

УДК 004.942

ВЕРИФИКАЦИЯ ЗАДЕРЖКИ В МИКРОАРХИТЕКТУРНЫХ МОДЕЛЯХ КОММУНИКАЦИОННЫХ ФАБРИК

Ю. О. Викторов,

инженер-практикант

А. Н. Готманов,

инженер-исследователь

Intel Corporation, г. Москва

Изложены основы метода анализа задержек в микроархитектурных моделях коммуникационных фабрик. На простых примерах показано, как получить верхнюю оценку задержки и верифицировать ее с помощью ранжирующих функций. Для моделей со структурными циклами предложен метод анализа, основанный на раздельном рассмотрении достижимых состояний системы.

Ключевые слова — качество обслуживания, формальная верификация, xMAS, система-на-кристалле, коммуникационная фабрика, ранжирующая функция, задержка передачи данных.

Введение

Современные системы-на-кристалле требуют эффективной организации обмена данными. Подсистему, отвечающую за пересылку сообщений и пакетов данных между устройствами, называют *коммуникационной фабрикой* (communication fabric) [1]. Коммуникационные фабрики обычно обладают сложной распределенной архитектурой с высоким уровнем параллелизма и конвейеризации для одновременной обработки множества транзакций, находящихся на разных стадиях исполнения.

В условиях ограниченных аппаратных ресурсов (размеры очередей, разрядности шин и т. д.) производительность работы системы связана с эффективностью их разделения конкурирующими процессами. Качество архитектуры коммуникационных фабрик напрямую зависит от умения оценивать на этапе проектирования минимальный уровень обслуживания, который система гарантирует своим агентам.

В данной статье рассматривается *задержка передачи* (latency) — интервал времени между отправлением данных источником и их доставкой в точку назначения. Имитационное моделирование дает возможность составить хорошее представление о средней производительности системы. Однако поведение системы в наихудшем случае может сильно отличаться от среднего. Надежную оценку производительности мож-

но получить с помощью формальной верификации. Для этого сначала необходимо построить формальную модель системы. Язык xMAS [2] позволяет создавать наглядные модели коммуникационных фабрик путем композиции примитивов (стандартных блоков) и может использоваться для моделирования микроархитектурных задержек.

Ограничение задержки («пакет попадает из точки A в точку B не более чем за N тактов») легко сформулировать в виде утверждения линейной темпоральной логики (Linear Temporal Logic — LTL) [3]. Однако его проверка часто оказывается за пределами возможностей современных алгоритмов верификации моделей (интерполяции [4], k -шаговой индукции [5], PDR [6]), так как требует рассмотрения порядка N последовательных тактов работы системы.

Предлагается альтернативный подход, основанный на методах статического анализа и ранжирующих функций. Начав с простой формализации понятия задержки в рамках LTL, мы покажем, как воспользоваться элементарными свойствами задержки для анализа времени выполнения многошаговых транзакций. Построение оценки на задержку представлено как вывод из гипотез в формальной теории и сопровождается параллельным построением ранжирующей функции.

Ранжирующая функция — это неотрицательная целочисленная функция состояния [7], оценивающая число тактов, оставшееся до за-

вершения транзакции. Для доказательства оценки задержки в N тактов достаточно найти соответствующую ранжирующую функцию R и показать, что:

- 1) R убывает на каждом такте;
- 2) глобальный максимум R не превосходит N .

Доказательство утверждений 1 и 2 проводится методом k -шаговой индукции при небольших значениях k , не зависящих от N . С этой задачей легко справляется большинство современных средств верификации моделей.

Таким образом, предлагаемый метод одновременно решает задачи отыскания задержки и ее верификации. В статье изложены теоретические основы метода и представлены первые результаты экспериментов на моделях небольшого размера.

Язык моделирования

xMAS — это язык формального моделирования микроархитектуры [2]. Модели xMAS строятся из небольшого набора стандартных блоков (примитивов), соединенных каналами для передачи пакетов с данными. Каждый канал соединяет ровно два примитива, один из которых является инициатором передачи, а другой — ее получателем. Все примитивы модели работают синхронно по одному тактовому сигналу. В данной статье используется базовый набор примитивов, дополненный специализированными примитивами для моделирования качества обслуживания (рис. 1).

Исток (source) порождает пакеты с заданным значением e . В зависимости от типа истока очередной пакет может появляться на каждом такте (*eager*), с недетерминированной задержкой (*non-det*) или не появляться никогда (*dead*).

В процессе обработки пакеты хранятся в *очередях* (queue) конечного размера. Очередь реализует дисциплину FIFO. Новый пакет может быть помещен в очередь только при наличии в ней свободного места. Задержка прохождения пакета через пустую очередь равна одному такту.

Примитив *преобразования* (func) изменяет значение пакета, используя функцию f . Пакет на выходе *барьера* (join) появляется только при наличии пакета на каждом из его входов, при этом выходной пакет вычисляется функцией h от вход-

ных пакетов. Проходя через *форк* (fork), входной пакет преобразуется в два выходных пакета с помощью функций f и g . Примитив *ветвления* (switch) вычисляет предикат s от входного пакета и в зависимости от его значения перенаправляет пакет на один из двух выходов. При *слиянии* (merge) входные пакеты передаются на выход без изменений. Если два пакета поступают на входы слияния одновременно, посылается один из них, а второй задерживается. Для выбора между пакетами используется справедливый алгоритм арбитража. Допускается слияние с произвольным числом входных каналов.

Сток (sink) поглощает пакеты, завершая их обработку в модели. Аналогично истоку, существуют три разновидности стока (*eager*, *non-det*, *dead*).

