РОССИЙСКАЯ АКАДЕМИЯ НАУК
ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ

На правах рукописи

ОРЛОВСКАЯ ЕЛЕНА ВИЛЕНОВНА

УДК 681.3.06

ИССЛЕДОВАНИЕ МЕТОДОВ ФОРМАЛЬНОЙ СПЕЦИФИКАЦИИ ПРОГРАММНО-АППАРАТНЫХ СИСТЕМ, ОБЕСПЕЧИВАЮЩИХ НАДЕЖНОСТЬ СИСТЕМ И ПОВТОРНОЕ ИСПОЛЬЗОВАНИЕ СПЕЦИФИКАЦИИ

Специальность: 05.13.11 - Математическое и программное обеспечение вычислительных машин, комплексов, сетей и систем

Автореферат
диссертации на соискание ученой степени
кандидата технических наук

МОСКВА 1992

Работа выполнена в Институте проблем информатики (Казанский филиал)

НАУЧНЫЕ РУКОВОДИТЕЛИ - Кандидат физико-математических наук, доцент Семик В. П., Кандидат технических наук, доцент Ландау И. Я.

ОФИЦИАЛЬНЫЕ ОППОНЕНТЫ - Доктор технических наук, начальник отд. Козмидиади В. А., Кандидат технических наук, ст.н.с. Андерс Б. Н.

ВЕДУЩЕЕ ПРЕДПРИЯТИЕ - ИТМиВТ

Защита состоится " " 1992г. в час. мин. на заседании специализированного совета Д 003.56.01 Института проблем информатики: 117900, Москва, ГСП-1, В-334, ул. Вавилова, 30/6. Телефон совета: 135-61-17. С диссертацией можно ознакомиться в библиотеке Института проблем информатики.

Автореферат разослан " " 1992 г

Ученый секретарь специализированного совета доктор технических наук С. Н. Гринченко

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

АКТУАЛЬНОСТЬ ТЕМЫ

Задача проектирования большой программно-аппаратной системы достаточно сложна. Одним из ключевых моментов ее решения является применяемая технология проектирования. От нее в первую очередь зависит надежность создаваемой системы и стоимость ее разработки.

Разработка систем идет, как прешло, в обстановке неточных, постоянно изменяющихся и конфликтующих между собой требований заказчика. Например, удешевление технических средств позволяет перенести на уровень оборудования часть функций, которые ранее выполнялись программным обеспечением (ПО). Такая потребность обычно вызывает необходимость в перепроектировании всей системы. Кроме того, рост мощности и падение стоимости вычислительных систем значительно опережает скорость разработки ПО. - Следовательно, продолжает представлять интерес повторное использование ПО, в частности, его мобильность.

Общеизвестно, что стоимость ошибок, допущенных на начальных этапах разработки системы чрезвычайно высок?. Например, исправление ошибки в уже изготовленной микросхеме современного уровня сложности равносильно перепроектированию микросхемы. Важным условием проектирования сложных систем является применение методов и инструментальных средств, позволяющих осуществить сквозную формализации и автоматизацию всех этапов жизненного цикла программно-аппаратной системы, а также адекватно использовать результаты предыдущих этапов разработки системы на последующих.

Из вышесказанного следует, что исследование, развитие и разработка формальных методов проектирования программно-аппаратных систем предоставляет большой интерес. Чрезвычайно актуальной становится задача создания инструментальной среды поддержки разработки их спецификаций.

ЦЕЛЬЮ РАБОТЫ является исследование, разработка и реализация новых методов спецификаций программно-аппаратных систем на принципах повышения надежности разрабатываемых систем, их повторного использования и комфортности процесса создания их спецификаций.

В соответствии с поставленной целью, основными задачами

диссертационной работы являются:

1.Выбор, развитие и разработка способов создания спецификаций систем, позволяющих строить надежные мобильные программно-аппаратные системы.

2.Исследование возможности уменьшения затрат на создание таких систем.

3.Создание на базе проведенных исследований рабочего макета интегрированной среды поддержки разработки спецификаций программно-аппаратных систем.

4.Проектирование с помощью созданного рабочего макета инструментальной среды поддержки разработки формальных спецификаций большой аппаратно-программной системы, анализ созданных спецификаций и инструментальной среды на удовлетворение предъявленным к ним требованиям надежности, повторного использования и комфортности разработки.

МЕТОДЫ ИССЛЕДОВАНИЯ

В диссертационной работе применялись метрологические способы оценки качества программ и макетирование.

