

КОНЦЕПЦИИ АРХИТЕКТУРЫ ЭЕМ IV-го ПОКОЛЕНИЯ

А.П.Заморин

Понятие "архитектура ЭЕМ" впервые введено Р.Бруксом (одним из создателей Системы 360/TBM) применительно к III-му поколению ЭЕМ и определено им как "полная и детальная спецификация интерфейса пользователь - ЭЕМ". Попытка распространить это понятие на предыдущие поколения ЭЕМ привела к тому, что все три поколения ЭЕМ стали отождествляться с одной единственной архитектурой ЭЕМ - архитектурой Дж.фон Неймана. Подобная постановка вряд ли является конструктивной, поскольку полностью игнорирует этапность в развитии архитектуры ЭЕМ. Определить основные черты архитектуры нового поколения - это значит прежде всего представить эволюцию ЭЕМ как поэтапный процесс реализации некоторой целевой функции, порождающий различные архитектурные новации.

Следует отметить, что предложенное определение архитектуры является концептуальным и его конкретизация зависит от характера взаимодействия пользователя с ЭЕМ. Отсюда возникают такие понятия, как архитектура системы, архитектура процессора, архитектура программного обеспечения и т.п. К тому же данное определение не содержит в явном виде важные архитектурные концепции коллективного доступа: управление параллельными процессами, разделение совместно используемых ресурсов, систему защиты и т.п. В то же время некоторые структурные аспекты можно лишь условно отнести к архитектурным концепциям: например, конвейерная обработка в процессоре. Так, следуя каноническому определению, вычислительную систему можно считать многопроцессорной лишь в случае, если в интерфейсе "пользователь-система" определена возможность спецификации параллельной работы на нескольких процессорах.

Все эти ограничения достаточно условны, как условно и само понятие архитектуры, и цель автора не в навязывании определенного мнения, а в попытке конкретизации характерных черт нового поколения ЭВМ, исходя из наиболее общего определения понятия архитектуры. В дальнейшем под пользователем будет пониматься любой потребитель ресурсов ЭВМ и соответственно будет рассматриваться интерфейс "пользователь - операционная среда" (как некоторая совокупность доступных для пользователя ресурсов), а под архитектурой вычислительной системы будет пониматься определенная среда с учетом всех допустимых спецификаций в интерфейсе пользователя.

Понятие "интерфейс" можно трактовать как форму представления проблемы, однозначно воспринимаемую общающимися сторонами. Порождение наиболее эффективной формы общения пользователя с ЭВМ и является той целью, в направлении которой эволюционирует архитектура ЭВМ. Очевидно, эффективность общения определяется трудоемкостью процесса интерпретации задачи в форму, принятую соглашениями интерфейса, что естественно достигается лишь в случае, если операционная среда адекватна среде применения (т.е. модели алгоритма решаемой задачи). Но вся проблематика интерфейса и заключена в значительном "семантическом разрыве" между внешней моделью задачи и внутренней моделью операционной среды. К тому же модель алгоритма динамична и отражает специфику проблемной области (виды объектов, процедуры обработки и т.п.), тогда как операционная среда, наоборот, консервативна и отражает достигнутый уровень технологии.

В общем случае решение задачи можно представить как многоэтапный процесс: алгоритмизация, программирование, планирование, управление, трансляция, интерпретация. И каждый этап требует выхода на интерфейс определенного уровня. Задача развития вычислительных средств состоит в повышении уровня автоматизации процесса решения задач, следовательно, основная закономерность развития архитектуры ЭВМ заключается в повышении уровня интерфейса взаимодействия ЭВМ с пользователем.

Так, операционная среда ЭВМ I-го поколения - это конкретные аппаратные механизмы реальной ЭВМ, характер и последовательность операций которых заданы пользователем. Соответственно в данной среде реализован самый низкий уровень интерфейса - уровень аппаратной интерпретации.

Операционная среда II-го поколения - это процедурная ЭВМ, воспринимающая описание задачи пользователя в некотором алгоритмиче-

ском языке. Соответственно в интерфейсе появляются такие понятия, как вызов процедуры, набор данных и другие. Уровень интерфейса поднимается на одну ступень выше. Однако пользователь планирует работы с учетом реальных ресурсов и управляет непосредственно процессом решения на реальной ЭВМ.

Наконец, операционная среда ЭВМ 3-го поколения - это множество независимо определенных процедурных ЭВМ, реализуемых одной реальной машиной. Соответственно возможно выделение в данной среде как отдельной процедурной машины, так и ее последующее усечение до машины фон Неймана, что собственно и дает основание для ее отнесения к ЭВМ данной архитектуры. В интерфейс пользователя включены средства спецификации машины, работ, наборов данных и т.п. Функции планирования работ и управления берет на себя ЭВМ. Следует отметить, что процедурная машина пользователя определяется в терминах реальной ЭВМ (процессор, канал, внешнее устройство и т.п.), что делает интерфейс по-прежнему ориентированным на систему с определенной архитектурой.

