

*А. А. Веселов, докт. техн. наук, профессор кафедры электронно-вычислительных машин
ФГБОУ ВО «Тверской государственный технический университет», г. Тверь, evm.tstu@gmail.com*

Экспериментальные исследования распределенной модели на основе сетей Петри

В статье приводятся результаты экспериментальных исследований эффективности применения распределенных моделей по сравнению с их монолитными аналогами. В качестве базовой модели используются модели, построенные на основе математического аппарата сетей Петри, в частности, особого расширения ординарных, временных сетей Петри с ингибиторными дугами, ориентированного на моделирование цифровых устройств автоматики и вычислительной техники. Показано, что быстродействие распределенных моделей может быть на два порядка больше, а время их загрузки в десятки раз меньше, чем аналогичные показатели у монолитных моделей.

Ключевые слова: распределенная и монолитная модели, сети Петри, компонента, функциональный модуль, внешний представитель, иерархическое представление, эффективность, быстродействие, время загрузки, размеры моделей.

Введение

Непрерывный рост размеров и сложности устройств современной цифровой электронной техники и сокращение сроков их морального старения приводит к необходимости дальнейшего совершенствования процессов их проектирования. Наиболее заметную роль в решении этой задачи играют инструментальные средства имитационного моделирования. Обычно они строятся на основе использования так называемых монолитных (или централизованных) моделей. Однако, как показывает опыт, такие модели применимы только при моделировании достаточно простых объектов небольшого размера. Увеличение размеров и сложности монолитных моделей приводит к тому, что они становятся чрезмерно громоздкими и медленными. В связи с этим в последнее время все большее внимание начинают обращать на себя распределенные модели.

В настоящее время в мировой практике уже имеется хорошо развитая теория [1–3] и инструментальные средства для реализации подобных систем. Наиболее известными из них являются «CPN-tools» [4], «Мера» [5] и «Диана» [6]. Несмотря на их несомненные достоинства, все они обладают одним общим недостатком, заключающимся в сложности их теоретической модели и программной реализации. Кроме того, такие системы были ориентированы на решение очень широкого круга задач и поэтому их, как правило, трудно приспособить для эффективного использования в конкретной предметной области. С учетом этого автором было разработано специальное расширение сетей Петри (DPN), ориентированное на моделирование поведения устройств цифровой электронной техники [7]. Затем на базе этого расширения была создана его распределенная модификация, получившая условное наименование DDPN [8]. Для проверки работоспособности

и оценки рабочих характеристик этой модели разработана система имитационного моделирования устройств цифровой автоматики и вычислительной техники DDPN-tools [9]. Система позволяет создавать как монолитные модели цифровых устройств, так и их распределенные аналоги. Все результаты, приведенные в данной работе, были получены с помощью именно этой системы.

Распределенная модель

Отличительная особенность распределенной DDPN-модели от других заключается в представлении ее компонент в виде двух удаленных друг от друга и взаимодействующих между собой частей: представительской и функциональной. Представительская часть компоненты выполняет роль удаленного полномочного представителя функционального модуля и выставляет внешнему окружению только его интерфейс. Через этот интерфейс внешняя среда способна оказывать дистанционное воздействие на удален-

ный функциональный модуль и получать от него соответствующие ответные реакции. Функциональная часть компоненты (функциональный модуль) реализует поведение моделируемого объекта. Взаимодействие между функциональным модулем компоненты и ее внешним представителем осуществляется путем обмена соответствующими сообщениями. В результате такого подхода к реализации распределенной компоненты внешней среде уже не требуется взаимодействовать непосредственно с функциональным модулем, а можно делать это через его внешнего представителя. А поскольку интерфейс внешнего представителя полностью совпадает с интерфейсом представляемого им функционального модуля, то у внешней среды создается иллюзия того, что он взаимодействует непосредственно с самим функциональным модулем компоненты. Это как раз и позволяет пользователю работать с удаленными моделями функциональных модулей через их внешних представителей, не вникая в детали и особенности их реализации (рис. 1).

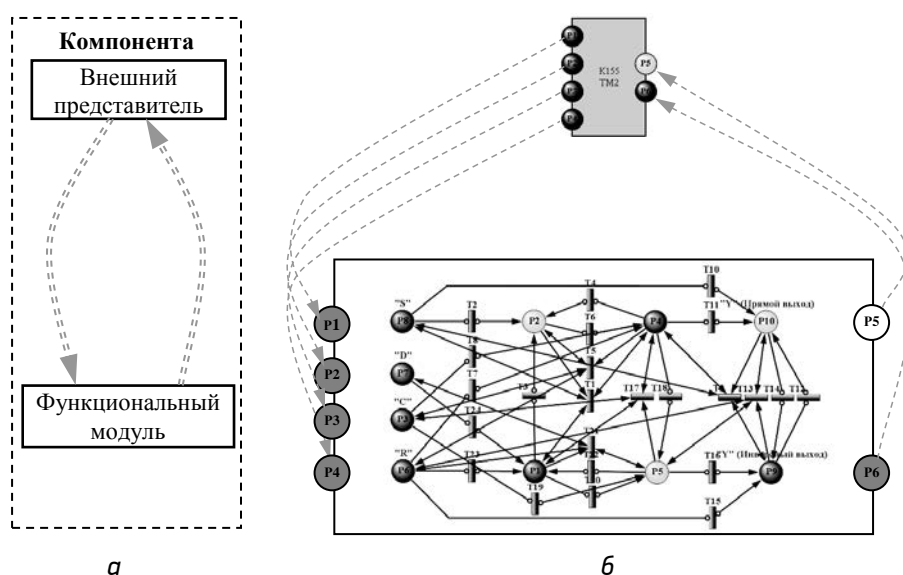


Рис. 1. Абстрактное (а) и конкретное (б) представление распределенной компоненты на примере модели функционального элемента триггера TM2