НАУЧНАЯ НОВИЗНА

- Зыбрана формальная методология проектирования спецификаций систем, отличающаяся от традиционных возможностью перенести вопросы распределения ресурсов на более поздние этапы разработки системы и гарантировать достижение хороших метрологических оценок качества создаваемых спецификаций. Методология заключается в построении спецификаций, удовлетворяющих аксиоматике, гарантирующей корректность интерфейса между частями системы.

- Разработан метод создания комфортной среды проектирования спецификаций. Метод заключается в применении для описания системы графического акранного редактора. ориентированного на семантику аксиом выбранной методологии. В отличие от традиционных сред программирования, редактор скрывает от пользователя жесткую дисциплин разработки спецификаций, не допуская внесения ошибок интерфейс между частями системы.

- Разработан новый подход к проверке эквивалентности типов данных. В отличив от подходов, принятых в большинстве современных полиморфных языков программировалия. он рассматривает тип данных как совокупность операций, связывающих эти операции аксиом, констант и ограничений типа. Сопоставление этой информации у различных типов данных позволяет решить вопрос о

возможности применения одной и той же конструкции на различных типах данных в различных частях описания системы. Разработанный подход позволяет сочетать надежность проверим типов данных при сохранении высокого уровня полиморфизма.

- разработаны формальные методы функционально-эквивалентных преобразований алгоритмов, записанных в методологии формальных функциональных спецификаций. Функционально-эквивалентные преобразования позволяют, в частности, привести последовательный алгоритм к параллельно-конвейерному виду.

- Разрпботана архитекитура инструментальной среды проектирования формальных спецификаций программно-аппаратных систем. Каждая компонента среды имеет только информационные связи с остальными компонентами. Отличительной особенностью среды является зависимость ее от последовательности выполнения компонент и готовности спецификаций.

ГРАФИЧЕСКАЯ ЦЕННОСТЬ

Выбранные и разработанные методы и реализованный рабочий макет инструментальной среды позволяют формализовать и автоматизировать процесс разработки спецификаций больших программно-аппаратных систем различного назначения, в частности, позволяют описывать алгоритмы работы сложных БИС. Создаваемые спецификации могут быть далее применены для разработки надежных мобильных систем, выполняющихся в вычислительных средах с безличным архитектурным составом технических средств. Это в итоге позволяет достичь приемлемого для разработчика баланса стоимости/производительности/эффективности целевой системы.

Например, разработанное формальное описание алгоритма последовательного выполнения инструкций S2-разрядного микропроцессора с системой команд Intel 80386 применяется для построения как модели его последовательного поведения, так и для создания спектра параллельно-конвейерных моделей, позволяющих в итоге достичь динамики передачи сигналов между частями модели и моделью и внешней средой, соответствующей динамике взаимодействия между частями реального процессора Intel 80386, а также между процессором и вычислительной системой в целом.

ДОСТОВЕРНОСТЬ НАУЧНЫХ ПОЛОЖЕНИЙ И ВЫВОДОВ подтверждается

1. Математическим доказательством теорем о функциональной эквивалентности преобразований, позволяющих привести некоторые

последовательные алгоритмы к более быстрому параллельно-конвейерному виду.

2. Созданием рабочего макета инструментальной среды разработки формальных спецификаций.

3. Разработкой с помощью рабочего макета инструментальной среды описания алгоритма последовательного выполнения инструкций 82-разрядного микропроцессора.

4. Применением итого огисания, а также разработанных методов функционально-эквивалентных преобразований, для создания широкого спектра поведенческих моделей выбранного микропроцессора.

5. Анализом разработанного рабочего макета инструментальной среды и функционального описания процессора с точки зрения надежности, повторного использования и комфортности.

РЕАЛИЗАЦИЯ РЕЗУЛЬТАТОВ РАБОТЫ

Выполненные в диссертации исследования и разработки осуществлялись в соответствии с планами научно-исследовательских работ ИЛИ АН в период с 1989 г. по настоящее время, а также в порядке личной инициативы.

Научные и практические результаты диссертации использованы в следующих томах ИЛИ АН

- НИР. Мультипрограммная операционная система для 32-битной ПЭВМ 32НП (0С2). Моделирование процессора Intel 80386.

- НИР. Инструментальная среда функционального определения и моделирования микропроцессоров.

Использование результатов диссертационной работы подтверждено актами об использовании, приведенными в Приложении 2.

АПРОБАЦИЯ РЕЗУЛЬТАТОВ РАБОТЫ

Основные положения работы и отдельные результаты докладывались на международных, всесоюзных и республиканских конференцях, в частности:

- на ежегодных отчетчых научных конференциях КФ АН СССР, начиная с 1988 г..