На рис. 2 показаны примеры моделей xMAS, использующих как базовые примитивы, так и до-

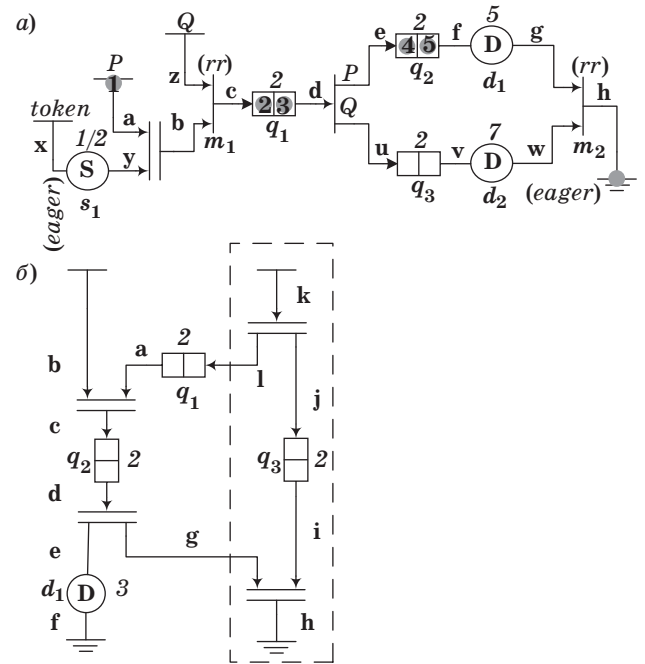


Рис. 2. Модели xMAS: а — модель коммуникационной фабрики; б — модель канала с кредитной логикой¹

¹ Имена примитивов обозначены полужирным курсивом (q_1), имена каналов — прямым полужирным (u). Светлым курсивом указаны параметры примитивов, такие как тип истока (*eager*) и величина задержки (5). Полужирные цифры (1–5) обозначают позиции, занимаемые пакетом P на его траектории движения.

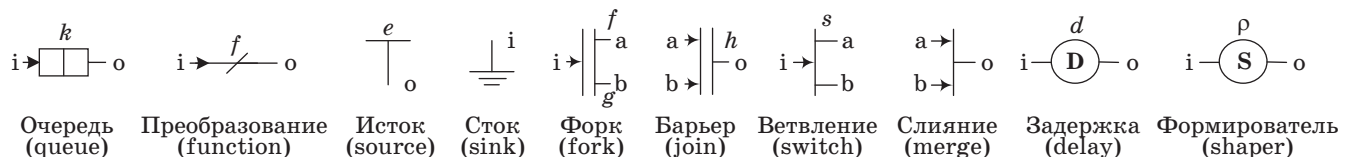


Рис. 1. Примитивы xMAS

полнительные примитивы качества обслуживания (см. рис. 1), последние подробно описаны ниже.

Протокол передачи данных на канале u использует два логических управляющих сигнала: $u.irdy$ (от initiator ready) и $u.trdy$ (от target ready). Сигнал $u.irdy$ равен True, если инициатор готов передать один пакет с данными. Аналогично, сигнал $u.trdy$ равен True, если получатель готов принять один пакет с данными. Передача пакета происходит при $u.xfer = u.irdy \cdot u.trdy = \text{True}$. Значение пакета задается сигналом $u.data$ и может быть любого конечного типа.

Все примитивы xMAS обладают простой синхронной семантикой, которая может быть выражена на языке битовых уравнений, связывающих значения сигналов $irdy$, $trdy$ и $data$ на примыкающих к примитиву каналах. Например, примитив форк с входом i и выходами a , b , параметризованный функциями f и g , описывается соотношениями

$$\begin{aligned} a.data &:= f(i.data); \\ b.data &:= g(i.data); \\ a.irdy &:= i.irdy \cdot b.trdy; \\ b.irdy &:= i.irdy \cdot a.trdy; \\ i.trdy &:= a.trdy \cdot b.trdy. \end{aligned}$$

Такое определение гарантирует, что передачи на каналах i , a , b происходят одновременно.

Устойчивость

Протокол xMAS обладает устойчивостью [8], т. е. управляющие сигналы $irdy$ и $trdy$ не могут менять значение с True на False до завершения передачи на канале. Полезно определить устойчивость в более общей форме для двух произвольных событий A и B :

$$A \triangleright B \equiv \mathbf{G}(A \cdot \neg B \rightarrow \mathbf{X}A).$$

Запись читается как « A устойчиво по отношению к B ». Мы пользуемся стандартными обозначениями LTL [3]: $\mathbf{G}A$ — «всегда A », $\mathbf{F}A$ — «однажды A », $\mathbf{X}A$ — «в следующий момент времени A », $\mathbf{Y}A$ — «в предыдущий момент времени A ». Устойчивость протокола передачи данных на канале u может быть выражена соотношениями

$$\begin{aligned} u.irdy &\triangleright u.trdy; \\ u.trdy &\triangleright u.irdy; \\ \forall x : (u = x &\triangleright u.trdy), \end{aligned}$$

где $u = x$ используется как сокращение для $u.irdy \cdot (u.data = x)$, а квантификация происходит по всем возможным значениям сигнала $u.data$.

Моделирование задержек

Для анализа задержек необходимо расширить базовый набор примитивов xMAS. Для моделирования задержек обработки введем новый примитив *ограниченной задержки* (delay). Поведение задержки величиной d с входом i и выходом o задается уравнениями

$$\begin{aligned} i.trdy &:= o.trdy \cdot (\text{cnt} = 0); \\ o.irdy &:= i.irdy \cdot (\text{cnt} = 0); \\ \text{next}(\text{cnt}) &:= \begin{cases} d, & i.irdy \cdot i.trdy; \\ \text{cnt} - 1, & i.irdy \cdot \neg i.trdy \cdot (\text{cnt} > 0); \\ \text{cnt}, & \neg i.irdy + \neg i.trdy \cdot (\text{cnt} = 0); \end{cases} \\ \text{init}(\text{cnt}) &:= d. \end{aligned}$$

Используемый в уравнениях оператор $\text{next}(x)$ определяет значение регистровой переменной x на следующем такте, $\text{init}(x)$ — в начальный момент времени. Целочисленная переменная cnt принимает значения в $\{0..d\}$ и задает число тактов ожидания до передачи. Случайная задержка с верхней границей d моделируется аналогично.

Другой тип задержек создается *формирователем потока* (shaper). Формирователь гарантирует, что пакеты на выходе появляются не чаще, чем с заданной частотой ρ , задерживая при необходимости входные пакеты. Пусть $\rho = R/N$ для целых положительных чисел R и N . Тогда поведение примитива можно описать уравнениями

$$\begin{aligned} i.trdy &:= o.trdy \cdot (\text{bkt} \geq N - R); \\ o.irdy &:= i.irdy \cdot (\text{bkt} \geq N - R); \\ \text{next}(\text{bkt}) &:= \begin{cases} \text{bkt} - N + R, & i.xfer; \\ \min\{\text{bkt} + R, N\}, & \neg i.xfer; \end{cases} \\ \text{init}(\text{bkt}) &:= N. \end{aligned}$$

Формирователь работает по принципу «leaky bucket», увеличивая целочисленную переменную bkt на R на каждом такте. Входящий пакет задерживается, если текущее значение bkt меньше, чем $N - R$. При передаче пакета переменная bkt уменьшается на N .