Fig. 1. The abstract and specific representation of the distributed component on the example model of the functional element of the TM2 trigger

Как видно из рис. 1, каждая компонента представляет собой элементарную двухуровневую структуру, состоящую из двух частей: функционального модуля компоненты (ФМ) и ее внешнего представителя (ВП). В процессе функционирования компоненты любое изменение состояния на входах ВП (P1-P4) приводит к формированию соответствующего сообщения, воспринимаемого ФМ. В ответ на это входное сообщение ФМ формирует ответную реакцию. Если эта реакция приводит к изменению состояния выходных позиций ФМ, то им формируется ответное сообщение для своего ВП, в результате которого происходит изменение состояния на выходах (P5, P6) его ВП. Таким образом, ВП предназначен только для предоставления на верхнем иерархическом уровне функциональности, реализуемой представляемым им ФМ.

В свою очередь каждый функциональный модуль компоненты тоже можно рассматривать как распределенную систему, включающую в свой состав группу внешних представителей функциональных модулей подчиненных компонент (рис. 2 и 3).

Из рисунка 3 видно, что функциональные модули подчиненных компонент не могут взаимодействовать между собой непосредственно. Это возможно только через их внешних представителей, входящих в состав функционального модуля компоненты, расположенной на более верхнем иерархическом уровне и владеющей внешними представителями функциональных модулей подчиненных компонент.

Показатели для оценки эффективности

Эффективность распределенных моделей оценивалась путем измерения рабочих характеристик и их сопоставления с аналогичными характеристиками их монолитных аналогов. В качестве таких рабочих характеристик были приняты следующие: быстродей-

ствие (скорость функционирования) моделей, время их загрузки и размеры.

Оценка быстродействия. Хорошо известно, что непосредственное измерение времени выполнения функций программного приложения сталкивается со многими трудностями, связанными с особенностями функционирования ОС современных компьютеров. Кроме того, время выполнения отдельных операций не всегда позволяет судить о быстродействии модели в целом. Поэтому в качестве показателя эффективности моделей по быстродействию было принято использовать значение модельного времени (T_m), достигаемого ими за некоторый фиксированный отрезок астрономического (физического) времени (T_f). Для этого модель запускалась на функционирование в течение некоторого заданного интервала времени, например, 600 с. В результате внутри модели начинает выполняться некоторая последовательность событий, вызванных срабатыванием ее переходов. Срабатывание каждого из них требует затраты некоторого времени, характеризующего временные задержки в моделируемых объектах, что приводит к постепенному изменению модельного времени. Очевидно, что, чем на большую величину продвинется модельное время за отведенный промежуток астрономического времени, тем выше ее быстродействие.

Для получения сравнительной оценки, по которой можно было бы судить о степени эффективности, принят безразмерный показатель эффективности по быстродействию ($K_{эб}$), определяемый в виде отношения значения модельного времени внутри распределенной модели ($T_{рм}$) к аналогичному значению в монолитной модели ($T_{мм}$):

$$K_{эб} = T_{рм} / T_{мм}.$$

Оценка времени загрузки. В качестве этого показателя использовалось время с начала загрузки, включая время, необходимое для отображения модели в виде соответствующего графа. Для вычисления сравнитель-

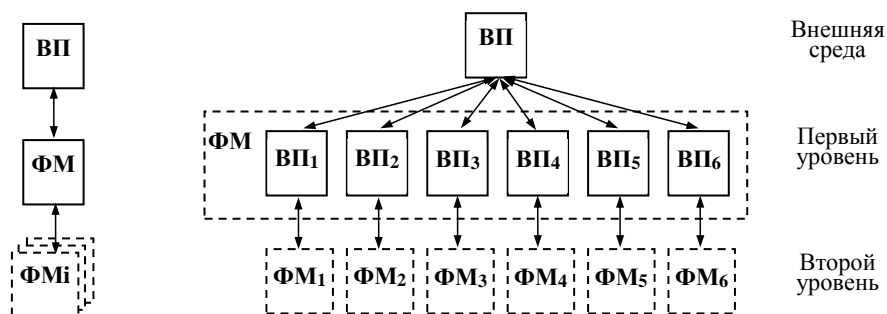


Рис. 2. Общий вид трехуровневой структуры распределенной модели функционального элемента триггера K155TM2

Fig. 2. A general view of three-level structure of the distributed model of the functional element of the K155TM2 trigger