- на международной конференции-ярмарке "Технология программирования 90-х". Киев. Май 1991 г. и других.

ПУБЛИКАЦИИ

За период с 1987 г. по настоящее время по теме диссертации опубликовано 9 работ.

СТРУКТУРА И ОБЪЕМ РАБОТЫ

Диссертация состоит из введения, пяти разделов, заключения, списка литературы и двух приложений. Основной текст изложен на 147 с. и содержит 52 рис. Список литературы содержит 124 библ. наименования. Приложения содержат 13 с. Всего 208 с.

НА ЗАЩИТУ ВЫНОСЯТСЯ

- архитектура инструментальной среды разработки формальных функциональных спецификаций;

- метод создания комфортной для разработчика среды, скрывающей жесткую дисциплину проектировантия формальных функциональных спецификаций. Среда не допускает внесения ошибок интерфейса между частями специфицируемой системы;

- метод проверки соответствия типов данных, основанный на понятиях эквивалентности типов и полиморфизма конструкций методологии формальных функциональных спецификаций;

- методы функционально-эквивалентных преобразований алгоритмов, записанных в методологии формальных функциональных спецификаций, к виду, отражающему удовлетворяющую разработчика динамику выполнения алгоритма в вычислительной среде с заданным составом исполнителей:

- применение методологии формальных функциональных спецификаций к описанию алгоритмов работы сложных БИС.

СОДЕРЖАНИЕ РАБОТЫ

ВО ВВЕДЕНИИ обосновывается актуальность исследуемых проблем, приводятся цели и задачи диссертационной работы, перечисляются методы исследований и основные положения, которые выносятся на защиту. Подчеркивается необходимость исследования и развития методов и создания инструментальных средств разработки спецификаций программно-аппаратных систем.

В нашей стране важную роль в развитии технологии программирования сыграли работы В.Л. Каткова (РИТМ), И.В.- Вельбицкого (Р-технология), Э.Х. Тыугу (ПРИЗ) и др. Из зарубежных работ широко известны PSL/PSA, SADT. Большое влияние на автора оказали работы J.Guttag и B.Liskov (CLU), а также работы В.Н. Касьянова. М.В. Трахенброта, A. Ahю, F.E. Allen и др.

В ПЕРВОМ РАЗДЕЛЕ проведен анализ современных методов проектирования программно-аппаратных систем и технологических средств

их поддержки. Сопоставлены функции существующих инструментальных средств. Выделены наиболее перспективные тенденции развития инструментальных средств проектирования программно-аппаратных систем: покрытие всего жизненного цикла разработки системы, интеграция необходимых инструментов разработки в единой среде, автоматизация всех этапов создания системы, формализованность, функциональность, объектная ориентированность, визуализация и комфортность работы пользователя с инструментальной средой.

Рассмотрены различные аспекты качества больших программно-аппаратных систем. Приоритетным предложено считать надежность. Показано, что наиболее эффективными способами достижения надежности являются:

- формализованность, функциональность и объектная ориентированность применяемых методов разработки системы;
- адекватность перехода с предшествующих этапов проектирования системы на последующие;
- автоматизированность и комфортность применения выбранных методов создания системы.

Введено понятие повторного использования результатов разработки, рассмотрены различны»! его аспекты: мобильность, заимствование процедур, библиотечных модулей, макроопределений и т.д. Отмечается, что

- наиболее эффективными методами снижения затрат на создание больших программно-аппаратных систем являются методы повторного использования;
- уровень повторного использования результатов, полученных при создании системы, зависит, в первую очередь, от применяемых при ее разработке методов проектирования и технологических средств их поддержки.

Подчеркивается особая значимость этапа разработки спецификаций, закладывающего основные свойства создаваемой системы, а также необходимость отделить этап спецификации функций системы от этапа выбора архитектуры вычислительной среды, в которой будет выполняться эта система.

На основании проведению: исследований поставлены задачи

диссертационной работы:

1. Выбрать методологию, позволяющую создавать надежные, мобильные программно-аппаратные системы, архитектурный состав исполнителей которых можно гибко пересматривать. Методология должна позволять описывать как системы, выполняющие отдельные функции, так и системы асинхронно взаимодействующих процессов сложности современных СБИС.

2. Разработать внутреннее представление формальных спецификаций программно-аппаратных систем произвольного назначения, позволяющее:

- полностью контролировать интерфейсы между частями системы;

- перенести распределение ресурсов-исполнителей на возможно более поздние этапы разработки системы;

- осуществить сквозную формализацию на всех последующих этапах разработки системы;

- автоматизировать последующие этапы разработки системы;

- создать комфортную среду автоматизированной разработки спецификаций.

