

УДК 004.383

Романчук В.А.

Рязанский государственный университет имени С.А.Есенина

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА МОДЕЛИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА БАЗЕ НЕЙРОПРОЦЕССОРА

Проводится описание программного комплекса для анализа, моделирования и отладки нейропроцессорных систем и устройств.

Целью работы является разработка программного комплекса, предназначенного для моделирования сложных вычислительных систем на базе нейропроцессора. Работа выполнена в соответствии с грантом РФФИ №12-07-97516 р-центр-а «Моделирование вычислительных систем на базе нейропроцессоров».

Для реализации средств моделирования в соответствии с методикой, был разработан программный комплекс "НейроКС". Языком программирования был выбран язык С# из-за его устойчивости к сбоям, независимости от аппаратного обеспечения, независимости от операционной системы, возможности создания современного интерфейса, высокого уровня оптимизации кода. Средой разработки была выбрана современная среда программирования Visual Studio 2010, разработанная фирмой Microsoft.

Исходные тексты включают около 150 тыс. строк программного кода, около 100 модулей. Разработаны 2 внешние dll библиотеки: NeuroProc, содержащая класс модели процессора семейства NM640x и Config, включающая настройки комплекса и функции доступа к реальному процессору (эмулятору).

Интерфейс комплекса является многодокументным и состоит из 3-х частей:

- область навигации (для перехода между экземплярами модулей);
- область системных сообщений (для вывода системных сообщений при работе с программой);
- рабочая область (для представления экземпляров модулей).

Структурная схема программного комплекса представлена на рис. 1.

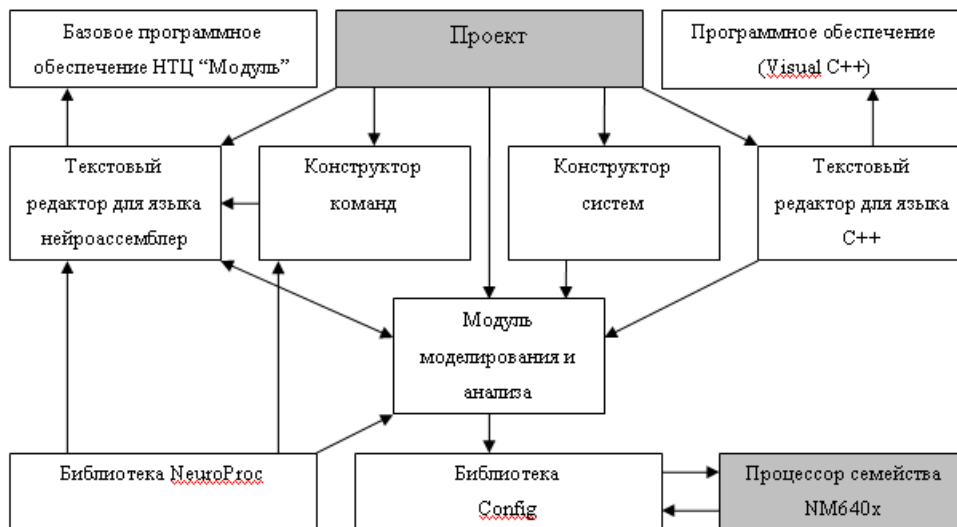


Рис.1. Структурная схема программного комплекса анализа и моделирования

Рассмотрим функциональные возможности каждой подсистемы комплекса:

1. "Текстовый редактор", "Текстовый редактор для C++". Подсистемы предназначены для создания и автоматизации процесса создания программных кодов. Функциональными возможностями являются:

- сохранение, загрузка файла (*.asm, *.txt, *.cpp);
- настройка параметров страницы, предварительный просмотр, печать;
- операция "копировать", "вставить", "вырезать", "удалить", "выделить все", "удалить все", "копировать все";
- отмена и повтор предыдущего действия;
- добавление, удаление, переход по закладке;
- поиск первого вхождения строки, поиск и замена;
- переход на строку, вставка даты, вставка директив процессора, вставка комментариев;
- подсветка синтаксиса; обновление подсветки синтаксиса;
- отправка кода программы во внешние программы: nmcc.exe, asm.exe, libr.exe, dump.exe, emurun.exe, temu.exe из базового программного обеспечения НТЦ "Модуль".