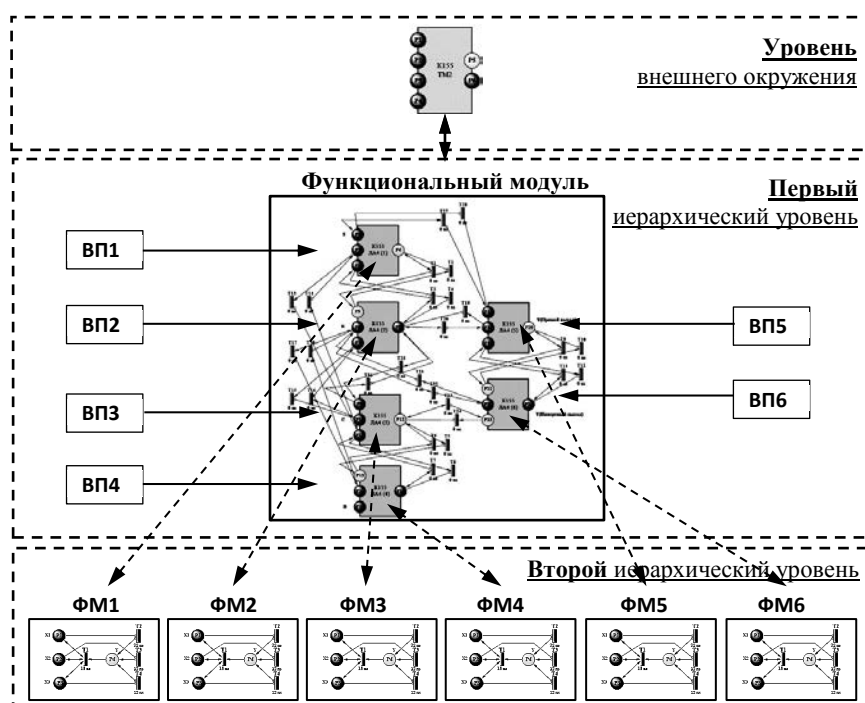


Рис. 3. Трехуровневая реализация распределенной модели функционального элемента триггера K155TM2

Fig. 3. Three-level implementation of the distributed model of the functional element of the K155TM2 trigger

ной оценки времени загрузки распределенной модели (Тзрм) и ее монолитного аналога (Тзм) использовался показатель в виде их отношения:

$$K_{эз} = T_{зрм} / T_{зм}.$$

Условия проведения эксперимента

Проводимые экспериментальные исследования были направлены на определение эффективности применения распределенных имитационных моделей по сравнению с мо-

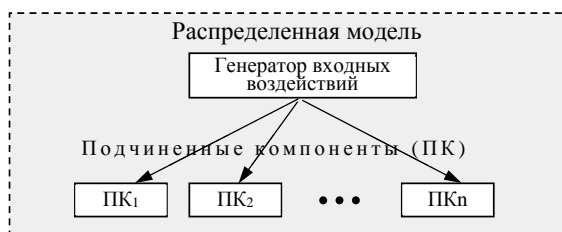


Рис. 4. Общая схема увеличения размеров моделируемой схемы

Fig. 4. The general diagram of increase in the sizes of the modelled schema

нолитными в зависимости от их размера. Для упрощения размер моделей принято изменять путем последовательного увеличения количества содержащихся в ней одинаковых компонент. Это позволяет достаточно просто создавать модели с различными размерами. Общая схема реализации такого подхода показана на рис. 4.

Примеры конкретной реализации распределенной модели с одним и двумя функциональными элементами JK-триггеров показаны на рис. 5. На этом рисунке позиция P1 и переходы T1, T2 моделируют генератор входных воздействий, поступающих на счетные входы триггеров. Изменение времени задержки переходов T1 и T2 позволяет задавать частоту изменения входных воздействий. Она выбирается таким образом, чтобы ее величина не превышала максимально допустимых значений, характерных для реальных функциональных элементов триггера, входящих в состав моделируемой схемы. Соотношение задержек в переходах T1 и T2 позволяет изменять скважность сигнала.

На рисунке 5 видно, что распределенная реализация модели дает возможность видеть только самое общее о ней представление, скрывающее детали внутреннего устройства

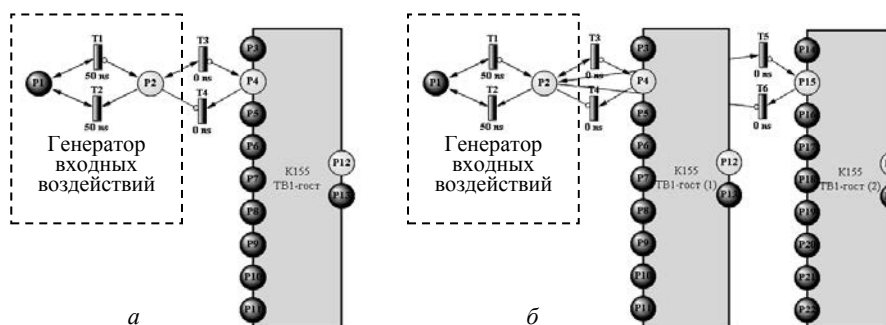


Рис. 5. Двухуровневые распределенные модели с одним (а) и двумя (б) JK-триггерами