3. Разработать способ укрупнения и повторного использования составных частей создаваемых спецификаций.

4. Разработать способ построения абстрактных типов данных, позволяющий обеспечить надежную проверку типов данных при высокой степени полиморфизма составных частей спецификации.

5. Разработать метод создание комфортной для разработчика среды, скрывающей жесткую дисциплину проектирования формальных спецификаций. Среда не должна допускать внесения ошибок интерфейса между частями специфицируемой системы.

6. Разработать способы повторного использования одной и той же спецификации системы для разработки множества функционально-эквивалентных систем, выполняющихся в различных по архитектурному составу вычислительных средах.

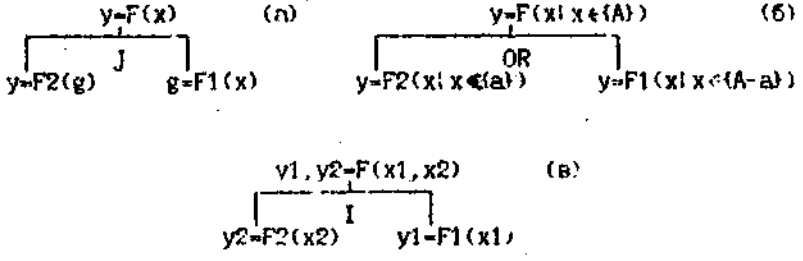
7. Реализовать на основе проведенных в диссертации исследований и разработок, рабочий макет инструментальной среды создания формальных спецификаций программно-аппаратных систем.

8. Проверить применимость рабочего макета инструментальной среды для проектирования спецификаций большой программно-аппаратной системы.

ВТОРОЙ РАЗДЕЛ целиком посвящен методам, которые предлагается использовать при разработке программно-аппаратных систем методологии формальных функциональных спецификаций, методу проверки соответствия типов данных и методам функционально-эквивалентных преобразований алгоритмов.

В качестве основного метода разработки систем выбрана методология формальных функциональных спецификаций. Ее главной посылкой является тот факт, что основная масса ошибок, совершаемых при разработке любых систем - это ошибки интерфейса. Применение методологии формальных функциональных спецификаций гарантирует отсутствие ошибок интерфейса между частями системы. Любая система в методологии формальных функциональных спецификаций рассматривается как математическая функция, преобразующая значения входных переменных в значения выходных. Алгоритм выполнения функции записывается в форме дерева ее иерархической декомпозиции. Для этого функция, а также каждая из ее подфункций, декомпозируется на подфункции с помощью базовых правил или их аналогов, применение которых гарантирует корректность интерфейсов между любыми частями дерева декомпозиции функции.Базовые правила (рис. 1) и их аналоги позволяют описы-

Базовые правила (структуры) декомпозиции.



а) Join - композиция подфункций; б) OR - разделение области определения функции между подфункциями; в) Include - разделение множества переменных функции между подфункциями.

Рис 1.

вать последовательное, параллельное или условное выполнение подфункций, а также рекурсивно обращаться к произвольному узлу дерева декомпозиции Функции. Согласно основной теореме структурного программирования этого достаточно, чтобы описать любой алгоритм. Автором диссертационной работы показывается, что благодаря простым и ясным интерфейсам между частями системы, проектируемой в методологии формальных функциональных спецификаций, дерево иерархической декомпозиции ее функции будет иметь хорошие метрологические характеристики качества.

Предложен способ контроля типов данных, развивающий методологию формальных функциональных спецификаций. Он позволяет соединить надежную проверку соответствия типов с высоким уровнем полиморфизма частей спецификации. Дается определение полиморфизма, который понимается как свойство конструкций методологии формальных функциональных спецификаций работать с переменными, типы данных которых изменяются от применения к применению. Используется алгебраический подход к построению абстрактных типов данных. В его развитие предлагается считать, что абстрактный тип данных представляет собой совокупность операций, связывающих эти операции аксиом, констант и ограничений. Суть предлагаемого метода проверки эквивалентности заключается в сопоставлении этих данных для типов переменных в развиваемом узле дерева декомпозиции функции и типов переменных в развивающей узел конструкции. Ири этом учитывается наличие совпадающих операций в типах и их присутствие в развивающей конструкции, допустимость констант в условиях разветвления развивающей конструкции, вложенность областей определения, соответствие аксиом и т.д.