Модель коммуникационной фабрики

Рассмотрим подробнее пример модели на рис. 2, *a*. Пакет со значением P порождается источником и ожидает срабатывания барьера на канале a . На второй вход барьера поступают управляющие пакеты (*токены*) от формирователя с частотой $1/2$. Это гарантирует, что на канале b за два такта работы появляется не более одного нового пакета. Далее пакет P проходит слияние, конкурируя с потоком пакетов Q , и, выиграв ар-

битраж, попадает в очередь по каналу c . Заметим, что пакеты Q порождаются недетерминированным истоком, и частота их появления может быть произвольной. Покидая первую очередь по каналу d , пакеты P и Q проходят ветвление и перераспределяются по каналам e и u в различные очереди, на выходе которых пакеты задерживаются. Пакеты покидают модель по каналу h после повторного слияния потоков.

Содержательно модель на рис. 2, *a* можно интерпретировать как простейшую коммуникационную фабрику, принимающую запросы от двух агентов (P и Q). Пакеты принимаются фабрикой по одному за такт, что моделируется слиянием на входе. Обработка запросов конвейеризирована. На первой ступени запросы обрабатываются последовательно и хранятся в общей очереди. На второй ступени запросы P и Q обрабатываются параллельно с различными задержками (например, фабрика может перенаправлять запросы различным устройствам для обработки). Перед удалением из фабрики запросы проходят завершающую фазу по одному за такт, что соответствует слиянию на выходе. Общую задержку обработки запроса P коммуникационной фабрикой естественно определить как количество тактов с момента появления пакета на канале a до момента его передачи на канале h .

Модель канала с кредитной логикой

Рассмотрим подробнее пример модели на рис. 2, *b*. Пакет на канале b ожидает срабатывания барьера, которое возможно лишь в том случае, если в очереди q_1 находится кредит, порождаемый подсхемой кредитной логики (выделена пунктиром).

Новый кредит выделяется при срабатывании форка с входом k . При этом одновременно создаются две копии кредита. Одна копия помещается в очередь q_1 и оттуда резервируется пакетами, ожидающими на канале b . Другая копия помещается в очередь q_3 и используется для учета числа выданных и еще не возвращенных кредитов. Когда очередь q_3 заполнена, выделение новых кредитов приостанавливается. Получив кредит, пакет попадает в очередь q_2 , при выходе из которой выделенный кредит освобождается (число «занятых» кредитов в очереди q_3 уменьшается на 1).

В любом достижимом состоянии модели число кредитов в очереди q_1 в сумме с числом пакетов в очереди q_2 равно общему числу выделенных кредитов, хранящихся в очереди q_3 . Корректная реализация схемы с кредитной логикой требует, чтобы число доступных кредитов не превосходило емкости канала, т. е. размер очереди q_3 должен быть не больше размера очереди q_2 .

Задержка

Отношение отклика

Неформально задержку можно определить как количество тактов между предусловием A и постусловием B , где A и B — произвольные события, связанные отношением отклика:

$$A \mapsto B \equiv G(A \rightarrow FB).$$

Иначе говоря, за событием A всегда следует событие B . Определим локальную задержку $\text{lat}[B]$, равную числу тактов между текущим моментом времени и следующим появлением события B :

$$\text{lat}[B] \leq k \equiv F^k B,$$

где $F^k B = B + XB + \dots + X^k B$, а X^k соответствует k последовательным применениям темпорального оператора X . Если событие B никогда не наступит, т. е. выполнено $G\neg B$, задержку $\text{lat}[B]$ будем считать равной ∞ . С каждым отношением отклика $A \mapsto B$ можно связать максимальную задержку за все время выполнения:

$$\text{Lat}[A \mapsto B] \leq k \equiv G(A \rightarrow (\text{lat}[B] \leq k)).$$

Если событие A не наступает ни разу, удобно считать, что $\text{Lat}[A \mapsto B] = 0$.

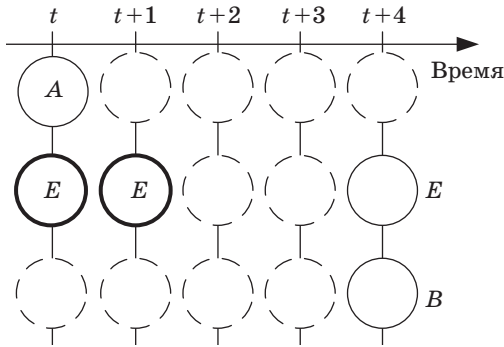
Естественно представить задержку обработки запроса P в модели на рис. 2, *a* как

$$\text{Lat}[(a = P) \mapsto h.xfer \cdot (h = P)],$$

т. е. как максимальное число тактов между появлением пакета P на канале a и передачей пакета P на канале h . Однако такое определение не вполне корректно, так как события $a = P$ и $h.xfer \cdot (h = P)$ могут быть вызваны разными пакетами, имеющими одно и то же значение P . Иными словами, модель на рис. 2, не позволяет выделить один пакет из общего потока и измерить задержку его обработки. Проблема решается простым изменением модели. Допустим, что исток, посылающий пакеты P , может присвоить одному из них особую пометку, превратив его в пакет P^* . Помеченный пакет выбирается случайно и обрабатывается моделью так же, как и обычный пакет P , сохраняя при этом пометку. Искомая задержка для помеченного пакета может быть корректно выражена как

$$\text{Lat}[(a = P^*) \mapsto h.xfer \cdot (h = P^*)].$$

По построению, модель в каждый момент времени содержит не более одного экземпляра P^* , позволяя однозначно идентифицировать пакет. Пометка присваивается случайно, поэтому каждый пакет P превращается в P^* в одном из испол-



■ Рис. 3. Задержка

нений модели. Следовательно, максимальная задержка запроса P^* (по всем исполнениям модели) совпадает с максимальной задержкой запроса P . Описанный прием добавления пометки применим к произвольной модели xMAS.

