

УДК 004.414.23

АРХИТЕКТУРА СИСТЕМЫ РАСПРЕДЕЛЕННОГО МОДЕЛИРОВАНИЯ ЦИФРОВЫХ СИГНАЛОВ

Лысенко Д. В., Морозов М. С., Святный В.А.

Донецкий национальный технический университет

Рассмотрен один из вариантов архитектуры системы моделирования цифровых сигналов, позволяющий производить моделирование распределенно. Предложены основные принципы использования такой архитектуры, позволяющие определять возможные нестандартные режимы работы функциональных узлов схемы. Данные принципы позволяют находить сложные ошибки в схеме, например, если несколько ошибок друг друга компенсируют и при обычном моделировании схема выдает верные результаты.

Одним из необходимых этапов проектирования любой сложной (в том числе и цифровой) техники является ее моделирование. Оно является одним из ключевых методов верификации и диагностики проекта до его производства. С появлением схем высокой степени интеграции, их создание без моделирования стало практически невозможным. Например, процессор Intel Itanium 2 содержит более миллиарда транзисторов[1]. Даже с использованием моделирования разработка такого устройства является крайне трудоемкой задачей. Но несмотря на использование новейших технологий при проектировании данной архитектуры, после выпуска в ней были обнаружены недостатки, например, высокие задержки кэш-памяти третьего уровня. Это была одной из причин коммерческого проигрыша данной архитектуры конкурентам (AMD64, SPARC). Также в крупных проектах были допущены ошибки, которые были следствием неполной верификации. Например, допущенная на аппаратном уровне процессоров Intel Pentium MMX ошибка, известная как F00F-bug. При выполнении инструкции «lock stpxchg8b eax» процессор перестает обрабатывать прерывания[2].

При массовом производстве какого-либо проекта вероятность проявления ошибок существенно возрастает. Поэтому диагностика проекта является одним из ключевых этапов проектирования сложных устройств, который может существенно отразиться на коммерческом успехе проекта. Таким образом, совершенствование принципов моделирования и систем моделирования в целом все еще остаются актуальными задачами.

Важность решения указанных общих проблем обусловила цель данной работы, которая состоит в разработке системы параллельного моделирования больших цифровых схем.

Для достижения поставленной цели необходимо сначала проанализировать некоторые распространенные системы моделирования. В настоящее время существует ряд программных систем моделирования цифровых схем. Среди них коммерческие продукты: NI MultiSim (ранее – Electronic Workbench), Micro-CAP, Active-HDL.

5 NI MultiSim и Micro-CAP являются SPICE-подобными системами. Они предоставляют очень широкие возможности моделирования и анализа аналоговых схем. Также возможно моделирование и цифровых схем. MultiSim и Micro-CAP обладают очень удобными интерфейсами для описания схем, не требуют знания языка описания схем. Это позволяет быстро освоить данные системы; моделированием простых схем могут заниматься не только квалифицированные инженеры.

Active-HDL производит моделирование проектов, описанных с помощью VHDL. VHDL в настоящее время является очень распространенным языком описания аппаратного обеспечения. На VHDL созданы большие библиотеки различных схем, его использование делает моделирование проекта очень удобным и уменьшает трудоемкость описания схемы.

Данным системам свойственны и некоторые недостатки. Все они являются закрытыми коммерческими проектами. Эти среды моделирования разработаны под одно семейство ОС – Windows. Моделирование очень больших схем становится невозможным

из-за слишком ограниченных вычислительных ресурсов. Также вследствие закрытости проектов самостоятельный перенос их на другие архитектуры невозможен в принципе.

Существуют и открытые системы моделирования: SPICE, Logisim. SPICE является системой моделирования схем общего назначения. Его можно использовать для моделирования цифровых и аналоговых схем.

Logisim – кроссплатформенная система моделирования цифровых схем. Система используется преимущественно в образовательных целях[3], для моделирования больших схем не предназначена.