Разработаны способы функционально-эквивалентных преобразований алгоритмов, записанных в методологии формальных функциональных спецификаций. Это, в частности, способы конвейеризации рекурсии итеративного типа, способы преобразования дерева декомпозиции к виду, описывающему выполнение функции на заданном наборе исполнителей, способы внесения кэширующих возможностей в имеющееся описание и т.д. Функционально-эквивалентные преобразования позволяют повысить повторное использование одной и той же формальной спецификации, создавая на ее основе множество сис-

тем, имеющих различные соотношения стоимости/производительности/эффективности и различные потребности в технических средствах, на которых они должны выполняться. С помощью функционально-эквивалентных преобразований можно формально привести последовательные алгоритмы к виду, описывающему, в частности, архитектурные особенности современных СБИС.

ТРЕТИЙ РАЗДЕЛ посвящен вопросам реализации рабочего макета инструментальной среды поддержки разработки формальных функциональных спецификаций программно-аппаратных систем.

Среду разработки спецификаций можно представить как некоторую систему программирования, которая объединяет в себе функции редактирования, трансляции, отладки и т.д., традиционные для таких систем. Интеграция этих функций в единой среде увеличивает производительность труда разработчика целевой системы.

Основой рабочего макета среды разработки спецификаций (рис. 2) является графический экранный редактор, не допускающий внесения ошибок интерфейса при описании функций системы. Эго упростило не только процесс описания системы разработчиком, но и процесс создания рабочего макета инструментальных средств, устранив, в частности, синтаксический анализ текста. Редактор позволяет пользователю декомпозировать функцию как с помощью базовых правил (структур) и операций типов данных, так и с помощью правил и операций', созданных самим пользователем.

При сохранении пользовательских структур и операций редактор автоматически, выделявляет их интерфейсную часть и создает ив нее краткую запись - синтаксис. Синтаксис имеют также базовые структуры и операции типа данных. Синтаксис служит редактору сценарием при ведении диалога с пользователем для декомпозиции функции. Осуществляемая редактором проверка корректности интерфейсов базируется на уникальности именования значений переменных и нумерации узлов. Это позволяет полностью контролировать передачу переменных между функцией и ее подфункциями. Редактор выполняет также проверку соответствия типов данных с точки зрения выполняемых над ними операций и с точки зрения соответствия областей определения. Он осуществляет все действия по контролю областей определения, которые возможно осуществить статически.

