обработки данных и передачи параметров вызываемой программе, другой для хранения адресов возврата;

д) ориентация на логическую обработку данных. Система команд содержит команды сравнения, тестирования и ветвления, битовой обработки данных в памяти и организации циклов.

Память процессора M8 имеет страничную организацию с переменной длиной страницы. В командах обработки данных доступны две смежные страницы памяти по 8 байт каждая. Байтовый адрес страницы хранится в регистре состояния процессора. Доступ к странице может расширяться префиксом константы до 128 и 2054 байт.

Форматы базовых команд процессора M8 приведены в таблице 30.

Таблица 30

Мнемоника	**Наименование**	**Байт**
K	Префикс константы	1
M	Префикс операнда	1
Opw	Операция	1
oplw Kw	Операция с константой	2
ldw M	Загрузка памяти	1
ldlw K	Загрузка константы	2
stw M	Выгрузка в память	1
Bxxw A	Тестирование и условный переход	2
bxxlw K A	Тестирование с константой и условный переход	3
Br A	Безусловный переход	2
Jsr A	Вызов процедуры	2

Базовые команды могут предваряться префиксом – операнда и константы.

Префикс константы содержит в информационном поле 4-разрядный код константы, которая расширяет адресное поле операнда, ветвления и адреса вызова программы.

Префикс операнда содержит 4-разрядное поле операнда в памяти или номер индексного регистра. Префикс позволяет задавать команды модификации данных в памяти и битовые команды. В таблице 31 приведены форматы команд с префиксом операнда.

Таблица 31

Мнемоника	**Наименование**	**Байт**
M opw	Операция модификации данных в памяти	2
M oplw Kw	Операция модификации данных в памяти с константой	3
M log N	Битовые команды в памяти	2
M stlw K	Загрузка константы в память	3
M stw M	Пересылка	2
M Blog N A	Проверка бита в памяти и условный переход	3
M Bxx A	Тестирование памяти и условный переход	3
M bxxlw K A	Тестирование памяти с константой и условный переход	4

Команды обработки данных имеют 2 формата - операции на стеке и с константой:

а) операции на стеке выполняются над верхними элементами стека с сохранением результата операции в стеке. Префикс операнда задает операцию модификации данных в памяти;

б) операции с константой выполняются над верхушкой стека и константой, которая расположена в следующих байтах после команды, результат операции записывается в стек. Префикс операнда задает операцию модификации данных в памяти.

Команды загрузки-выгрузки стека содержат поле адресации памяти. Префикс константы позволяет модифицировать адресное поле команды. Префикс операнда для команды выгрузки задает операцию пересылки данных в памяти.

Команды вызова программ имеют длину 2 байта и содержат 12-разрядный адрес передачи управления, который позволяет прямо адресовать 4 Кбайт текущего или нулевого банка ПЗУ. Префикс константы позволяет задавать значение 4 старших разрядов счетчика команд и выполнять передачу управления в любой банк ПЗУ.

Команды проверки и ветвления имеют длину 2 байта. Первый байт команды содержит код проверки и тип данных. Старший разряд второго байта содержит условие передачи управления, остальные разряды содержат 7-разрядное смещение адреса передачи управления в дополнительном коде относительно счетчика команд. Префикс константы расширяет 7-разрядное смещение до 11 (15) разрядов.

Процессор имеет двухступенчатый конвейер, обеспечивающий выполнение байта команда за один такт. Обработка 16 и 32-разрядных данных требует от 1(2) до 3(6) дополнительных тактов соответственно в зависимости от обработки данных на стеке или в памяти, где требуется дополнительный такт на запись результата.

В приложении приведена оценка затрат памяти процессора M8 на выполнение типовых операций при обработке 8 и 32 разрядных данных.

В таблице 32 приведены сравнительные оценки процессора M8, МК 8051, ARM и AVR при обработке 8 и 32-разрядных данных.

Таблица 32 Сравнительная оценка процессора M8, МК 8051, ARM и AVR

МК		**AVR**	**8051**	**ARM**	**M8**
8 разрядов	Байт	7.01	5.15	8.85	4.34
	Оценка памяти	**1.62**	**1.19**	**2.04**	**1.00**
	Тактов	4.16	6.86	7.09	4.34
	Оценка производительности	**0.96**	**1.58**	**1.63**	**1.00**
32-разряда	Байт	14.37	10.56	8.85	4.78
	Оценка памяти	**3.31**	**2.43**	**2.04**	**1.00**
	Тактов	8.53	14.06	7.09	6.04
	Оценка производительности	**1.40**	**2.30**	**1.17**	**1.00**

Таким образом процессор M8 при обработке 8-разрядных данных обеспечивает сокращение программного кода более чем в полтора раза по сравнению с МК AVR при практически равной с ним производительности. При обработке 32-разрядных данных процессор M8 обеспечивает более чем трехкратное сокращение программного кода по сравнению с МК AVR и двухкратное по сравнению МК при несколько большей производительности.