Fig. 5. The two-level distributed models with one (a) and two (б) JK-flip-flops

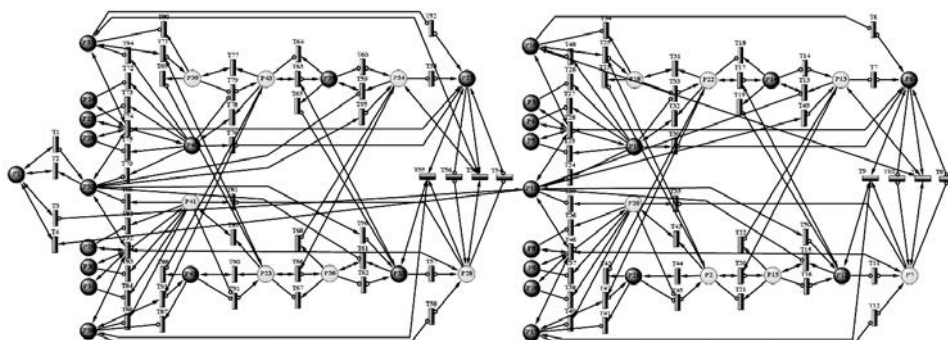


Рис. 6. Монолитный вариант модели схемного решения с двумя JK-триггерами

Fig. 6. Monolithic variant of model of the circuit decision with two JK-flip-flops

каждой из ее компонент. Отображается состояние и поведение модели только на ее самом верхнем иерархическом уровне. Однако при необходимости пользователь всегда может раскрыть содержимое любой из ее подчиненных компонент и пронаблюдать их состояние и поведение уже на более детальном уровне (см. рис. 3).

Для сравнения: на рис. 6 показано изображение монолитного аналога модели с двумя JR-триггерами, представленной на рис. 5б. Как видно из этого рисунка, в монолитном варианте реализации модели полностью отображается структура модели со всеми деталями и особенностями ее реализации.

Структуры моделей, подобные представленным на рис. 5 и 6, были использованы для проведения экспериментальных исследований монолитных и распределенных моделей, содержащих компоненты, моделирующие поведение и других функциональных элементов цифровой электронной техники.

Исследования быстродействия моделей

Практический опыт работы с моделями показывает, что при увеличении размеров их реакция на внешние воздействия становится все более замедленной. Другими словами, по мере роста размеров модели становятся все более медленными. То же самое относится и к распределенным моделям. Покажем это на примере, приведенном на рис. 7, на котором представлены результаты экспериментальных исследований влияния размеров монолитной и двухуровневой распределенной моделей на достижимое ими модельное время. Измерение модельного времени проводилось при значении интервала астрономического времени $T_f = 180$ с. Как уже говорилось, это время непрерывной работы моделей, в результате функционирования которых в них происходило изменение модельного времени. После окончания этого интервала фиксировалось значение модель-

ного времени. Изменение размера моделей проводилось путем увеличения количества содержащихся в них моделей JK-триггеров в диапазоне от 1 до 60.

Из рисунка 7 видно, что модельное время (T_m) в распределенной модели, во всем диапазоне изменения количества компонент, достигает значений, которые заметно больше, чем у монолитных моделей (T_{mm}). Это означает, что распределенные модели работают быстрее, чем их монолитные аналоги. Насколько велико это увеличение, показано на графике изменения показателя эффективности по быстродействию ($K_{эб}$). Так, при увеличении размеров моделей в 60 раз от своего первоначального значения быстродействие распределенной модели может быть почти в 18 раз лучше по сравнению с монолитной моделью, т. е. не на проценты или десятки процентов, а более, чем на порядок.

Однако при увеличении количества иерархических уровней у распределенных моделей влияние изменения размера на их быстродействие несколько изменяется (рис. 8). Эти изменения проявляются в том, что достигаемые значения модельного времени и соответствующие показатели быстродействия становятся меньше, чем у моделей с меньшим количеством иерархических уровней.

Анализ графиков на приведенном рисунке позволяет утверждать следующее:

- преимущество распределенных моделей по сравнению с их монолитными аналогами на всем исследуемом диапазоне изменения их размеров не постоянно, а постепенно увеличивается (по мере увеличения размеров распределенных моделей);
- характер изменения графиков эффективности по быстродействию у распределенных моделей с разным количеством иерархических уровней практически совпадает;
- введение новых иерархических уровней в распределенную модель приводит к уменьшению ее эффективности по быстродействию. Например, при количестве компо-

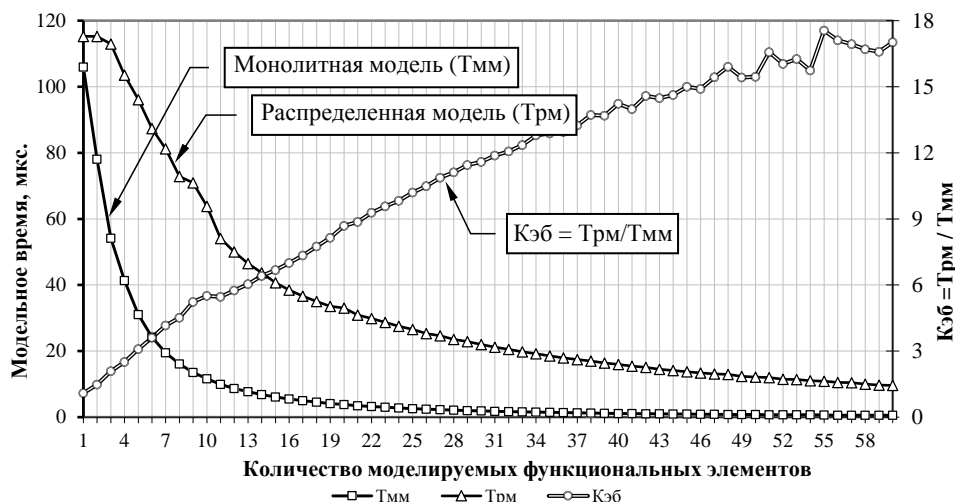


Рис. 7. Влияние размера моделей на модельное время, достижимое ими за 3 мин. астрономического времени их работы

Fig. 7. Influence of the size of models on model time, accessible them in 3 minutes of astronomical time of their work

нент в распределенных моделях, равном 60, эффективность модели по быстродействию снижается с 17 до 13.