Архитектура рабочего макета инструментальной среды

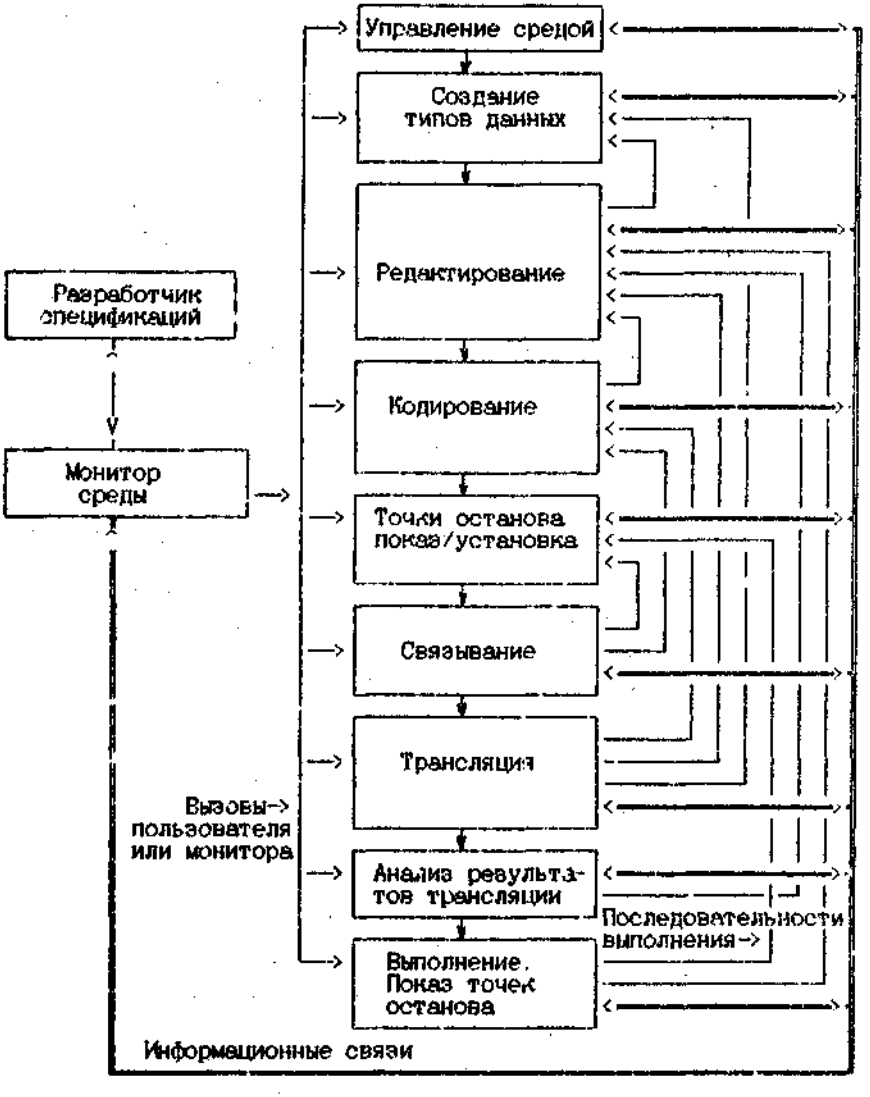


Рис. 2.

Проверка соответствия областей определения осуществляется выполнением процедур наложения отрезков, множеств и т.д. для соответствующего типа данных. Далее при выполнении описания может быть динамически проверено попадание значения выходной переменной функции в заданную область.

Описание типа данных осуществляется специальной компонентой среды (рис. 2). Спецификация операции типа данных представляется в виде ее синтаксиса. Реализацией операции типа является написанная пользователем процедура на языке программирование.

Рабочий макет среды разработки спецификаций включает в себя кодировщики, автоматически переводящие формальное описание системы в виде дерева иерархической декомпозиции на выбранные языки программирования. Создаваемый ими текст содержит информацию, позволяющую возвращаться в него при возникновении ошибок. Отладочные возможности предназначены для приведения описания системы в соответствие с необходимой разработчику логикой выполнения ее функции. Пользователь устанавливает (убирает, добавляет) точки останова с помощью отладочной компоненты среды. Поскольку генерация текста Канадой части описания осуществляется только один раз (до внесения изменений в само описание), сгенерированная процедура не содержит информации о конкретных точках останова. Эта информация вносится впоследствии путем редактирования текста нужной процедуры и подмены текста этой процедуры в тексте исполняющейся программы. Для осуществления такого подхода среда имеет связывающую компоненту, которая автоматически собирает исполнительный текст заданной пользователем части разрабатываемой системы из ранее сгенерированных процедур.

Инструментальная среда разработки спецификаций автоматизирует начальный этап проектирования программно-аппаратных систем. Она является частью более полных инструментальных средств проектирования. Среда разработки спецификаций может иметь и самостоятельную ценность. В частности, она может применяться для быстрого прототипирования или как кросс-система подготовки ПО для вычислительных систем, не имеющих технологических средств его создания.

В ЧЕТВЕРТОМ РАЗДЕЛЕ рассматриваются вопросы применения рабочего макета инструментальной среды поддержки разработки формальных функциональных спецификации к описанию поведения 32-разрядного микропроцессора Intel 8C386. Пример можно считать подходящим для апробации методологии формальных функциональных спецификаций и инструментальной среды ее поддержки.

Введено понятие моделей поведения произвольного процессора. Такие модели позволяют отладить алгоритм выполнения инструкций процессора и добиться приемлемого для разработчика совмещения во времени выполнения его функций. Показано, что отправной точкой создания моделей поведения процессора может стать формальное описание алгоритма последовательного выполнения эго инструкций в методологии формальных функциональных спецификаций.

Приведено формальное функциональное описание верхних уровней декомпозиции алгоритма последовательного выполнения инструкций процессора с системой команд Intel 80386 (рис. 3). Его рабо-

Основной цикл работы процессора

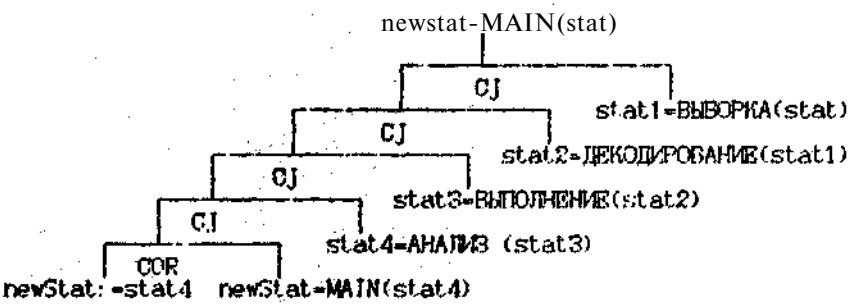


Рис. 3.