В дальнейшем нам также потребуется обобщенная форма отношения отклика и связанная с ним задержка:

$$A \mapsto_E B \equiv \mathbf{G}(A \cdot \mathbf{G}FE \rightarrow \mathbf{F}B).$$

Событие B следует за A , только если в ходе ожидания периодически наступает событие E , называемое модулем. Локальная задержка $\text{lat}_E[B]$ определяется как число наступлений события E до ближайшего наступления события B . Отличие между обычной задержкой и задержкой по модулю поясняет рис. 3. В момент времени t $\text{lat}[B] = 4$, $\text{lat}_E[B] = 2$ (последнее событие E не учитывается при подсчете задержки). $\text{Lat}[A \mapsto_E B]$ определяется аналогично $\text{Lat}[A \mapsto B]$, как максимум $\text{lat}_E[B]$ на всем исполнении. Два типа задержки связаны простым соотношением

$$\text{Lat}[A \mapsto B] = \text{Lat}[A \mapsto_{\text{True}} B].$$

Вычисление задержки

Задержка для сложного отношения отклика может быть сведена к вычислению более простых задержек. Например, пусть известно, что для некоторых событий A, B, C

$$\text{Lat}[A \mapsto B] \leq k; \text{Lat}[B \mapsto C] \leq l.$$

Тогда верно и то, что между появлением события A и последующим появлением события C проходит не более $k + l$ тактов, т. е.

$$\text{Lat}[A \mapsto C] \leq k + l.$$

Найденные закономерности можно сформулировать в виде синтаксических правил вывода для задержек:

$$\frac{}{\text{Lat}[A \mapsto A] = 0} (\text{Ref-Pre});$$

$$\frac{\text{Lat}[A \mapsto_E B] \leq k}{\text{Lat}[A \cdot C \mapsto_E B] \leq k} (\text{Pre-And});$$

$$B \triangleright C, C \triangleright B,$$

$$\text{Lat}[A \mapsto_E B] \leq l,$$

$$\text{Lat}[A \mapsto_E C] \leq k$$

$$\frac{}{\text{Lat}[A \mapsto_E B \cdot C] \leq \max\{l, k\}} (\text{Post-And});$$

$$\frac{\text{Lat}[A \mapsto_E B] \leq l}{\text{Lat}[A \cdot_{E \cdot F} B] \leq l} (\text{Mod-And});$$

$$A \triangleright B,$$

$$\frac{\text{Lat}[A \mapsto_E B] \leq l}{\text{Lat}[A \mapsto_E A \cdot B] \leq l} (\text{Persist});$$

$$\text{Lat}[A \mapsto_E B] \leq l,$$

$$\frac{\text{Lat}[B \mapsto_E C] \leq k}{\text{Lat}[A \mapsto_E C] \leq l + k} (\text{Seq});$$

$$\frac{}{\text{Lat}[\mathbf{X}A \mapsto A] \leq 1} (\text{Next});$$

$$\frac{}{\text{Lat}[\text{True} \mapsto_E E] = 0} (\text{Ref-Mod});$$

$$\text{Lat}[A \mapsto_E C] \leq l,$$

$$\text{Lat}[B \mapsto_E C] \leq k$$

$$\frac{}{\text{Lat}[A + B \mapsto_E C] \leq \max\{l, k\}} (\text{Pre-Or});$$

$$\frac{\text{Lat}[A \mapsto_E B] \leq l}{\text{Lat}[A \mapsto_E B + C] \leq l} (\text{Post-Or});$$

$$\text{Lat}[A \mapsto_E B] \leq l,$$

$$\frac{\text{Lat}[A \mapsto_F B] \leq k}{\text{Lat}[A \mapsto_{E+F} B] \leq k + l} (\text{Mod-Or});$$

$$\text{Lat}[A \mapsto_E B] \leq l,$$

$$\frac{\text{Lat}[A \mapsto_E B] \leq k}{\text{Lat}[A \mapsto_E B] \leq \min\{l, k\}} (\text{Min});$$

$$A \triangleright B, B \triangleright E,$$

$$\text{Lat}[A \mapsto_E B] \leq l,$$

$$\text{Lat}[A \mapsto E] \leq k$$

$$\frac{}{\text{Lat}[A \mapsto B \cdot E] \leq l + k + k \cdot l} (\text{Iter});$$

$$\frac{\text{Lat}[A \mapsto B] \leq k}{\text{Lat}[\mathbf{Y}(A \cdot \neg B) \mapsto B] \leq k - 1} (\text{Subseq}).$$

Теорема 1. Если формула Φ выводима из гипотез Γ с помощью правил вывода, то $\Gamma \vdash_{\text{LTL}} \Phi$.

Пользуясь правилами вывода, можно показать, что в модели на рис. 2, a

$$\text{Lat}[(a = P^*) \mapsto \text{h.xfer} \cdot (\text{h} = P^*)] \leq 50.$$

■ Таблица 1. Положения пакета P^* в модели на рис. 2, а

Номер	Положение	Канал
1	Выход истока	a
2	Хвост очереди q_1	d
3	Голова очереди q_1	d
4	Хвост очереди q_2	f
5	Голова очереди q_2	f

На пути от канала a к каналу h пакет P^* проходит положения 1–5 (см. рис. 2, а). С каждым положением связан выходной канал соответствующего истока или очереди (табл. 1).

Полное время обработки пакета P определяется временем его пребывания в каждом из положений 1–5. Используя правила (*Seq*) и (*Next*), получаем оценку

$$\begin{aligned} & \text{Lat}[(a = P^*) \mapsto h.xfer \cdot (h = P^*)] \leq \\ & \leq \text{Lat}[a.irdy \mapsto a.trdy] + 2 \cdot \text{Lat}[d.irdy \mapsto d.trdy] + \\ & + 2 \cdot \text{Lat}[f.irdy \mapsto f.trdy] + 4. \end{aligned} \quad (1)$$

Задержки передачи пакета на канале (такие как $\text{Lat}[a.irdy \mapsto a.trdy]$) будем называть локальными задержками.