Подобная картина наблюдается и в распределенных моделях с количеством уровней более трех. На рисунке 9 показаны гра-

фики изменения модельного времени при $T_f = 180$ с. в зависимости от количества компонент, представленных многоуровневыми распределенными моделями универсального десятичного счетчика с синхронной параллельной записью информации (ИЕ9).

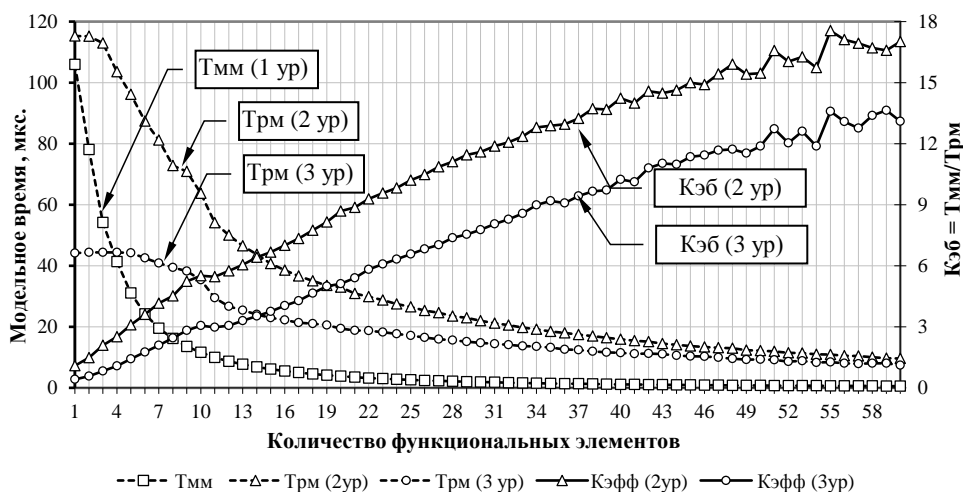


Рис. 8. Изменение модельного времени у распределенных моделей с разным количеством компонент типа TB1 и иерархических уровней ($T_f = 180$ с.)

Fig. 8. Change of model time at the distributed models with different quantity a TB1 component and hierarchical levels ($T_f = 180$ s)

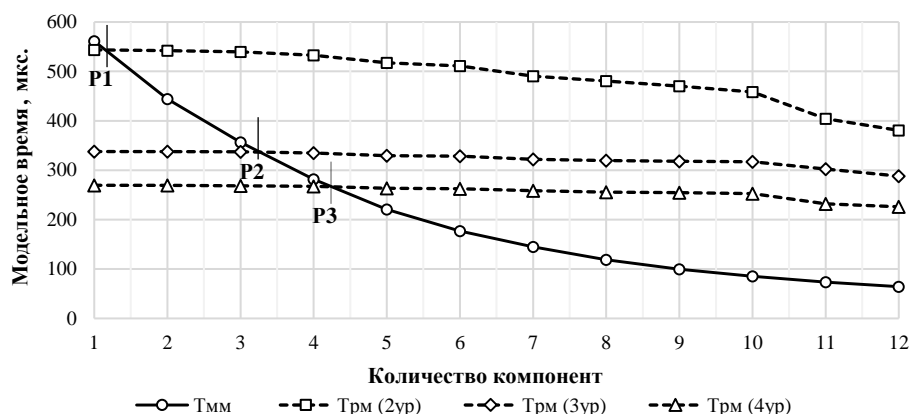


Рис. 9. Изменение модельного времени распределенных моделей в зависимости от количества компонент и иерархических уровней (ИЕ9)

Fig. 9. Change of model time of the distributed models depending on quantity of components and hierarchical levels (ИЕ9)

Как видно из рисунка, модельное время у двухуровневой распределенной модели с одной компонентой почти не отличается от модельного времени в монолитной модели (точка P1). А вот для трех- и четырехуровневых распределенных моделей значение этого показателя заметно ниже. Причем, чем больше количество иерархических уровней у распределенных моделей, тем эта разница становится более заметна. Однако, несмотря на то, что по мере увеличения размеров моделей, модельное время в них уменьшается, но темп этого изменения значительно меньше, чем у монолитных моделей. Это приводит к тому, что при достижении некоторого порогового значения модельное время в них вначале становится равным модельному времени монолитных моделей, а затем становится больше. При дальнейшем увеличении количества компонент в распределенной модели этот разрыв продолжает постоянно увеличиваться. На рисунке 9 эти пороговые значения обозначены соответствующими символами. P1 — для двухуровневой модели, P2 — для трехуровневой и P3 — для четырехуровневой.