Таким образом, все рассмотренные системы моделирования осуществляют моделирование нераспределено и их использование для симулирования больших схем затруднено. Поэтому следует рассматривать иные варианты моделирования больших схем.

К такой системе моделирования можно предъявить три основных требования:

1. Система должна как можно точнее описывать работу схемы, быть гибкой и расширяемой. Описание структурных единиц моделируемой схемы желательно делать на языке высокого уровня.
2. Система должна выполнять моделирование схемы за приемлемое время. Для ускорения процесса можно использовать распределенные вычисления.
3. Система должна быть удобной для пользователя (user-friendly), способствовать уменьшению количества ошибок при описании схемы в целом и ее структурных элементов.

Результаты моделирования будут более достоверными, если в схеме проверять коэффициент разветвления, учитывать временные задержки. Так же желательно учесть разброс параметров элементов. При моделировании цифровых схем чаще всего задержки считают максимально возможными, либо производят моделирование без задержек. В таком случае невозможно определить гонки сигналов по результатам симуляции.

Для моделирования больших схем приобретает актуальность проблема распараллеливания процесса моделирования. Всю схему нужно условно разделить на некоторые блоки, расчеты внутри которых можно выполнять параллельно. При этом нужно попытаться обеспечить равномерность загрузки процессоров, которые используются системой моделирования.

Рассмотрим один из возможных вариантов архитектуры системы моделирования цифровых схем. Внутри системы моделирования схему можно представить в виде направленного графа. При этом вершинам будут соответствовать некоторые логические блоки, а ребрам – связи между блоками. Блоками могут быть либо простейшие логические элементы, либо участки схемы. Направления ребер будут указывать на направленность связей. Например, если некоторые выходы логического блока 1 подключены к некоторым входам логического блока 2, то в схеме будет присутствовать ребро, направленное от вершины 1 к вершине 2. Пример представления простейшей схемы приведен на рис. 1.

Граф представляет неполную информацию о схеме. Дополнительно необходимо ввести характеристики ребер – список соединений, которые описывает каждое ребро. Для данного примера характеристики ребер приведены на рис. 2.

Следует отметить, что одно ребро может описывать несколько связей.

Для распараллеливания процесса моделирования схему нужно разделить на некоторые блоки, внутри блоков моделирование

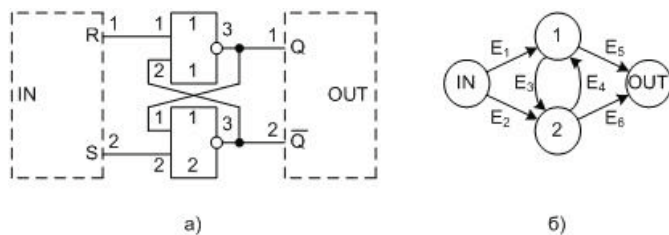


Рисунок 1 – Пример цифровой схемы (а) и ее представление в виде графа (б)

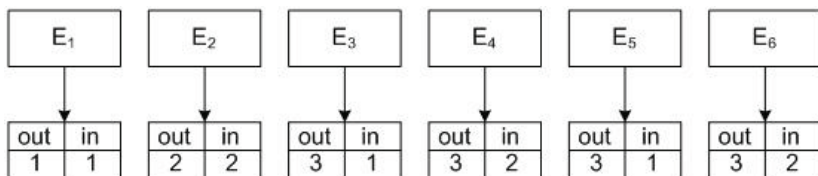


Рисунок 2 – Характеристики ребер

осуществлять параллельно. Блоки необходимо формировать таким образом, чтобы между ними было минимальное число связей. Желательно, чтобы блоки были одинакового размера. Это позволит несколько «выровнять» загрузку системы.

Логическая структура одного блока изображена на рис. 3.