Вывод локальных задержек схематично показан на рис. 4. В скобках указаны используемые правила распространения задержек для примитивов xMAS. Сами правила выглядят следующим образом. Для:

— задержки величины d с входом i и выходом o

$$\frac{\text{Lat}[o.irdy \mapsto o.trdy] \leq k}{\text{Lat}[i.irdy \mapsto i.trdy] \leq k + d} \text{ (Delay-Bwd);}$$

— очереди емкости ≥ 2 с входом i и выходом o

$$\frac{\text{Lat}[o.irdy \mapsto o.trdy] \leq k}{\text{Lat}[i.irdy \mapsto i.trdy] \leq k} \text{ (Queue-Bwd);}$$

— ветвления с входом i , выходом a и соответствующим условием $s_a(x)$

$$\frac{\text{Lat}[a.irdy \mapsto a.trdy] \leq k}{\text{Lat}[i.irdy \cdot s_a(i.data) \mapsto i.trdy] \leq k} \text{ (Switch-Bwd);}$$

— формирователя с входом i , выходом o и $\rho = R/N$

$$\frac{\text{Lat}[i.irdy] \leq k}{\text{Lat}[i \mapsto o.irdy] \leq \max\{k, N/R\}} \text{ (Shaper-Fwd);}$$

— слияния с входом a и выходом o , L_a — задержка арбитража для входа a

$$\frac{\text{Lat}[o.irdy \mapsto o.trdy] \leq k}{\text{Lat}[a.irdy \mapsto a.trdy] \leq k + L_a + k \cdot L_a} \text{ (Merge-Bwd);}$$

— барьера с входами a, b и выходом o

$$\begin{aligned} & \text{Lat}[i \mapsto b.irdy] \leq k, \\ & \frac{\text{Lat}[o.irdy \mapsto o.trdy] \leq l}{\text{Lat}[a.irdy \mapsto a.trdy] \leq k + l} \text{ (Join-Bwd).} \end{aligned}$$

Оценки вычисляются последовательно в процессе обхода модели, начиная с каналов h и x , по направлению к каналу a . Привести вывод целиком не представляется возможным из-за его объема. Рассмотрим в качестве примера вычисление задержки на канале f . Пусть уже известно, что

$$\text{Lat}[g.irdy \mapsto g.trdy] \leq 1. \quad (2)$$

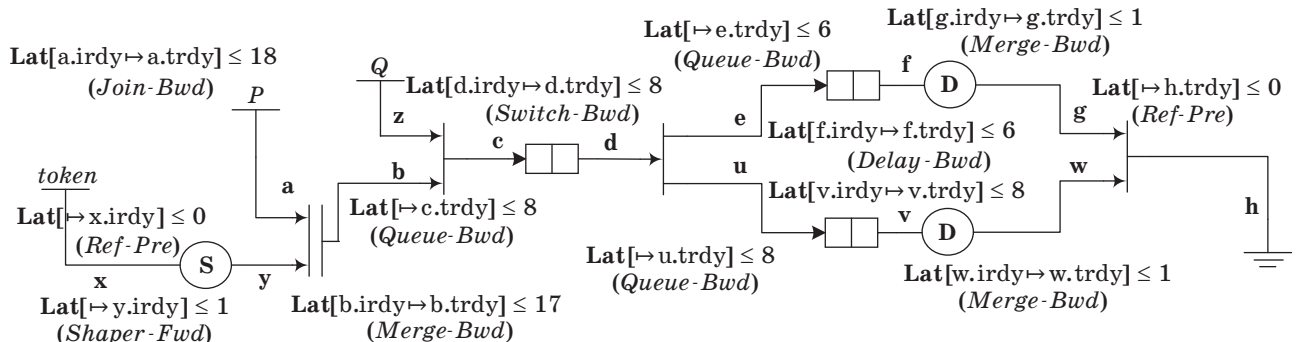
Канал g является выходом примитива задержки с входом f . Воспользуемся оценкой на канале g для получения аналогичной оценки на канале f . Из свойств примитива следует, что

$$\text{Lat}[f.irdy \mapsto (cnt = 0)] \leq 5, \quad (3)$$

т. е. при наличии сигнала $f.irdy$ внутренний счетчик задержки убывает до 0 не более чем за $d = 5$ тактов. Принимая неравенства (2) и (3) в качестве гипотез, можно показать, что

$$\text{Lat}[f.irdy \mapsto f.trdy] \leq 6.$$

Полный вывод локальной задержки на канале f в модели коммуникационной фабрики на рис. 4. приведен в табл. 2. В качестве началь-



■ Рис. 4. Локальные задержки в модели коммуникационной фабрики

Таблица 2. Пошаговый вывод для локальной задержки на канале f

№ шага	Следствие	Правило	Используемые предпосылки
1	$\text{Lat}[g.\text{irdy} \mapsto g.\text{trdy}] \leq 1$	—	Уравнение (2)
2	$\text{Lat}[f.\text{irdy} \mapsto (\text{cnt} = 0)] \leq 5$	—	Уравнение (3)
3	$\text{Lat}[f.\text{irdy} \mapsto f.\text{irdy} \cdot (\text{cnt} = 0)] \leq 5$	(Persist)	Шаг 2, $f.\text{irdy} \triangleright (\text{cnt} = 0)$
4	$\text{Lat}[f.\text{irdy} \mapsto g.\text{irdy}] \leq 5$	—	Шаг 3, $g.\text{irdy} = f.\text{irdy} \cdot (\text{cnt} = 0)$
5	$\text{Lat}[g.\text{irdy} \mapsto g.\text{irdy} \cdot g.\text{trdy}] \leq 1$	(Persist)	Шаг 1, $g.\text{irdy} \triangleright g.\text{trdy}$
6	$\text{Lat}[f.\text{irdy} \mapsto g.\text{irdy} \cdot g.\text{trdy}] \leq 6$	(Seq)	Шаги 4, 5
7	$\text{Lat}[f.\text{irdy} \mapsto f.\text{trdy}] \leq 6$	(Post-Or)	Шаг 6, $g.\text{irdy} \cdot g.\text{trdy} \rightarrow f.\text{trdy}$

ных шагов используются уравнения (2) и (3). Предпосылками для дальнейших шагов вывода выступают ранее полученные утверждения, а также свойства примитивов и канального протокола xMAS.

Этот пошаговый вывод опирается только на семантику примитива и свойства протокола xMAS и может быть обобщен в виде вспомогательного правила распространения (Delay-Bwd), приведенного выше.

Правило (Delay-Bwd) опирается на истинность гипотезы $\text{Lat}[i.\text{irdy} \mapsto (\text{cnt} = 0)] \leq d$ и ряд других свойств примитива и протокола, указанных в качестве предпосылок, используемых в табл. 2. Для удобства договоримся, что при использовании правила распространения все необходимые предположения автоматически добавляются к исходному множеству гипотез Г. Истинность полученной при этом оценки на задержку будет зависеть от истинности принятых гипотез.