Другими словами, при небольших размерах распределенные модели могут быть даже

более медленными, чем их монолитные аналоги ($K_{эб} < 1$). Однако по мере дальнейшего увеличения размера моделей их эффективность вначале сравнивается с эффективностью монолитных моделей ($K_{эб} = 1$), а затем становится выше ($K_{эб} > 1$) и, по мере дальнейшего увеличения размеров, продолжает непрерывно увеличиваться.

На рисунке 10 показаны соответствующие графики изменения показателя эффективности для распределенных моделей с разным количеством иерархических уровней.

Здесь также (как и на рис. 9) показаны пороговые значения размеров у распределенных моделей с разным количеством иерархических уровней. Легко заметить, что показатели эффективности по быстродействию у распределенных моделей имеют не постоянные (зафиксированные) значения, а, по мере увеличения их размеров, также постепенно увеличиваются. После достижения количества компонент в распределенных моделях, равного 12, распределенные модели, в зависимости от количества иерархических уровней, начинают работать в 3,5–6 раз быстрее.

В этом заключается первая особенность проведения иерархических распределенных моделей. Дальнейшие исследования позволили

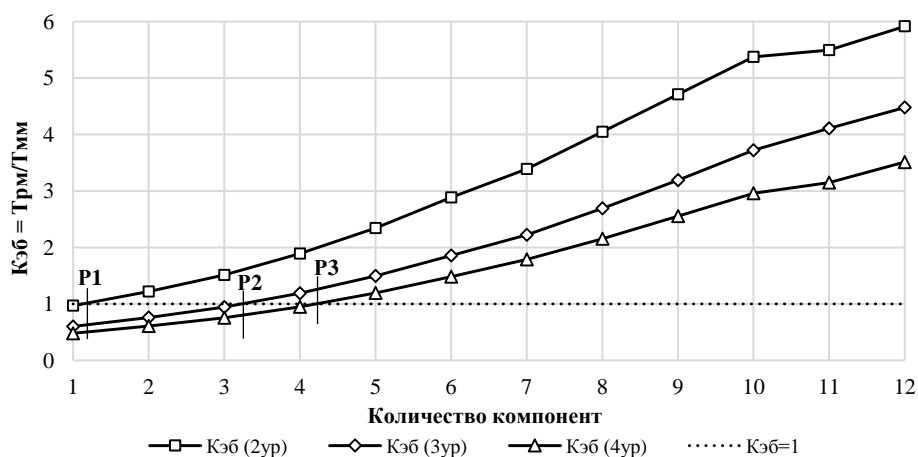


Рис. 10. Изменение показателя быстродействия распределенных моделей в зависимости от количества компонент и иерархических уровней (ИЕ9)

Fig. 10. Change of an indicator of speed of the distributed models depending on quantity of components and hierarchical levels (ИЕ9)

выявить и еще одну важную вторую их особенность. Она состоит в том, что по мере дальнейшего увеличения размеров распределенных моделей (больше 12 компонент), распределенные модели с большим количеством иерархических уровней становятся более эффективными, чем модели с меньшим их ко-

личеством. Для иллюстрации этой особенности на рис. 11 показаны графики изменения коэффициента эффективности по быстродействию у распределенных моделей с разным количеством иерархических уровней при более широком диапазоне изменения их размеров (от 20 до 150 компонент).

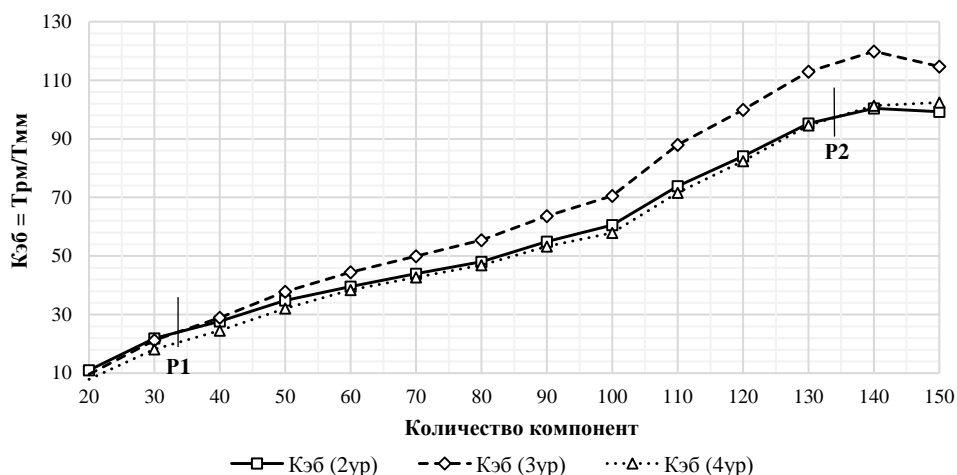


Рис. 11. Изменение показателя быстродействия распределенных моделей в зависимости от количества компонент и иерархических уровней (ИЕ9)

Fig. 11. Change of an indicator of speed of the distributed models depending on quantity of components and hierarchical levels (ИЕ9)

Как видно из рисунка 11, все графики очень похожи друг на друга. Наиболее сильное отличие между ними наблюдается у графика, показывающего зависимость Кэб от размеров трехуровневой модели, в то время как два других графика находятся несколько ниже и почти совпадают.