У процессора M8 есть возможность повысить производительность за счет использования 32-разрядной шины данных ОЗУ. За счет этого число тактов на обработку 32-разрядных данных в соответствии с приложением сокращается до 5.1, а это означает, что процессор M8 опережает по производительности 32-разрядный процессор ARM7 Thumb почти в полтора раза.

7. Заключение

Полученные результаты можно использовать для сравнительной оценки микроконтроллеров по объему программного кода и производительности, а также по другим параметрам. В качестве примеров приведены оценки объема программного кода процессора M8 и МК 8051, стоимости МК и трудоемкости их программирования.

7.1. Оценка эффективности процессора M8

Оценка длины команды. Анализ доли средней длины команд процессора M8 по расширениям типовых операций, приведенных в приложении, дает соотношение 0.2:0.6:0.2 между байтовыми, двухбайтовыми и трехбайтовыми командами. Соответственно средняя длина команды для процессора M8 составляет 2 байта или 0.87 от длины команд МК 8051.

По организации обработки данных и форматам команд сравниваемые процессоры практически не отличаются.

По составу команд в процессоре M8 сокращена доля команд пересылок на 0.18 за счет использования стека, исключены команды проверок, которые совмещены с операцией условного перехода. Доля вспомогательных команд увеличена до 0.21 из-за префиксных команд. В результате суммарная доля команд составляет 0.97 от доли команд МК 8051.

Таким образом, общая оценка эффективности процессора M8 по объему кода составляет 0.87*1*1*0.97 или 0.84 от объема кода МК 8051, что практически полностью совпадает с оценкой 0.82, приведенной в таблице 32.

7.2. Оценка стоимости МК

Для оценки стоимости МК за основу принята относительная площадь кристалла. В площади кристалла учитываются площади процессора, ПЗУ, ОЗУ и периферии, а также коэффициент технологии (проектных норм).

В таблице 33 приведены оценки площади МК и соответственно их стоимости. Взятое соотношение площадей приблизительно соответствует 18 контактному микроконтроллеру с объемом Flash памяти программ 16 Кб, ОЗУ – 4 Кб и типовому набору периферии.

Относительная площадь RISC процессоров сокращена по сравнению с CISC процессором 8051 на 40%, доля 32-разрядного процессора ARM увеличена в 4 раза.

Относительная площадь ПЗУ взята из оценок объема программного кода при обработке 32-разрядных данных (таблица 29).

Для процессора UNC площадь ОЗУ увеличена на 20%, поскольку в нем используется двухпортовая память.

Коэффициент проектных норм для всех микроконтроллеров установлен одинаковым.

Таблица 33 Оценка стоимости МК

Параметр	8051	PIC	AVR	UNC	ARM	M8
Процессор	1.00	0.60	0.60	0.60	2.60	0.65
ПЗУ	1.00	1.33	1.36	1.21	1.72	0.82
ОЗУ	1.00	1.00	1.00	1.20	1.00	1.00
Периферия	1.00	1.00	1.00	1.00	1.00	1.00
Всего	4.00	3.93	3.96	4.01	6.32	3.47
Проектные нормы	1.00	1.00	1.00	1.00	1.00	1.00
Итого	4.00	3.93	3.96	4.01	6.32	3.47
Оценка	**1.15**	**1.13**	**1.14**	**1.15**	**1.82**	**1.00**

7.3. Оценка трудоемкости программирования

Оценка трудоемкости программирования выполнена на основе монографии М.Х.Холстеда «Начала науки о программах» М.Финансы и статистика. 1981.

Автор показывает, что трудоемкость программирования пропорциональна квадрату длины программы. Длина программы определяется как произведение:

$L = (N_{коп} + N_{оп}) * ln_2 (n_{коп} + n_{оп}) ,$

где $N_{коп}$, $N_{оп}$ – общее число отдельных операций и операндов в программе;

$n_{коп}$, $n_{оп}$ – суммарный словарь оригинальных операций и операндов.

В таблице 34 приведены оценки трудоемкости программирования МК. Поскольку МК 8051 имеет CISC архитектуру с одноадресной системой команд, для него введен коэффициент 1.1. В соответствии с рекомендациями работы Ф.Брукса «Мифический человеко-месяц» в расчетах трудоемкости использована степень 1.5 от длины программы.