Очередь событий содержит все изменения входных сигналов, задержки и другие изменения внутренних состояний логических элементов. В случае изменения входных сигналов управляющий класс принимает от других блоков новые значения сигналов и добавляет их в очередь. Также управляющий класс анализирует очередь событий и в случае необходимости вызывает классы логических элементов для обработки событий. Классы логических элементов также могут добавлять события в очередь и удалять их. Таким образом, реализуются различные задержки элементов, изменение их внутренних состояний. В случае изменения сигналов на выходах логических элементов вызывается управляющий класс, который анализирует эти сигналы и при необходимости передает

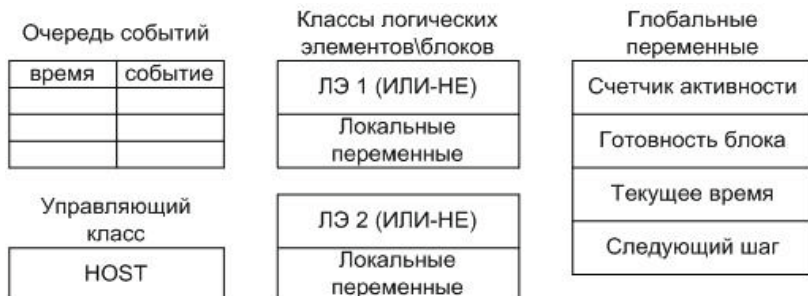


Рисунок 3 – Общая структура одного блока моделирования

новые значения выходов в другие блоки.

В системе должны присутствовать и глобальные переменные. Например, счетчик активности. В начале каждого шага моделирования (при изменении текущего времени) он сбрасывается, а при вызове классов логических переменных – наращивается. Данный счетчик используется в системе для предотвращения заикливания.

Флаг готовности блока взводится в случае, если в очереди событий все события обработаны. В системе для каждого блока присутствует свой флаг готовности. Когда флаги готовности для всех блоков взведены, можно переходить к следующему шагу моделирования.

Переменная «текущее время» показывает момент времени, для которого осуществляется обработка событий. В переменную «следующий шаг» блоки заносят время ближайшего события.

Рассмотрим пример работы системы для моделирования простейшей схемы (рис. 1). Для упрощения будем использовать логику с тремя состояниями: 1 – логическая единица, 0 – логический ноль, X – неопределенное состояние. Время задержки элементов будем считать равным 10 мкс.

Временные диаграммы приведены на рис. 4.

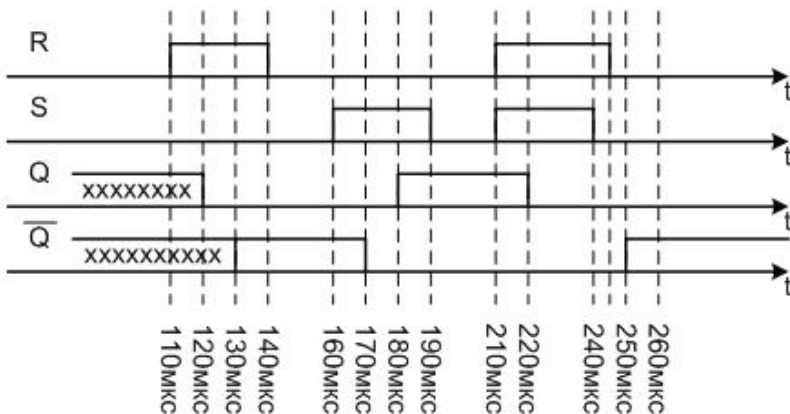


Рисунок 4 – Временные диаграммы входных и выходных сигналов

Таблица 1 – Очередь событий

Время	Событие	Примечание
110	Изменение вх. сигналов ЛЭ1	
120	Задержка ЛЭ1	
120	Изменение вх. сигналов ЛЭ2	
130	Задержка ЛЭ2	
130	Изменение вх. сигналов ЛЭ1	
140	Изменение вх. сигналов ЛЭ1	
160	Изменение вх. сигналов ЛЭ2	
170	Задержка ЛЭ2	
170	Изменение вх. сигналов ЛЭ1	
180	Задержка ЛЭ1	
180	Изменение вх. сигналов ЛЭ2	
190	Изменение вх. сигналов ЛЭ2	
210	Изменение вх. сигналов ЛЭ1	
210	Изменение вх. сигналов ЛЭ2	
220	Задержка ЛЭ1	
240	Изменение вх. сигналов ЛЭ2	
245	Изменение вх. сигналов ЛЭ1	
250	Задержка ЛЭ2	
250	Изменение вх. сигналов ЛЭ1	
255	Задержка ЛЭ1	удалена