Если посмотреть на представленные графики в диапазоне изменения их размеров от 1 до 34, то можно обнаружить, что на этом участке наиболее эффективной является двухуровневая распределенная модель, которая находится несколько выше других графиков. Однако, по мере дальнейшего увеличения размеров распределенных моделей, разница между ней и графиком изменения показателя эффективности трехуровневых моделей постепенно сокращается. Однако после достижения некоторого порогового уровня, равного 34 компонентам (обозначен символом P1), их быстродействие становятся одинаковым. При дальнейшем увеличении размера разница между ними начинает постепенно увеличиваться, но уже в пользу трехуровневых моделей.

Похожее поведение наблюдается и у графиков изменения эффективности для двух-

и четырехуровневых распределенных моделей. Только теперь пороговое значение, обуславливающее смену их взаимного преимущества по мере увеличения количества компонент, сдвигается на более высокое значение размера, равное 134 и обозначенное символом P2. Таким образом, наблюдается тенденция, заключающаяся в том, что по мере увеличения размера распределенные модели с большим количеством иерархических уровней становятся более быстродействующими, чем модели с меньшим количеством уровней.

Аналогичным образом ведут себя и распределенные модели, содержащие компоненты, имитирующие поведение других типов функциональных элементов цифровых микросхем. На рисунке 12 приведены графики изменения показателей эффективности по быстродействию распределенных моделей с двумя иерархическими уровнями. Характер изменения этих графиков показывает, что эффективность по быстродействию для распределенных моделей с компонентами разного типа также увеличивается по мере увеличения их количества. Однако темп этого изменения разный и зависит от типа функционального элемента, моделируемого этими

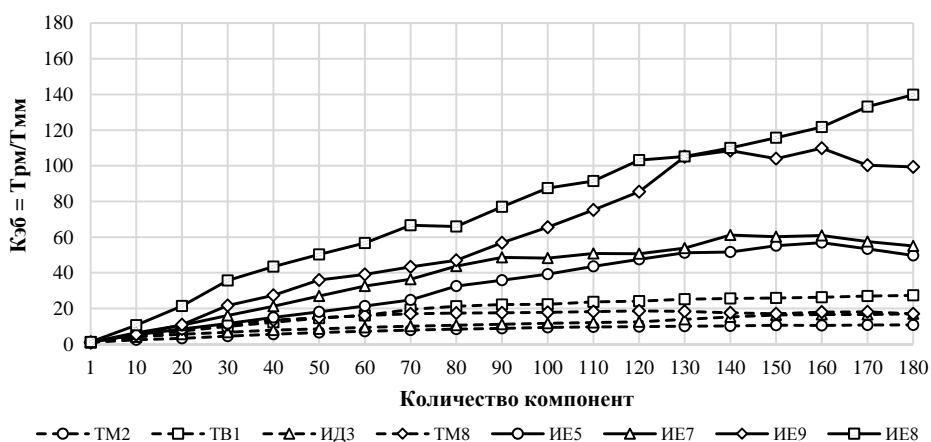


Рис. 12. Зависимости быстродействия двухуровневых распределенных моделей от количества и типа компонент (Тф = 1000 с.)

Fig. 12. Dependences of speed of 2-level distributed models on quantity and type of components (Тф = 1000 s)

компонентами. Дело в том, что, как будет показано ниже, размеры распределенных моделей зависят не только от количества компонент, но и от их размера. Именно этим можно объяснить то обстоятельство, что у распределенных моделей с компонентами небольшого размера (показаны пунктирными линиями) показатели эффективности по быстродействию оказываются заметно меньше, чем у моделей с компонентами большого размера (показаны сплошными линиями).

К простым компонентам относятся компоненты небольшого размера, моделирующие поведение функциональных элементов цифровой техники, таких как ТМ2 (D-триггер с динамическим управлением), ТВ1 (JK-триггер), ИД3 (дешифратор) и ТМ8 (D-триггер с общей установкой в ноль).

При изменении размера распределенных моделей, за счет увеличения количества содержащихся в них компонент малого размера, их максимальная эффективность по быстродействию может достигать значений от 11 до 27. А у распределенных моделей с компонентами большого размера значение этого же показателя уже располагается в диапазоне от 55 до 140.

На рисунке 13 показаны зависимости показателя эффективности по быстродействию у распределенных моделей с тремя уровнями иерархии. Как видно из этого рисунка, приведенные графики показывают несколько большую эффективность трехуровневых распределенных моделей по сравнению с двухуровневыми моделями. Однако в большей мере это касается моделей, состоящих из компонент с более сложной структурой и размерами, показанных сплошными линиями.

И наконец, на рис. 14 представлены графики изменения показателей эффективности у четырехуровневых распределенных моделей. В этом семействе отсутствуют зависимости для моделей, содержащих компоненты с простыми размерами, поскольку их структуры уже не позволяют создавать модели с количеством иерархических уровней (уровней детализации) больше трех.