Аналогично (Delay-Bwd), каждое правило связано с набором гипотез. Например, правило (Merge-Bwd) предполагает истинным неравенство

$$\text{Lat}[a.\text{irdy} \mapsto_{o.\text{xfer}} \sigma_a] \leq L_a.$$

Задержка арбитража L_a равна максимальному числу раундов арбитража до получения входом a наивысшего приоритета. Понятие наивысшего приоритета зависит от выбранного алгоритма арбитража и задается предикатом выбора σ_a , удовлетворяющего требованиям

$$\begin{aligned} & (a.\text{irdy} \cdot \sigma_a) \triangleright o.\text{xfer}; \\ & \sigma_a \rightarrow (a.\text{trdy} = a.\text{irdy} \cdot o.\text{trdy}), \end{aligned}$$

т. е. истинность σ_a гарантирует передачу на канале a. Предикат σ_a и задержка L_a могут быть определены для большинства справедливых алгоритмов арбитража. Например, для кругового алгоритма можно положить $\sigma_a = (s = i_a)$, где s — номер текущего выбранного входа (внутренняя переменная алгоритма), а i_a — номер входа a; L_a равна общему числу входов минус 1.

Подставляя оценки для каналов a, d и f (см. рис. 4) в формулу (1), получаем

$$\begin{aligned} & \text{Lat}[(a = P^*) \mapsto h.\text{xfer} \cdot (h = P^*)] \leq \\ & \leq 18 + 2 \cdot 8 + 2 \cdot 6 + 4 = 50. \end{aligned} \quad (4)$$

Описанный метод анализа применим к широкому классу микроархитектурных моделей. При отсутствии структурных циклов все локальные задержки находятся последовательным обходом с использованием соответствующих правил распространения. Оценка для многошаговой транзакции получается суммированием локальных оценок по всей ее траектории.

Для моделей со структурными циклами необходимо учитывать множество достижимых состояний системы. Например, для модели канала с кредитной логикой (см. рис. 2, б) выполнено соотношение

$$q_1.\text{num} + q_2.\text{num} = q_3.\text{num}. \quad (5)$$

Все состояния очередей, удовлетворяющие (5), показаны на рис. 5, а—в и перечислены в табл. 3. Для простоты положим размеры всех очередей равными 2.

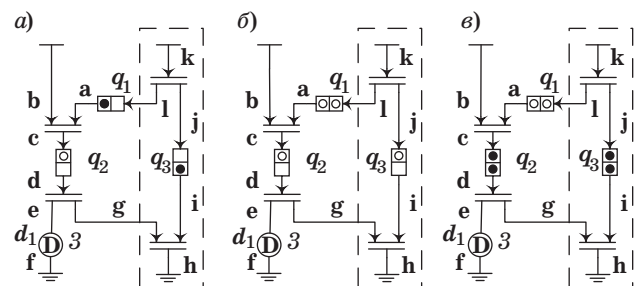


Рис. 5. Достижимые состояния A (а), B (б), C (в) канала с кредитной логикой²

² Метками черного цвета обозначены места в очередях, занятые пакетами. Белые метки обозначают свободные места. Состояние позиций, не имеющих меток, может быть любым.

■ Таблица 3. Допустимые состояния очередей

Состояние	q_1	q_2	q_3
A	Не пуста	Не полна	Не пуста
B	Пуста	Не полна	Не полна
C	Пуста	Полна	Полна

В состоянии A необходимый кредит доступен и передача на канале b может произойти немедленно, т. е. $\text{Lat}[A \cdot b.\text{irdy} \mapsto b.\text{trdy}] \leq 0$. Переходы между состояниями характеризуются задержками

$$\text{Lat}[B \mapsto A] \leq 1; \text{Lat}[C \mapsto B] \leq d,$$

где d — параметр примитива задержки. Пользуясь правилом (*Seq*) и устойчивостью сигнала $b.\text{irdy}$ относительно $b.\text{trdy}$, можно показать, что

$$\text{Lat}[B \cdot b.\text{irdy} \mapsto b.\text{trdy}] \leq 1;$$

$$\text{Lat}[C \cdot b.\text{irdy} \mapsto b.\text{trdy}] \leq d + 1.$$

Величина $\text{Lat}[b.\text{irdy} \mapsto b.\text{trdy}]$ для произвольного состояния модели может быть получена применением правила (*Pre-Or*) с учетом того, что $A + B + C$ истинно в каждом состоянии модели:

$$\text{Lat}[b.\text{irdy} \mapsto b.\text{trdy}] \leq \max\{0, 1, d + 1\} = d + 1 = 4.$$

Ранжирующие функции

Мы пользуемся правилами распространения задержек для примитивов xMAS, чтобы построить оценку на задержку обработки пакета. Вывод опирается на ряд предположений о свойствах примитивов и протокола передачи данных. На практике принятые гипотезы могут оказаться ложными (например, из-за изменений в семантике примитива), а в ходе вывода могут быть допущены ошибки. Для проверки правильности оценки воспользуемся средствами верификации моделей и методом ранжирующих функций.

Для двух произвольных событий A и B определим интервал от A до B как

$$A : B = A + \text{pre}(A : B) \cdot \text{pre}(\neg B).$$

Оператор pre возвращает значение своего аргумента на предыдущем такте (False в начальный момент времени). Выражение $(A : B)$ становится истинным при каждом наступлении события A и остается таковым до первого наступления события B .

Целая неотрицательная величина φ называется ранжирующей функцией для отношения отклика $A \mapsto_E B$, если

$$\mathbf{G}((A : B) \cdot \neg B \cdot \neg E \rightarrow \varphi^+ \leq \varphi) \text{ (Idle);}$$

$$\mathbf{G}((A : B) \cdot \neg B \cdot E \rightarrow \varphi^+ < \varphi) \text{ (Step),}$$

где выражение φ^+ равно значению величины φ на следующем такте.

Условие $(A : B) \cdot \neg B$ означает, что событие A уже наступило, а событие B — еще нет. В этой ситуации (*Idle*) требует невозрастания, а (*Step*) — строгого убывания φ при $E = \text{True}$. Содержательно величину φ можно считать «мерой» расстояния от текущего момента времени до наступления события B . Заметим, что поведение функции φ вне интервала $(A : B)$, а также в момент наступления события B может быть произвольным.