та представляется двумя взаимосвязанными циклами, выход из которых осуществляется по значению переменных, отражающих падение напряжения и сброс. Основной цикл заключается в последовательном выполнении выборки инструкции, декодировании, выполнении и анализе результатов выполнения инструкции. Переменные stat* отражают изменение внутреннего состояния процессора. Начальный цикл выполняет инициирующие функции и запускает основной цикл.

Большое внимание уделено описании взаимодействия микросхемы с внешней средой. Рассмотрено описание последовательного и конвейерного взаимодействия микросхемы с внешней средой в методологии формальных функциональных спецификаций. Для описания конвейерного взаимодействия, шинный интерфейс микросхемы рассматривается не как единый исполнитель функции взаимодействия, а как три независимых исполнителя, которые работают одновременно. Это позволяет совмещать во времени некоторые действия шинного интерфейса, описывая таким образом его конвейерную работу. Отмечается, что описание конвейерного взаимодействия шинного интерфейса с внешней средой может быть получено функционально-эквивалентным преобразованием описания их последовательного взаимодействия.

Основными факторами, которые позволили описать алгоритм последовательного выполнения инструкций микропроцессора Intel 80386 являются:

- наличие у его инструкций общих и подобных действий и структур памяти, над которыми они работают;

- повышенные возможности повторного использования, предоставляемые выбранной методологией формальных функциональных спецификаций и разработанным рабочим макетом инструментальной среды ее поддержки.

Проведен краткий анализ повторного использования разработанного описания. Оно оказалось достаточно высоким. Так, для описания инструкций условного перехода, имеющих 30 мнемоник, зависящих от условий перехода и двух размеров смещения, потребовалось написание всего одной структуры. В группе инструкций сложения повторное использование частей описания приближается к 95%, & в группе арифметических инструкций в целом к 80%. Расчет производился в байтах внутреннего представления

описания.

В ПЯТОМ РАЗДЕЛЕ проведен краткий анализ полученных в диссертационной работе результатов.

Сопоставлены функции рабочего макета инструментальной среды и инструментальных систем программирования, рассмотренных в разделе 1. Отмечается достаточно полное покрытие этих функций разработанным рабочим макетом инструментальной среды. Указывается на направленность инструментальной среды разработкой формальных функциональных спецификаций на автоматизацию наименее автоматизированных начальных этапов проектирования программно-аппаратных систем. Основными отличиями рабочего макета инструментальной среды разработки формальных функциональных спецификаций от ближайших прототипов являются:

- замена традиционного цикла: редактирование, трансляция, выполнение, исправление ошибок циклом редактирование - выполнение;

- усиленные полиморфные возможности абстракции данных и абстракции управления.

Проанализированы возможности повторного использования формального описания алгоритма последовательного выполнения инструкций микропроцессора с системой команд Intel 80386 для создания функционального описания микропроцессора с системой команд Intel 80486, который является более поздней и более мощной моделью данной серии микропроцессоров Несмотря на значительное увеличение функциональных возможностей, объем описаний, которые придется дописать, не должен превысить 44% полного описания микропроцессора с системой команд Intel 80386.

В ЗАКЛЮЧЕНИИ сформулированы основные выводы и результаты диссертационной работы.

В ПРИЛОЖЕНИИ *1* приведено внутреннее представление дерева иерархической декомпозиции функции в разработанном рабочем макете инструментальной среды.

В ПРИЛОЖЕНИИ 2 приведены акты об использовании результатов работы.

ОСНОВНЫЕ ВЫВОДЫ И РЕЗУЛЬТАТЫ РАБОТЫ

1. Показано, что в отличие от традиционных способов проектирование больших программно-аппаратных систем в методологии формальных функциональных спецификаций позволяет формализовать и автоматизировать все этапы проектирования системы, адекватно использовать результаты предыдущих этапов проектирования на последующих, перенести решение вопроса о составе технических средств, на которых должна выполняться разрабатываемая система, на более поздние этапы проектирования.

2. Разработана архитектура интегрированной инструментальной среды поддержки разработки формальных функциональных спецификаций программно-аппаратных систем и реализован рабочий макет этой среды. Для этого:

- разработан рабочий макет графического экранного редактора. Его отличительной особенностью является поддержка аксиоматики методологии формальных функциональных спецификаций. Редактор позволяет контролировать интерфейсы между частями специфицируемой системы, спрятать жесткую дисциплину методологии в комфортную для оператора среду разработки спецификаций, создавать пользовательские абстракции управления, дающие возможность укрупнять уровень описания;

- разработано внутреннее представление спецификации. Отличительной особенностью представления является его свойство быть сценарием диалога редактора с пользователем;

- методы алгебраичесчой спецификации абстрактных типов данных развиты понятием эквивалентности на иерархических деревьях формальной функциональной спецификации. Предлагаемый подход отличается от известных тем, что позволяет попылить надежность проверки соответствий типов данных, сохраняя уровень полиморфизма составных частей спецификаций создаваемой системы, превышающей соответствующие возможности полиморфных языков Ада, CLU и т.д.