Что касается дальнейшего развития вычислительной техники, то обеспечение высокой реальной эффективности во всем диапазоне применения вычислительных средств остается, как и ранее, актуальной задачей. При этом имеются в виду все компоненты вычислительного процесса: от постановки задачи и до вывода результатов. В рамках развития архитектуры ЭВМ 3-го поколения существует несколько независимых направлений работ, преследующих достижение упомянутой цели.

I. Развитие средств традиционного (поэтапного) преобразования модели алгоритма в объекты операционной среды с использованием средств программирования, трансляторов и аппаратных средств интерпретации. Последнее достижение в данном направлении - виртуальная операционная среда (виртуальная машина), что исключает из интерфейса необходимость учета реальных ресурсов операционной среды (конфигурации ЭВМ) [1]. При этом естественны потери на трансляцию, хранение нескольких текстов программ, эмуляцию виртуальной машины в реальную операционную среду, распределение ресурсов, контроль и управление вычислительным процессом. Чем выше уровень исходного представления задачи и менее развита операционная среда, тем более ощутимы эти задержки. Эффективность в данном случае может быть повышена в основном за счет ускорения этапов трансляции и выполнения, что достигается, в основном, повышением производительности ЭВМ.

2. Исходное преобразование модели алгоритма в соответствии с архитектурой операционной среды, особенно в случае, если среда обладает рядом структурных особенностей, повышающих эффективность процесса обработки. В данном случае за счет определенных потерь на преобразование достигается высокая эффективность этапов компиляции и выполнения. Естественно ожидать, что класс задач, допускающих эффективное преобразование в операционную среду определенного типа ограничен, либо обусловлен дополнительными требованиями, например, высоким уровнем мультипрограммирования в случае архитектуры с управлением потоком данных [2]. К тому же, как показывает опыт, реализация данного подхода требует, как правило, разработки новых средств программирования.

3. Исходное преобразование операционной среды под заданную модель алгоритма. Данный подход развивается в ряде направлений: рекурсивные машины [3], вычислительные системы с программируемой структурой [4], ЭВМ с динамической архитектурой [5]. И хотя в данном случае затраты на преобразование модели алгоритма могут быть сведены к минимуму, источником снижения эффективности являются дополнительные ресурсы, необходимые для динамической перестройки операционной среды под требуемую модель. Следует отметить архитектуру В-6700 как пример статического выбора операционной среды под заданную модель алгоритма. И хотя ресурсы на перестройку здесь сведены к нулю, проблематичной остается задача эффективного отображения различных моделей алгоритмов на операционную среду, семантически ориентированную на определенную алгоритмическую модель.

4. Создание универсальной операционной среды высокого уровня как некоторого промежуточного представления, позволяющего свести к минимуму потери на преобразование, а также реализовать наиболее эффективную схему интерпретации. Работы в данном направлении привели к появлению ряда архитектур нового типа: функционально-ориентированной архитектуры [6], объектно-ориентированной архитектуры [7] и других.

Концепции функциональной архитектуры предусматривают функциональные объединения аппаратных и программных компонент вычислительной системы в функциональные модули. Основные преимущества подобной архитектуры состоят в упрощении проектирования систем со сложной конфигурацией, в возможности выбора способов реализации функций (аппаратного, микропрограммного, программного). В интерфейсе пользователя определяются только требуемые функции управления и обработки.

Практическое подтверждение получила объектная архитектура, воплощенная в Системе ЗВ IBM [7], в микропроцессоре IAPX432 [8] и в ряде других систем. Основные концепции объектно-ориентированной архитектуры сводятся к использованию объектного машинного языка высокого уровня, что позволяет отобразить множество моделей алгоритмов в абстрактные объекты определенного вида, размещаемые в практически неограниченном пространстве памяти одного уровня, использовать встроенные средства поддержки баз данных, управление процессами, логическое определение средств ввода-вывода.

Объект представляет собой функционально законченный элемент задачи. Он может являться эквивалентом программы, набора данных, описания набора данных, рабочей области, очереди и т.п. Объект замкнут: он может содержать как функциональную часть, так и собственную рабочую область. Объект имеет собственное адресное пространство. Обращение к объекту осуществляется по уникальному имени, которое преобразуется в ссылку с использованием объекта типа "контекст". На уровне объекта может быть задана и проведена любая форма защиты доступа. Количество объектов практически не ограничено.

