АНАЛИЗ МАШИННОГО КОДА МИКРОКОНТРОЛЛЕРА 51-ГО СЕМЕЙСТВА

А. Н. Моксяков, С. Н. Гончаров, М. В. Марунин, И. А. Немченко

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

Язык машинных кодов – запись машинной программы в виде последовательности восьмиричных или 16-ричных цифр, где группа цифр задает значение байта или слова машинной программы. Для различных архитектур электронно-вычислительных машин (ЭВМ), микропроцессоров (МП) и микроконтроллеров (МК) используются различные языки ассемблера. Ассемблер – программа, выполняющая трансляцию программы на языке ассемблера в машинный код.

Дизассемблер − транслятор, преобразующий машинный код, объектный файл или библиотечные модули в текст программы на языке ассемблера. По режиму работы с пользователем дизассемблеры делятся на автоматические и интерактивные. Основная трудность при работе дизассемблера − отличить данные от машинного кода, поэтому на первых проходах автоматически или интерактивно собирается информация о границах процедур и функций, а на последнем проходе формируется итоговый листинг. Интерактивность позволяет улучшить этот процесс, так как, просматривая дамп дизассемблируемой области памяти, программист может сразу выделить строковые константы, дать содержательные имена известным точкам входа, прокомментировать разобранные им фрагменты программы. Чаще всего дизассемблер используют для анализа программы (или ее части), исходный текст которой неизвестен, с целью модификации, копирования или взлома. Реже − для поиска ошибок в программах и компиляторах, а также для анализа оптимизации создаваемого компилятором машинного кода.

Актуальность вопроса связанного с дизассемблированием программного кода вытекает из повсеместного использования микропроцессорных систем и форс-мажорными обстоятельствами с документами различного рода на микропроцессорную систему (например, потеря блок-схемы алгоритма работы «программное обеспечение»).

В качестве «подопытного кролика» используется МК 51-го семейства фирмы Atmel. На основе этой архитектуры выпускается огромное число различных контроллеров, содержащих на «борту» широкую гамму периферийных устройств – от портов до аналого-цифровых преобразователей, цифро-аналоговых преобразователей и интерфейса CAN. Кроме того, данный МК прост в изучении, эксплуатации и в достаточном количестве присутствует на работе. Серьезным аргументом в пользу дизассемблирования программ, написанных под МК 51-го семейства, является отсутствие бита защиты на считывание внутреннего содержимого резидентной памяти программ. Структурная схема микроконтроллера представлена на рис. 1.

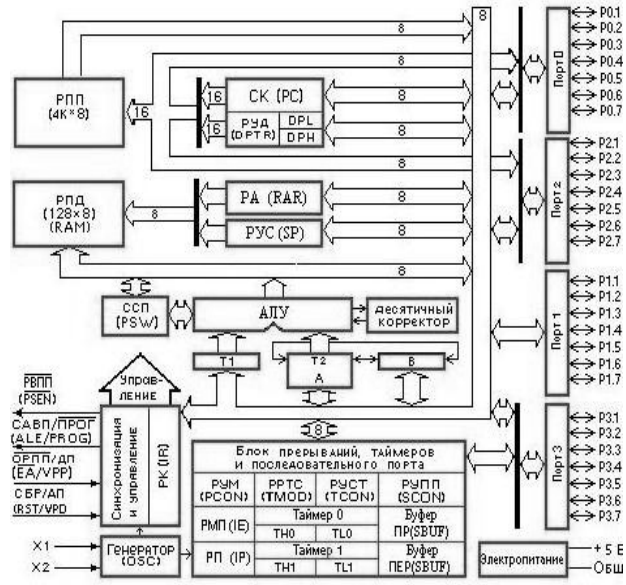

Рис. 1. Структурная схема микросхемы

Система команд МК включает в свой состав 111 основных команд. Длина команд составляет один, два или три байта, причем большинство команд (94 %) одно- или двухбайтовые. Первый байт команды любых типов и формата всегда содержит код операции (КОП). Второй и третий байты содержат либо адреса операндов, либо непосредственные операнды. Все команды выполняются за один или два машинных цикла (1,0 мкс при тактовой частоте 12 МГц соответственно), исключение составляют лишь команды умножения и деления, которые выполняются за четыре машинных цикла (4,0 мкс). Команды разделены на группы в соответствии с операндами и содержат следующую информацию: код операции, мнемоническое обозначение и комментарии. Избыточность машинного кода микроконтроллера бывает двух типов:

− избыточность систем команд;

− избыточность операнда.