Отличительной особенностью среды является зависимость ее поведения от последовательности выполнения компонент и готовности спецификаций.

3. Разработаны методы функционально-эквивалентных преобразований алгоритмов, залистанных в методологии формальных функциональных спецификаций. Разработанные методы позволяют образо-

вывать одну и ту же спецификацию конкретной системы к множеству спецификаций функционально-эквивалентных систем, выполняющихся в различных по архитектурному составу вычислительных средах. Установлено, что применение разработанных методов повышает надежность специфицируемых систем и повторное использование спецификаций.

4. С помощью рабочего макета инструментальной среды разработки спецификаций спроектированы формальные функциональные спецификации алгоритма последовательного выполнения инструкций 32-разрядного микропроцессора с системой команд Intel 80386. Его можно применить для создания ряда поведенческих моделей данного микропроцессора, а также для создания формального описания поведения следующих микропроцессоров данной серии. Тем самым установлена достаточность реализованного макета инструментальной среды для создания формальных функциональных спецификаций поведения СБИС.

ПУБЛИКАЦИИ.

1. Орловская Е.В. Методы оценки качества программ. Казань. КФ ИЛИ АН СССР. -1987. ВИНИТИ деп. N 6075-В87. -78 с.

2. Орловская В.В., Семенова Н.И. Инструментальные средства поддержки разработки программного обеспечения. КФ ИЛИ АН СССР, Казань, 1988. -81 с., -Деп. в ВИНИТИ, N 3670-ВВ8.

3. Никонов В.А., Орловская В.В., Семенова Н.И., Устюгова В.Н. Формализация и комфорт. Опыт совмещения. /Прикладные проблемы информатики. Казань. КФ АН СССР. -1989. - С. 124-129.

4. Орловская Б.В., Устюгова В.Н., Габбасова Р.С. Опыт функционального описания процессора Intel 80386. /Прикладные проблемы информатики. Казань. КНЦ АН СССР. -1990. -С. 16-36.

5. Агафонов НЮ., Орловская В. В.. Функциональное взаимодействие микросхемы с внешней средой на примере шинного интерфейса микропроцессора Intel 80386. /Прикладные проблемы информатики. Казань. КНЦ АН СССР. -1990. -С. 37-43.

6. Орловская ВВ., Штильман Л. Ф.. Функционально-эквивалентные преобразования алгоритмов. Казань. КФ ИЛИ АН СССР. -1990.

ВИНИТИ деп. N 2231-В90. -26 с.

7. Орловская В.В. Эквивалентность типов данных - ключ к повышению повторного использования программных разработок. // Программирование. -N 4, 1991. -С. 11-17.

8. Борисович Л.В., Габбасова Р.С., Орловская Е.В., Семенова Н.И., Устюгова В.Н., Цейтлин М.Р., Штильман Л.Ф. Об одном подходе к проектированию сложных программно-аппаратных систем. /Конференция-ярмарка "Технология программирования 90-х". Киев. Май 1991. -С. 150-152.

9. Орловская Е.В., Устюгова В.Н., Габбасова Р.С. Интегрированная среда разработки формальных функциональных спецификаций. /Прикладные проблемы информатики. Казань. КНЦ АН СССР. -1991. -С. 25-33.

ЛИЧНЫЙ ВКЛАД АВТОРА. Результаты, составляющие основное содержание работы, являются составной частью методов и инструментальных средств, входящих в инструментальную систему разработки больших программно-аппаратных систем. Методы функционально-эквивалентных преобразований алгоритмов разработаны автором совместно со Штильманом Л.Ф. на паритетных началах. В опубликованных работах, выполненных в соавторстве, лично Орловская Е.В. внесла следующий вклад: /2/ - разработан принцип анализа и проанализирована основная масса представленных в работе инструментальных систем; /3/ - разработан принцип создания графического редактора, ориентированного на семантику методологии формальных функциональных спецификаций; /4/ - разработан общий принцип описания логики поведения СБИС с помощью методологии формальных функциональных спецификаций, а также методы повторного использования готовых частей описания при создании следующих; /6/ - разработан общий принцип описания последовательного и параллельно-конвейерного взаимодействия микропроцессора с внешней средой; /8/, /9/ - разработана архитектура, принципы интеграции инструментальной среды и некоторые компоненты ее рабочего макета.