Таблица 34 Оценка трудоемкости программирования

Параметр	8051	AVR	UNC	ARM	M8
Число команд (таблица)	1.00	1.36	1.01	1.72	0.82
Число переменных (таблица)	1.60	2.03	1.46	2.25	1.46
Сумма	2.60	3.39	2.47	3.97	2.28
Сложность команд	1.10	1.00	1.00	1.00	1.00
Итого	2.86	3.39	2.47	3.97	2.28
Оценка длины программы	**1.25**	**1.49**	**1.08**	**1.74**	**1.00**
Оценка трудоемкости программирования	**1.40**	**1.82**	**1.12**	**2.30**	**1.00**

Таким образом, регистровая архитектура при программировании на машинном языке вдвое увеличивает трудоемкость программирования по сравнению с архитектурой, ориентированной на обработку данных в памяти.

Приложение

МК 8051

Операция	Вес	Машинный код	Доля	Байт	Вес	Байт	Такт
Mov d,s/#	0.91	mov direct,#	0.30	3	0.90	2.42	0.60
		mov Rn,#	0.15	2	0.30		
		mov @Ri,#					
		mov direct,direct	0.30	3	0.90		
		movx A,@Ri mov direct,A					
		mov direct,Rn	0.16	2	0.32		
		mov A,@Ri/Rn mov Rn,A					

Un d,s	0.05	mov A,direct
		rxx A
		mov direct,A
			0.046	5	0.23	0.40	0.01

		mov direct,@Ri un direct	0.002	4	0.01		

		mov A,@Ri
		un A
		mov @Ri,A
			0.002	3	0.01		

un d	0.09	un direct	0.06	2	0.12		
		un Rn	0.03	1	0.03		
bin d,s,s/#	0.16	log direct,#	0.03	3	0.09	0.62	0.14

		mov A,direct
		bin A,#
		mov direct,A
			0.02	6	0.12		

		mov A,Rn
		bin A,#
		mov Rn,A
			0.02	4	0.08		

		mov A,@Ri
		bin A,#
		mov direct,A
			0.01	5	0.05		

		mov A,direct log direct,A	0.02	4	0.08		
		mov A,Rn log direct,A	0.02	3	0.06		

		mov A,direct
		bin A,direct
		mov direct,A
			0.01	6	0.06		

		mov A,Rn
		bin A,Rn
		mov Rn,A
			0.01	3	0.03		

		mov A,@Ri
		bin A,direct
		mov direct,A
			0.006	6	0.04		

		mov A,@Ri
		bin A,Rn
		mov Rn,A
			0.004	3	0.01		

tst s,s/#	0.05	mov A,direct/# bin A,direct	0.03	4	0.12	0.14	-
		mov A,Rn bin A,Rn	0.02	2	0.04		
		mov A,@Ri bin A,@Ri					
bxx	0.33	cjne direct,#,rel	0.05	3	0.15	0.76	0.20
		cjne Rn,#,rel	0.04	2	0.08		
		jxx rel	0.17	2	0.36		
		jbc bit,rel	0.02	3	0.06		
		djnz Rn,rel	0.04	2	0.08		
		djnz direct,rel	0.01	3	0.03		
jmp	0.09	jmp adr11	0.09	2	0.18	0.18	0.18
call	0.18	acall adr11	0.15	2	0.30	0.43	0.43
		lcall adr16	0.03	3	0.09		
ret	0.04	Ret	0.04	1	0.04		
прочие	0.10		0.10	2	0.20	0.20	0.20
					Всего байт	**5.15**	1.66
					Итого тактов	5.15	**6.86**

Примечание. В столбце «Вес» серым фоном выделены команды, которые требуют дополнительного такта на запись результата операции в память, а также передачи управления.

МК PIC16

Операция	Вес	Машинный код	Доля	Байт	Вес	Байт	Такт
mov d,s/#	0.91	movlw k movwf f	0.91	4	3.64	3.64	-
un d,s	0.05	movf f,0 movlw f unf f	0.05	6	0.30	0.48	-
un d	0.09	unf f	0.09	2	0.18		
bin d,s,s/#	0.16	movf f,0 binf f,0 / k movlw f	0.10	6	0.60	0.84	-
		movf f,0 binf f,1	0.06	4	0.24		
tst s,s/#	0.10	movf f,0 binf f,0	0.10	4	0.40	0.40	-
		movf f,0 loglw k					
bxx	0.33	cpfsxx f	0.33	2	0.66	1.32	0.33
		decfsz f					
		btfsx f,b					
		goto k	0.33	2	0.66		
jmp	0.09	goto k	0.09	2	0.18	0.18	0.18
call	0.18	call k	0.22	2	0.44	0.44	0.44
ret	0.04	ret					
		movlw k option	0.05	4	0.20	0.20	-
					Всего байт	**7.50**	0.95
					Итого тактов	7.50	**8.45**

Примечание. В столбце «Вес» серым фоном выделены команды, которые требуют дополнительных тактов на передачу управления.

