УДК 621.37.037
DOI: 10.30987/conferencearticle_5c19e6a04b4369.27533420

А.С. Веремчук, В.С. Панищев
(г. Курск, Юго-Западный государственный университет)

МЕТОДИКА РЕАЛИЗАЦИИ УНИВЕРСАЛЬНОГО ЭМУЛЯТОРА

Приведена методика реализации универсального эмулятора.

This thesis presents the method of implementation of the universal emulator.

Ключевые слова: универсальный эмулятор, процессор, команды вызова, программное обеспечение.

Keywords: universal emulator, processor, call commands, software.

Встроенные системы широко используются в различных системах обработки и передачи данных. Данные системы имеют множество функций, которые могут быть не заявлены производителем. Одним из способов выявления недокументированных функций является анализ программного кода (ПК).

В [1] представлен метод эмуляции выполнения дизассемблированного программного кода, с использованием ПО IDA Pro, который позволяет снизить трудоемкость, повысить эффективность анализа дизассемблированного программного обеспечения.

Алгоритм эмуляции выполнения инструкции выглядит следующим образом:

1. Чтение кода операции (КОП).
2. Получение операндов (значений регистров, ячеек памяти и пр.).
3. Вычисление инструкции.
4. Вычисление флагов при необходимости.

Структура эмулятора

В основе эмулятора [1] лежит универсальный вычислитель, который производит все арифметические и логические вычисления, а также производит взятие значения по указанному адресу. Вычислитель получает данные от блоков-регистров, ОЗУ и ПЗУ процессора. Для работы эмулятора необходим блок мнемонических конструкций команд, поставляющий вычислителю выражения для проведения арифметических и логических операций.

С целью практической реализации универсального эмулятора проведена детализация структурной схемы (рис. 1).

В центре схемы расположен универсальный вычислитель – вычислительный блок эмулятора. Как было упомянуто, с помощью этого блока производятся все вычислительные операции:

расчёт выражений, представляющих мнемонические конструкции для команд процессора;

расчёт условий и адресов переходов команд, выражений списка наблюдений;

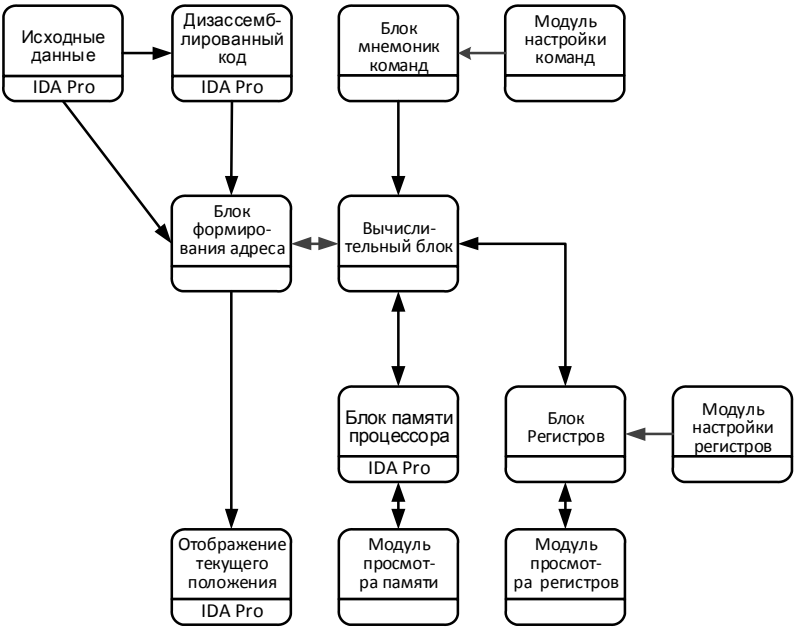

Рис.1. Структурная схема универсального эмулятора

Поскольку каждое семейство микропроцессоров имеет собственный набор команд, часто включающий специфические для данного семейства процессоров функции, возникает необходимость в создании универсального описателя, который позволял бы с помощью стандартного набора операций воспроизводить любую функцию (функции), выполняемую конкретной командой процессора.

Для этого необходим блок-мнемоник команд, в котором для каждой команды процессора хранятся описанные на формализованном языке, мнемоники выполняемых командой операций.

Блок настройки мнемоник позволяет пользователю создавать и изменять мнемонические конструкции для каждой команды процессора.

Блок формирования адреса команды анализирует исходящие ссылки текущей команды, а в случае команд вызова и перехода производит проверку условий перехода и вычисление адреса перехода.

Блок регистров процессора содержит список регистров, разделённых по группам, в пределах которых они имеют одинаковые свойства разрядности, функционального назначения.

Модуль настройки регистров процессора служит для создания и настройки пользователем списка регистров эмулятора, выбирая их из списка регистров, предоставляемого процессорным модулем. Предусмотрена возможность дополнять список регистров эмулятора новыми регистрами, в случае их отсутствия. Предоставляет возможность задавать названия отдельным битам в пределах регистра, а также группам бит (произвольному числу бит, идущих подряд) в пределах одного регистра.

Модуль просмотра и редактирования содержимого регистров позволяет пользователю контролировать и изменять значения отдельных регистров во время работы отладчика эмулятора.

Модуль просмотра ячеек памяти служит для контроля пользователем значений отдельных байт памяти во время работы отладчика эмулятора.

Блок памяти процессора эмулирует ОЗУ и ПЗУ устройства. В роли этого блока выступает ядро IDA Pro.

Универсальный эмулятор, построенный по приведённой схеме, реализован в виде подключаемого к дизассемблеру IDA Pro модуля на языке C++.

Список литературы

1. Designing an object-oriented decompiler – Department of Software Engineering and Computer Science Blekinge Institute of Technology // D. Eriksson. –2002. –P.23.

Материал поступил в редколлегию 14.10.18.

УДК. 004.942 , 004.896, 519.63
DOI: 10.30987/conferencearticle_5c19e6a0ba03e2.48318163

И.В. Доненко, А.В. Доненко, В.А. Лукьяненко
(г. Симферополь, Крымский федеральный университет
им. В.И. Вернадского)

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ НЕЛИНЕЙНЫХ ПАРАБОЛИЧЕСКИХ УРАВНЕНИЙ В ФРАКТАЛЬНОМ ОТОБРАЖЕНИИ ПОЛЯ ЛАЗЕРНОГО ИЗЛУЧЕНИЯ В СВОБОДНОМ ПРОСТРАНСТВЕ

Рассмотрена эволюция фрактальной дифракционной картины с математической точки зрения. Решая нелинейные параболические уравнения для фрактальных отображений, появилась возможность моделировать эволюцию дифракционных картин.

Modern fractal theory is experiencing rapid growth with the development of mathematical and computer modeling. Solving the non-linear parabolic equations for fractal mappings, it was possible to model the evolution of diffraction patterns.

Ключевые слова: фрактальная картина, нелинейные параболические уравнения, математика, математическое моделирование, фрактал, оптика

Keywords: fractal picture, nonlinear parabolic equations, mathematics, mathematical modeling, fractal, optics

В последние годы решение таких нелинейных параболических уравнений, разрушающихся за конечное или бесконечное время (см. эволюцию треугольника Серпинского или ковра Серпинского [1]), т.е. режимы с обострением, стали активно изучаться – это и не могло оставить нас в стороне, авторы решили рассмотреть с математической точки зрения