Теорема 2. Пусть $\text{rk}[A \mapsto_E B] \doteq \varphi$ и $\mathbf{G}(A \rightarrow (\varphi \leq M))$. Тогда $\text{Lat}[A \mapsto_E B] \leq M$.

Согласно теореме 2, для обоснования оценки M на задержку достаточно построить ранжирующую функцию φ и доказать, что выполнены условия (*Idle*), (*Step*) и $\mathbf{G}(A \rightarrow (\varphi \leq M))$. На практике часто удается построить простые ранжирующие функции и проверить их свойства средствами одношаговой индукции.

Аналогично задержке, вычисление сложной ранжирующей функции может быть сведено к более простым случаям. Однако при комбинировании ранжирующих функций для разных отношений отклика возникает техническая трудность, связанная с неопределенностью поведения функции φ , удовлетворяющей условиям (*Idle*) и (*Step*), вне интервала $A : B$. Пусть, например:

$$\text{rk}[A \mapsto B] \doteq \varphi;$$

$$\text{rk}[B \mapsto C] \doteq \psi.$$

По аналогии с правилом (*Seq*) для задержки, естественно было бы определить ранжирующую функцию для $A \mapsto C$ как $\varphi + \psi$. Однако поведение функции ψ может быть произвольным на интервале от наступления события A до наступления события B (аналогично для функции φ на интервале от B до C). Это не позволяет гарантировать убывания суммы $\varphi + \psi$ на всем интервале от A до C . В таких случаях мы будем доопределять значение функций, пользуясь двумя специальными операторами: \uparrow_X^M и \downarrow_X^M .

Для целочисленной величины φ , условия X и постоянной M определим

$$\varphi \downarrow_X \equiv \varphi \cdot [\neg X];$$

$$\varphi \uparrow_X^M \equiv \varphi \cdot [\neg X] + M \cdot [X].$$

Используя доопределяющие операторы, можно сформулировать правило

$$\text{rk}[A \mapsto_E B] \doteq \varphi;$$

$$\frac{\text{rk}[B \mapsto_E C] \doteq \psi, \quad \mathbf{G}(B \rightarrow (\psi \leq M))}{\text{rk}[A \mapsto_E C] \doteq \varphi \downarrow_{B:C} + \psi \uparrow_{\neg(B:C)}^M} \text{ (Rk-Seq)}.$$

Аналогично (*Rk-Seq*), каждому правилу вывода для задержек можно поставить в соответствие правило для ранжирующих функций. Роль гипотез при выводе выполняют базовые ранжирующие функции, описывающие поведение примитивов. Например, для примитива задержки с входом *i*, выходом *o* и параметром *d* справедливо равенство

$$\mathbf{rk}[i.\text{irdy} \mapsto (\text{cnt} = 0)] \doteq \text{cnt}.$$

Повторяя вывод, показанный на рис. 4, можно найти $\mathbf{rk}[f.\text{irdy} \mapsto f.\text{trdy}]$, $\mathbf{rk}[d.\text{irdy} \mapsto d.\text{trdy}]$ и $\mathbf{rk}[a.\text{irdy} \mapsto a.\text{trdy}]$. Из них по формуле, аналогичной (2), получается ранжирующая функция для полного отношения отклика. Далее с помощью индукции можно установить свойства (*Idle*) и (*Step*) и верхнюю границу 50 тактов, тем самым верифицировав оценку (4).

Результаты экспериментов

Описанный метод анализа задержек тестировался на небольших примерах. В табл. 4 сравниваются оценки на задержку обработки пакета *P* в модели на рис. 3, полученные разными методами. Напомним, что общая задержка (от *a* до *h*) ограничена задержками на каналах *f*, *d* и *a* по формуле (1).

Метод ограниченной верификации моделей (Bounded Model Checking — ВМС) исследует всевозможные исполнения модели из начального состояния в течение ограниченного числа тактов и позволяет получить нижнюю оценку задержки. *k*-шаговая индукция рассматривает исполнения модели из произвольного состояния (возможно, ограниченного дополнительными инвариантами [2]) и дает консервативную оценку сверху.

Оценки для ВМС и *k*-индукции можно считать приблизительно равными. Небольшое расхождение обусловлено возникающей синхронизацией состояний различных примитивов (например, внутренних счетчиков задержек и формирователей, текущих значений приоритетов в алгоритмах арбитража и т. д.). Такие эффекты крайне тяжело учесть, не прибегая к полному перебору состояний модели. Более интересны различия между двумя последними столбцами табл. 4. Полная задержка, полученная по правилам вывода, оказывается примерно в 1,5 раза больше оценки другими методами.

Расхождение можно сократить, повышая точность правил распространения, но полного совпадения не будет. Оценка в 50 тактов получена по формуле (1), где время пребывания в каждой точке траектории оценивается по наихудшему случаю. В реальных исполнениях модели наихудший случай реализуется только на подмно-

■ Таблица 4. Сравнение оценок задержки, полученных разными методами

Задержка	ВМС	<i>k</i> -индукция	Правила вывода
f	6	6	6
d	7	8	8
a	13	17	18
a→h	22	34	50

жестве точек траектории, что дает меньшее значение задержки. Например, можно показать, что если в модели на рис. 2, *a* пакет *P* надолго задерживается в очереди *q*₁, то время его прохождения через очередь *q*₂ будет заметно ниже наихудшего.

Интервалы играют важную роль при построении ранжирующих функций. Заметим, что в общем случае схемная реализация предиката *A:B* предполагает введение новых регистровых переменных для вычисления значения оператора **pre**. Наличие дополнительных переменных, не ограниченных инвариантами, может нарушать сходимость метода *k*-шаговой индукции. На практике интервал всегда можно заменить более простым выражением. Например, если *A* устойчиво по отношению к *B*, то (*A:B*) = *A*.

Использование ранжирующих функций должно давать значительный выигрыш по производительности в сложных случаях. Например, для моделей с числом очередей порядка 10–100 оценка ВМС может быть очень далека от точной, а *k*-шаговая индукция не завершается за разумное время. Мы полагаем, что метод ранжирующих функций будет работать на больших моделях, и планируем проверить это экспериментально. Для простых моделей большинство алгоритмов верификации позволяет получить результат за короткое время (порядка нескольких секунд), и сравнение их быстродействия не представляет интереса.