Рассмотрим подробнее формирование очереди событий:

Инициализация – пусть выходы логических элементов изначально находятся в неопределенном состоянии.

110 мкс – изменяется один из входных сигналов логического элемента (ЛЭ) 1. На входе появляются сигналы 1 и X. В результате работы ЛЭ на выходе должно установиться значение $Q = \overline{1 \vee X} = 0$. Так как текущее и новое значения различаются, в очередь добавляется новое событие – задержка.

120 мкс – по истечении задержки на выходе ЛЭ1 устанавливается новое значение. Так как выход ЛЭ1 соединен со входом ЛЭ2, добавляется событие изменения входных сигналов для ЛЭ2.

120 мкс – изменение входов ЛЭ2. Новое значение выхода

$\overline{Q} = \overline{0 \vee 0} = 1$. Добавляется событие задержки

130 мкс – проходит время задержки ЛЭ2, изменяется сигнал на входе ЛЭ1 – создается соответствующее событие.

130 мкс – изменение входных сигналов ЛЭ1. Новое значение выхода $Q = \overline{1 \vee 1} = 0$, то есть значение не меняется.

140 мкс – изменение входных сигналов ЛЭ1. Новое значение выхода $Q = \overline{0 \vee 1} = 0$ равно старому, событие задержки не добавляется.

160 мкс – изменение входных сигналов ЛЭ2. $\overline{Q} = \overline{1 \vee 0} = 0$. $\overline{Q}^t \neq \overline{Q}^{t-1}$, создается событие задержки.

Далее моделирование будет происходить аналогично до 240 мкс.

240 мкс – изменение входных сигналов ЛЭ2. Новое значение $\overline{Q} = \overline{0 \vee 0} = 1$. Добавляется событие задержки.

245 мкс – изменение входных сигналов ЛЭ1. Новое значение $\overline{Q} = \overline{0 \vee 0} = 1$. Добавляется задержка.

250 мкс – изменение выходного сигнала ЛЭ2. Изменяется входной сигнал ЛЭ1.

250 мкс – ЛЭ1 находится в состоянии переходного процесса – переключения из 0 в 1. Для примера реализуем инерционную задержку, в таком случае событие задержки ЛЭ1 на 255 мкс удаляется из очереди.

Выводы

На основе краткого анализа распространенных систем моделирования, выявлены некоторые проблемы моделирования больших цифровых схем, среди которых основными являются необходимость их параллельного моделирования и возможность расширения функционала системы. Решению названных проблем способствует предложенная в данной работе архитектура системы моделирования больших цифровых схем. Преимущества данной архитектуры – возможность распараллеливания, большая гибкость.

Среди недостатков можно отметить высокую сложность описания работы логических элементов и большое количество связей между блоками, что может значительно замедлить моделирование. В качестве перспективы дальнейших исследований видится разработка интерфейса, языка описания схем, системы оптимального разделения схемы на блоки с целью ускорения процесса моделирования.

Литература

- [1] Intel Microprocessor Quick Reference Guide [Электронный ресурс] – Режим доступа: <http://www.intel.com/pressroom/kits/quickreffam.htm>
- [2] Net reacts to “F0” Pentium bug - CNET News [Электронный ресурс] – Режим доступа: <http://news.cnet.com/2100-1033-205157.html>
- [3] Logisim [Электронный ресурс] – Режим доступа: <http://ozark.hendrix.edu/~burch/logisim/ru/index.html>