Функциональные возможности модуля "Текстовый редактор C++" аналогичны приведенным выше, но отсутствуют возможности:

- подсветка синтаксиса; обновление подсветки синтаксиса;
- вставка директив процессора;
- отправка программы в nmcc.exe, asm.exe, libr.exe, dump.exe, emurun.exe, temu.exe из базового программного обеспечения НТЦ "Модуль".

Вместо этого, реализована возможность:

- отправка кода программы во внешний компилятор C++.

Вспомогательные средства для тестирования и отладки, помимо специальных программ базового программного обеспечения, реализованы в модуле "Анализатор программного кода". Данный модуль позволяет отладить команду без запуска на реальном процессоре (в отличие от программ `temu.exe` и `emugun.exe`).

Рассмотрим функциональные возможности подсистемы "Анализатор программного кода":

- сохранение, загрузка файла (*.proс);
- настройка параметров страницы, предварительный просмотр, печать;
- выбор команды путем активирования элементов интерфейса;
- моделирование выполнения команды на процессоре;
- представление справочной информации о команде;
- расчет времени потерь из-за внутреннего параллелизма процессора;
- генерация шаблона кода на основе команды;
- очистка регистров, памяти, стека, матрицы и т.д.;
- перенос новых значений на место старых, что обеспечивает возможность непрерывного цикла программирования;
- аналитический и графический вид представления данных;

Выбор структуры реализован в модуле "Конструктор систем".

Рассмотрим функциональные возможности текстового редактора для создания и автоматизации процесса создания программных кодов:

- сохранение, загрузка файла (*.kons);
- настройка параметров страницы, предварительный просмотр, печать;
- аналитическое и графическое представление конвейерной, векторной, конвейерно-векторной и векторно-конвейерной структур;

Анализ выполнения программы производится в модуле моделирования и анализа. Исходными данными для анализа являются: исходный код программы, дополнительный C++ код (необязателен) и архитектура системы.

Функциональными возможностями модуля являются:

- сохранение, загрузка файла (*.term);
- загрузка данных и получение статистической информации о подпрограммах;
- моделирование полного цикла и по шагам;
- анализ выполнения подпрограммы на процессоре, анализ выполнения программы на НПС;
- аналитическое и графическое представление результатов

анализа и моделирования;

- доступ к аппаратной части.

Результатами анализа являются:

- Для каждой подпрограммы
 - Число команд в подпрограмме;
 - Время работы процессора;
 - Время полного цикла процессора;
 - Время простоя;
 - Коэффициент загрузки;
 - Простои в командах;
 - Простои декодирования, выполнения, перехода,

загрузки.

- Для НПС
 - Время работы системы;
 - Время работы на одном процессоре;
 - Количество используемых процессоров;
 - Коэффициент эффективности;
 - Время выигрыша;
 - Время простоя;
 - Коэффициент загрузки системы;
 - Простои в командах;
 - Простои декодирования, выполнения, перехода,

загрузки.

При выборе какого либо процессора есть возможность получения доступа к реальному процессору (вместо моделирования) и работы с ним.

Функциями работы с процессором являются:

- Функции для Host части (компьютер);
 - показ версии библиотеки связи;
 - установка времени ожидания;
 - получения числа доступных модулей;
 - получение дескриптора устройства;
 - закрытие дескриптора;
 - перезагрузка устройства;
 - загрузка кода начальной инициализации;
 - получение дескриптора доступа к процессору;
 - определение состояния устройства;
 - отправка прерывания на процессор;
 - закрытие дескриптора доступа;
 - загрузка и исполнение программы;
 - синхронизация с nm-процессом (массивом);
 - синхронизация с nm-процессом (скаляром);
 - запись в память;
 - чтение из памяти;

- перезагрузка устройства и его инициализация.
- Функции для NM части (процессор);
 - определение номера процессора;
 - синхронизация с host-процессом (массивом);
 - синхронизация с host-процессом (скаляром).

Литература

1. Агуров, П.В. С#. Сборник рецептов / П.В. Агуров. – СПб.: БХВ-Петербург, 2007. – 432 с.
2. Ватсон, К. С# / К.Ватсон [и др.]. – СПб.: Издательство "Лори", 2006. – 862 с.