В результате традиционные функции управляющих программ оказываются перенесенными на уровень интерпретации объектного представления, что резко уменьшает объем системного программного обеспечения, повышает быстродействие и надежность управления. Это соответствует тенденциям снижения объема программного обеспечения за счет увеличения объема аппаратных средств, обеспечивает повышение эффективности программирования и надежности программного обеспечения, создает базу дальнейшего развития как в части внедрения новых технологий, так и реализации новых функций. В то же время объектное представление допускает различные формы интерпретации (программную, микропрограммную, аппаратную).

Именно данное (четвертое) направление в наибольшей степени характеризует архитектуру нового поколения. Для данной архитектуры характерно существенное повышение уровня представления операционной среды и, следовательно, интерфейса пользователя, что значительно повышает эффективность на начальных этапах постановки задачи (алгоритмизация и программирование). Исключение из интерфейса пользователя спецификации ресурсов, ввиду погружения в операционную среду функций планирования, управления и обработки, делает программирование независимым от принципов реализации ЭВМ.

Можно утверждать, что объектно-ориентированная архитектура представляет собой качественно новое поколение архитектуры ЭВМ.

Исходя из этого, можно сформулировать концепции ЭВМ IV-го поколения следующим образом:

1) операционная среда имеет проблемный характер, т.е. определяется в концептуальных понятиях решаемой проблемы и отображаемых процессов: задача представляется в виде совокупности объектов и отношений между ними;

2) высокоуровневый объектный язык использует единое представление для всех объектов задачи (программы, процесса, базы данных) и набор функций высокого уровня для управления и обработки объек-

3) объектная память имеет один уровень адресации с практически неограниченным адресным пространством с уникальными идентификаторами для всех объектов задачи, что исключает из интерфейса пользователя понятия механизмов доступа к данным и управления памятью;

4) независимость объектного представления от принципов технической реализации и соответственно интеграция в аппаратуру средств управления ресурсами физического уровня, что исключает из интерфейса пользователя понятие системных ресурсов; ограничение в спецификации работ и данных;

5) логическое определение в объектном языке средств ввода-вывода, сети передачи данных и других средств общения, что позволяет развивать систему с сохранением уровня совместимости;

6) включение в объектный язык таких средств поддержки систем программирования, как управление базой данных, контроль доступа, управление параллельными процессами и др., что значительно повышает эффективность и надежность программирования;

7) интегрирование в аппаратуру средств управления физического уровня, включая традиционные функции управляющей программы, что существенно снижает объем операционной системы, повышает эффективность управления.

Все это достаточно хорошо согласуется с ранее данными определениями архитектуры ЭВМ IV-го поколения [9, 10], но содержит дополнительную конкретизацию этого понятия, логически вытекающую из приведенного анализа. Следует отметить, что практически для всех направлений создания ЭВМ IV-го поколения характерна низкая эффективность при использовании традиционных систем программирования,

что объясняется повышенным уровнем операционной среды, который требует более мощных языковых средств.

Л и т е р а т у р а

1. ПРЖИЯЛКОВСКИЙ В.В., ЛОМОВ Д.С. Технические и программные средства Единой системы ЭЕМ. -М.: Статистика, 1980. - 227 с.
2. БРИТОН Д. Новая компьютерная архитектура, основанная на потоке данных. -Электроника, №9, 1979, с. 88-90.
3. ГЛУШКОВ В.М. Рекурсивные машины и вычислительная техника. -Киев, 1974. - 42 с. (Препринт/ИК АН УССР: 57).
4. КОРНЕЕВ В.В., ХОРОШЕВСКИЙ В.Г. Вычислительные системы с программируемой с.руктурой. -Электронное моделирование, №1, 1981, с. 42-52.
5. ТОРГАШЕВ В.А. ЭЕМ с динамической архитектурой. -Л., 1980, -34 с. (Препринт/Ленингр. науч.-исслед.вычислит.центр: 32).
6. Распределение функций в современных вычислительных системах. -Экспресс-информ. Вычислительная техника, №18, 1982, с.6-9.
7. ДЕРНАК А. Система 38 - показатель стремления фирмы УВМ идти в ногу со временем. -Электроника, № 6, 1979, с. 26-43.
8. Описание микропроцессорной системы APX 432 фирмы Интел. -Экспресс-информ. Вычислительная техника, №48, 1981, с.6-11.
9. БАБАЯН Б.А. Структуры современных ЭЕМ. -М., 1975. - 15 с. (Препринт/ИТМ и ВТ АН СССР: 4).
10. БАЛАШОВ Е.П., ЧАСТИКОВ А.П. Эволюция вычислительных систем. -М.: Знание, 1981. - 236 с.

Поступила в ред.-изд.отд.
7 июня 1983 года