МК AVR

Операция	Вес	Машинный код	Доля		Вес	Байт	Такт
mov d,s/#	0.91	ldi Rd,k sts m,Rr	0.20	4-6 (5)	1.00	3.06	-
		ldi Rd,k	0.22	2	0.44		
		ldi Rd,k st X,Rr	0.03	4	0.12		
		lds Rd,m sts m,Rr	0.10	4-8 (6)	0.60		
		lds Rd,m	0.10	2-4 (3)	0.30		
		mov Rd,Rr	0.22	2	0.44		
		ld Rd,X sts m,Rr	0.02	6	0.12		
		st X,Rr	0.02	2	0.04		
un d,s	0.05	lds Rd,m un Rd (sts m,Rr)	0.045	4-10 (6)	0.27	1.12	-
		ld Rd,X un Rd st X,Rr	0.005	6	0.03		
un d	0.15	un Rd	0.09	2	0.18		
		lds Rd,m un Rd sts m,Rr	0.06	6-10 (8)	0.64		
bin d,s,s/#	0.16	lds Rd,m lds Rd,m bin Rd,Rr sts m,Rr	0.02	8-14 (11)	0.22	0.72	-
		lds Rd,m bin Rd,Rr/k sts m,Rr	0.02	6-10 (8)	0.16		
		lds Rd,m bin Rd,Rr st X,Rr	0.01	6-8 (7)	0.07		
		lds Rd,m bin Rd,k	0.01	4-6 (5)	0.05		
		bin Rd,Rr st X,Rr	0.01	4	0.04		
		bin Rd,Rr/k	0.09	2	0.18		
tst s,s/#	0.15	lds Rd,m lds Rd,m cp Rd,Rr	0.03	6-10 (8)	0.24	0.57	-
		lds Rd,m cp Rd,k /Rr	0.03	4-6 (5)	0.15		
		cp Rd,Rr/k	0.09	2	0.18		
bxx	0.33	brXX	0.33	2	0.66	0.66	0.17
jmp	0.09	rjmp k	0.09	2	0.18	0.18	0.09
call	0.18	rcall k	0.14	2	0.28	0.34	0.22
		call k	0.04	4	0.08		
ret	0.04	Ret	0.04	2	0.08		
push/pop		push/pop	0.18	2	0.36	0.36	0.18
					Всего байт	**7.01**	0.66
					Итого тактов	3.50	**4.16**

МК MARC4

Операция	Доля	Машинный код	Доля	Байт	Вес	Байт	Такт
mov d,s/#	0.73	[X]@ ($xx)/lit [Y]! ($xx)	0.65	4	2.60	2.96	-
		[X]@ ($xx) [Y]! ($xx)	0.18	2	0.36		
un d,s	0.05	[X]@ ($xx) un [Y]! ($xx)	0.05	5	0.25	1.20	-
un d	0.19	[X]@ ($xx) un [Y]! ($xx)	0.19	5	0.95		
bin d,s,s/#	0.16	[X]@ ($xx) / lit (lit) [X]@ ($xx) bin [Y]! ($xx)	0.16	7	1.12	1.12	-
tst s,s/#	0.15	[X]@ ($xx) / lit (lit) [X]@ ($xx) bin	0.15	5	0.75	0.75	-
bxx	0.33	bra $xxx	0.08	2	0.16	0.41	0.17
		sbra $xxx	0.25	1	0.25		
jmp	0.09	set_b bra $xxx	0.02	3	0.06	0.20	0.09
		set_b sbra $xxx	0.07	2	0.14		
call	0.18	call $xxx	0.09	2	0.18	0.32	0.22
		scall $xxx	0.09	1	0.09		
ret	0.04	ret	0.04	1	0.04		
					Всего байт	**6.96**	0.48
					Итого тактов	6.96	**7.44**