Заключение

Разработан метод анализа задержек в микроархитектурных моделях коммуникационных фабрик. Метод основан на последовательном применении элементарных свойств задержки для вывода все более сложных оценок из простых гипотез, описывающих поведение отдельных примитивов, и протокола передачи данных. Итоговая оценка консервативно предполагает наихудшую возможную задержку в каждой точке анализируемой траектории. Предложен способ построения ранжирующих функций, позволяющий провести быструю верификацию полученной оценки.

Литература

1. Dally W. J., Towles B. Principles and Practices of Interconnection Networks. — Morgan Kaufmann Publishers, 2004. — 550 p.
2. Chatterjee S., Kishinevsky M. Automatic generation of inductive invariants from highlevel microarchitectural models of communication fabrics // Computer Aided Verification'10. LNCS. 2010. Vol. 6174. P. 321–338.
3. Galton A. et al. Temporal Logic. Feb. 2008. <http://plato.stanford.edu/entries/logic-temporal/> (дата обращения: 15.12.2011).
4. McMillan K. L. Interpolation and SAT-based Model Checking // Proc. CAV'03. 2003. P. 1–13.
5. Sheeran M. et al. Checking Safety Properties Using Induction and a SAT-Solver // FMCAD. LNCS. 2000. Vol. 1954. P. 108–125.
6. Bradley A. SAT-based model checking without unrolling // Proc. VMCAI'11. 2011. P. 70–87.
7. Chawdhary A. et al. Ranking Abstractions // Proc. ESOP. 2008. P. 148–162.
8. Готманов А., Кишиневский М., Чэттерджи С. Моделирование и верификация коммуникационных фабрик при проектировании систем на кристалле // Проблемы разработки перспективных микро- и нанoeлектронных систем — 2012: сб. тр. / ИППМ РАН. М., 2012. С. 61–66.

УВАЖАЕМЫЕ АВТОРЫ!

При подготовке рукописей статей необходимо руководствоваться следующими рекомендациями.

Статьи должны содержать изложение новых научных результатов. Название статьи должно быть кратким, но информативным. В названии недопустимо использование сокращений, кроме самых общепринятых (РАН, РФ, САПР и т. п.).

Объем статьи (текст, таблицы, иллюстрации и библиография) не должен превышать эквивалента в 20 страниц, напечатанных на бумаге формата А4 на одной стороне через 1,5 интервала Word шрифтом Times New Roman размером 13, поля не менее двух сантиметров.

Обязательными элементами оформления статьи являются: индекс УДК, заглавие, инициалы и фамилия автора (авторов), ученая степень, звание (при отсутствии — должность), полное название организации, аннотация и ключевые слова на русском и английском языках, электронные адреса авторов, которые по требованию ВАК должны быть опубликованы на страницах журнала. При написании аннотации не используйте аббревиатур и не делайте ссылок на источники в списке литературы.

Статьи авторов, не имеющих ученой степени, рекомендуется публиковать в соавторстве с научным руководителем, наличие подписи научного руководителя на рукописи обязательно; в случае самостоятельной публикации обязательно предоставляйте заверенную по месту работы рекомендацию научного руководителя с указанием его фамилии, имени, отчества, места работы, должности, ученого звания, ученой степени — эта информация будет опубликована в ссылке на первой странице.

Формулы набирайте в Word, не используя формульный редактор (Mathtype или Equation), при необходимости можно использовать формульный редактор; для набора одной формулы не используйте два редактора; при наборе формул в формульном редакторе знаки препинания, ограничивающие формулу, набирайте вместе с формулой; для установки размера шрифта никогда не пользуйтесь вкладкой Other..., используйте заводские установки редактора, не подгоняйте размер символов в формулах под размер шрифта в тексте статьи, не растягивайте и не сжимайте мышью формулы, вставленные в текст; в формулах не отделяйте пробелами знаки: + = -.

Для набора формул в Word никогда не используйте Конструктор (на верхней панели: «Работа с формулами» — «Конструктор»), т. к. этот ресурс предназначен только для внутреннего использования в Word и не поддерживается программами, предназначенными для изготовления оригинал-макета журнала.

При наборе символов в тексте помните, что символы, обозначаемые латинскими буквами, набираются светлым курсивом, русскими и греческими — светлым прямым, векторы и матрицы — прямым полужирным шрифтом.

Иллюстрации в текст не заверстываются и предоставляются отдельными исходными файлами, подающимися редактированию: — рисунки, графики, диаграммы, блок-схемы изготавливаются в векторных программах: Visio 4, 5, 2002–2003 (*.vsd); Coreldraw (*.cdr); Excel; Word; AdobeIllustrator; AutoCad (*.dxf); Компас; Matlab (*.ps, *.pdf или экспорт в формат *.ai); — фото и растровые — в формате *.tif, *.png с максимальным разрешением (не менее 300 pixels/inch).

Наличие подписанных подписей обязательно (желательно не повторяющих дословно комментарии к рисункам в тексте статьи).

В редакцию предоставляются:

— сведения об авторе (фамилия, имя, отчество, место работы, должность, ученое звание, учебное заведение и год его окончания, ученая степень и год защиты диссертации, область научных интересов, количество научных публикаций, домашний и служебный адреса и телефоны, e-mail), фото авторов: анфас, в темной одежде на белом фоне, должны быть видны плечи и грудь, высокая степень четкости изображения без теней и отблесков на лице, фото можно представить в электронном виде в формате *.tif, *.png с максимальным разрешением — не менее 300 pixels/inch при минимальном размере фото 40 × 55 мм; — экспертное заключение.

Список литературы составляется по порядку ссылок в тексте и оформляется следующим образом:

— для книг и сборников — фамилия и инициалы авторов, полное название книги (сборника), город, издательство, год, общее количество страниц;

— для журнальных статей — фамилия и инициалы авторов, полное название статьи, название журнала, год издания, номер журнала, номера страниц;

— ссылки на иностранную литературу следует давать на языке оригинала без сокращений;

— при использовании web-материалов указывайте адрес сайта и дату обращения.

Более подробно правила подготовки текста с образцами изложены на нашем сайте в разделе «Оформление статей».

Контакты

Куда: 190000, Санкт-Петербург,

Б. Морская ул., д. 67, ГУАП, РИЦ

Кому: Редакция журнала «Информационно-управляющие системы»

Тел.: (812) 494-70-02

Эл. почта: 80x@mail.ru

Сайт: www.i-us.ru