Формат команды микроконтроллера 51-го семейства не содержит информационной избыточности ввиду отсутствия префиксов и служебных байтов.

Избыточность системы команд заключается в том, что в ней отсутствуют команды с кодами A5h. Избыточность операнда определяется в основном конфигурацией конкретной микропроцессорной системы.

Очевидно, что кроме структуры языка и типа программ существенное влияние на статистическое распределение команд оказывает компилятор, с помощью которого был получен исполняемый код. Логично предположить, что именно системой генерации кода можно объяснить существование малой группы команд с высокой вероятностью появления. Вероятность появления команд в 51-ом семействе микроконтроллеров стандартного проекта для компилятора Keil Software v.8.x приведена на рис. 2. Статистика набрана с использованием программного обеспечения, написанного на Borland C++ v3.1, HEX файл разобран по определенным критериям на команды и данные. На рис. 3 представлен машинный код программы, написанной для микроконтроллера 51-го семейства с использованием его стандартной периферии.

Разделение машинного кода на команды и данные производится для анализа структуры программного обеспечения, что позволяет модифицировать и верифицировать работу устройства путем сравнения шаблонного кода определенных функций и исследуемый машинный код.

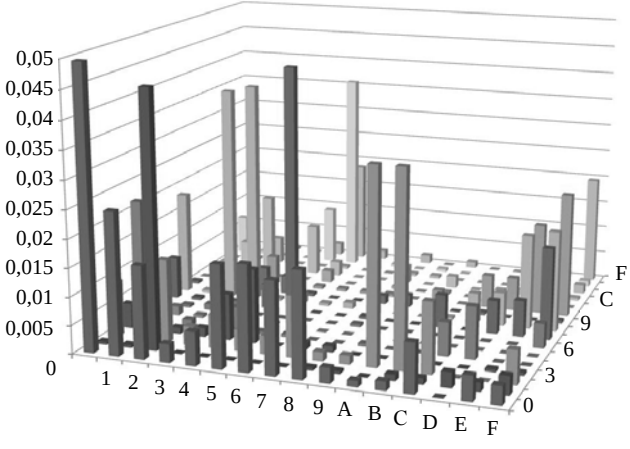

Рис. 2. Вероятность появления команд для Phyton Project-51

E4	9F	5E	DE	F9	0E	29	D9	C9	98	B0	D0
6C	AD	ED	B8	83	20	9A	BF	B3	B6	03	B6
16	83	E3	63	0B	12	94	64	3B	84	0D	6D
9E	B1	F0	0F	93	44	87	08	A3	D2	1E	01
06	E7	FE	D4	1B	76	89	D3	2B	E0	10	DA
8E	D5	D6	D6	A3	E8	A1	D1	93	7E	38	D8
36	4B	D8	0D	2B	DA	AF	0A	1B	4C	36	03
BE	79	CB	61	B3	8C	BC	66	83	1A	25	6F
26	2F	C5	BA	3B	BE	B2	BD	0B	28	2B	D4
AE	1D	9B	64	C2	B0	EC	63	F2	26	75	6A
57	13	95	BF	4A	82	E2	B8	7A	14	7B	B1
DF	21	86	D3	D2	D4	F1	D4	E2	42	68	DD
47	77	88	08	5A	E6	FF	0F	6A	70	66	06
CF	45	A0	0A	E2	78	D7	0D	D2	EE	4E	04
77	DB	AE	D1	6A	4A	D9	D6	5A	DC	40	DF
FF	F9	BD	BD	F2	1C	CA	BA	C2	8A	53	B3
67	BF	D3	66	7A	2E	C4	61	4A	B8	5D	68
EF	8D	AF	34	AF	35	8F	82	8E	83	E0	FF

Рис. 3. Машинный код

На данный момент работа ведется в области сбора статистики, что позволяет проверять критерии разделения команд и данных. А также параллельно анализируется машинный код наиболее часто встречающихся итераций.

Литература

1. Фомичев В. М. Дискретная математика и криптология. М.: ДИАЛОГ-МИФИ, 2003.

2. Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В. Основы криптографии. М.: Гелиос АРВ, 2001.

3. Сташин В. В. и др. Проектирование цифровых устройств на однокристальных микроконтроллерах. М.: Энергоатомиздат, 1990.