МК UNC80

Операция	Машинный код	Доля	Байт	Вес	Вес	Такт
mov d,s/#	mov d,s/k	0.91	2	1.82	1.82	-
un d,s	mov d,s un d	0.05	4	0.20	0.58	-
un d	un d	0.19	2	0.38		
bin d,s,s/#	bin d,s/k	0.14	2	0.28	0.36	-
	mov d,s bin d,s	0.02	4	0.08		
tst s,s/#	cmp s,s/k	0.17	2	0.34	0.34	-
bxx	bcc	0.33	2	0.66	0.66	0.33
jmp	br	0.07	2	0.14	0.22	0.18
	jmp	0.02	4	0.08		
call	jsr	0.18	4	0.72	0.72	0.44
ret	ret	0.04	2	0.08	0.08	
Вспомогательные	ldl r,k	0.21	2	0.42	0.42	
				Всего байт	**5.20**	0.95
				Итого тактов	5.20	**6.15**

МК ARM7 Thumb

Операция	Вес	Машинный код	Доля	Байт	Вес	Байт	Такт
mov d,s/#	0.91	mov Rd,Rs	0.16	2	0.32	3.31	0.89
		mov Rd,#K	0.16	4	0.64		
		(mov Rb,#K) ldr Rd,[Rb,Ro/Imm] str Rd,[Rb,Ro/Imm]	0.59	2-6 (4)	2.35		
un d,s	0.05	(mov Rb,#K) (ldr Rd,[Rb,Ro/Imm]) un Rd (str Rd,[Rb,Ro/Imm])	0.24	2-10 (6)	1.44	1.44	0.48
un d	0.19						
Bin d,s,s/#	0.16	(mov Rb,#K) (ldr Rd,[Rb,Ro/Imm]) (ldr Rd,[Rb,Ro/Imm]) bin Rd,Rs (str Rd,[Rb,Ro/Imm])	0.16	2-12 (7)	1.12	1.12	0.40
tst s,s/#	0.15	(mov Rb,#K) (ldr Rd,[Rb,Ro/Imm]) (ldr Rd,[Rb,Ro/Imm]) Cmp/bin Rd,Rs/#K	0.15	2-10 (6)	0.90	0.90	0.23
bxx	0.33	Bcc	0.33	2	0.66	0.66	0.17
jmp	0.09	b …	0.09	2	0.18	0.18	0.09
call	0.18	push … RL bl …	0.22	4	0.88	0.88	0.22 0.18
ret	0.04	pop … PC bx LR					
push/pop	0.18	push/pop	0.18	2	0.36	0.36	
					Всего байт	**8.85**	2.66
					Всего тактов	4.43	**7.09**

Примечание. В столбце «Доля» серым фоном выделены команды, которые требуют дополнительных тактов на доступ к памяти и выполнение передач управления (последний столбец таблицы).

МК М8

Операция	Вес	Машинный код	Доля	Байт	Вес	Итого	Байт		Такт		Такт	
mov d,s/#	0.73	M st M	0.28	2	0.56	1.61			2	0.56		
		Ldl L	0.06	2	0.12		1	0.06				
		Ld M	0.12	1	0.12				1	0.12		
		st M					1	0.27				
		stl M L	0.27	3	0.81							
un d,s	0.05	(M ld) Un St M	0.05	2-3 (2.7)	0.14	0.32			1-2 (1.7)	0.24	1	0.14
Un d	0.09	M un	0.09	2	0.18							
bin d,s,s/#	0.16	(M ld) (M ld) Bin / binl L St M	0.04	3-4 (3.7)	0.15	0.46	0-1 (0.5)	0.02	2-4 (3)	0.12	1	0.04
		M log N	0.04	2	0.08				2-3 (2.5)	0.10	1	0.04
		(M ld) M bin	0.04	2-3 (2.7)	0.11		1	0.04	1	0.04		
		M binl L	0.04	3	0.12							
bxx	0.33	(M ld) M bxx A	0.17	3-4 (3.7)	0.63	1.12			2-3 (2.5)	0.42	1	0.17
		(M) bxxl L A	0.16	3-4 (3.7)	0.59		1	0.16				
jmp	0.09	Br A	0.09	2	0.18	0.18	-	-	-	-		
call	0.18	Jsr A	0.18	2	0.36	0.40	-	-	-	-		
ret	0.04	ret	0.04	1	0.04		-	-	-	-		
K	0.21	k/ldr k	0.21	1-2(1.2)	0.25	0.25	-	-	-	-		
Всего байт/тактов на обработку 8-32 разрядных данных						**4.34**	0.27	0.55	0.79	1.58	0.20	0.39
							0.17	1.65	0.47	4.74	0.12	1.17
Итого байт/тактов						**4.34**	0.44	**4.78**	1.26	**6.04**	0.32	**5.10**

Примечание. В последних столбцах приведены дополнительные такты на обработку 16 и 32-разрядных данных при 8- и 32-разрядной шине данных. В последней строке приведена средняя оценка числа тактов.