Основным отличием представленных на рис. 14 графиков от показанных выше является то, что они показывают почти такую же эффективность, как и у моделей с двумя иерархическими уровнями. Тем не менее общий характер изменения графиков на рис. 12–14

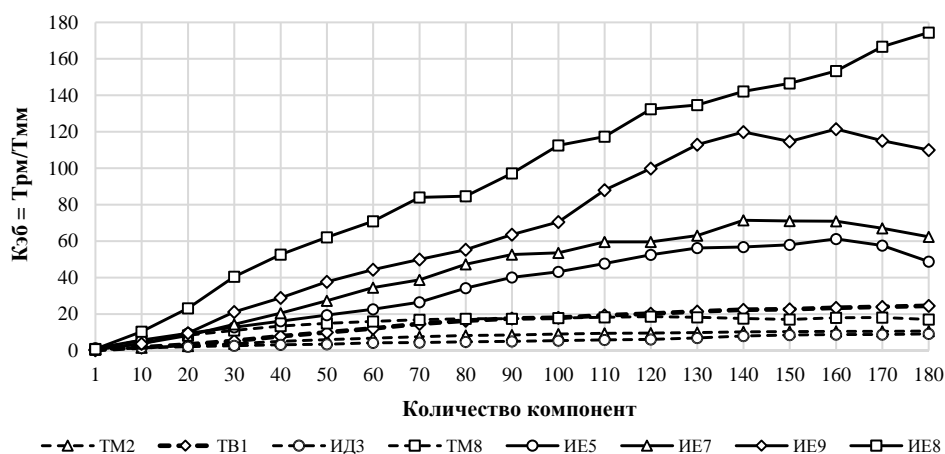


Рис. 13. Зависимости быстродействия трехуровневых распределенных моделей от количества и типа компонент ($T\phi = 1000$ с.)

Fig. 13. Dependences of speed of 3-level distributed models on quantity and type of components ($T\phi = 1000$ s)

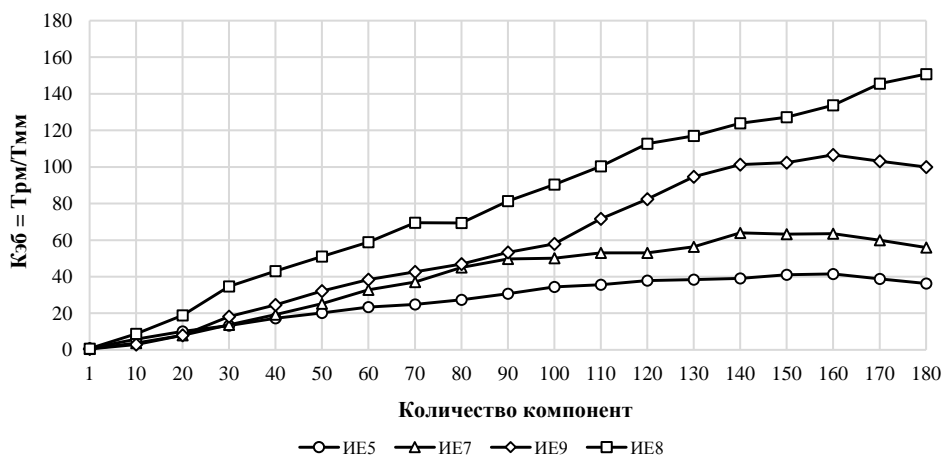


Рис. 14. Зависимости быстродействия четырехуровневых распределенных моделей от количества и типа компонент ($T\phi = 1000$ с.)

Fig. 14. Dependences of speed of 4-level distributed models on quantity and type of components ($T\phi = 1000$ s)

достаточно хорошо сохраняется. Этот характер проявляется в явно выраженной тенденции к непрерывному росту эффективности по быстродействию распределенных моделей при увеличении их размеров. Исключение составляют лишь те графики, которые отображают показатель быстродействия моделей с компонентами, обладающие достаточно простой структурой (ТМ2, ТВ1, ИД3 и ТМ8 на рис. 12 и 13, показанные пунктирными линиями). Характер их изменения отличается тем, что эти графики относительно быстро выходят на некоторый уровень, но при дальнейшем увеличении размера модели либо продолжают оставаться за достигнутым уровне, либо изменяются очень медленно. Это обстоятельство еще раз подтверждает тот факт, что, чем больше размер распределенной модели, тем модели с большим количеством иерархических уровней становятся более быстрыми по сравнению со своими аналогами с меньшим количеством уровней.

Для удобства сравнения (сопоставления) графиков на рис. 12–14, все они показаны в одинаковом диапазоне изменения показателя эффективности по быстродействию от 0 до 180.

Исследование времени загрузки распределенных моделей

Проведенные исследования показали, что на время загрузки распределенных моделей особое влияние оказывают три фактора: количество компонент, размеры компонент и количество иерархических уровней. Вполне очевидно, что независимо от типа модели (монолитная или распределенная) чем больше размер модели, тем большее время потребуется для ее загрузки.

На рисунке 15 представлены графики изменения времени загрузки для монолитных моделей в зависимости от количества компонент, имитирующих поведение разных функциональных элементов.

Первое, что обращает на себя внимание, это то, что все приведенные графики имеют явно выраженный экспоненциальный характер. Другая же их особенность проявляется в том, что наиболее низкими темпами изменяется время загрузки для моделей с компонентами небольшого размера (показаны пунктирными линиями). А наиболее высокими темпами изменяются графики времени загрузки для моделей с компонентами боль-

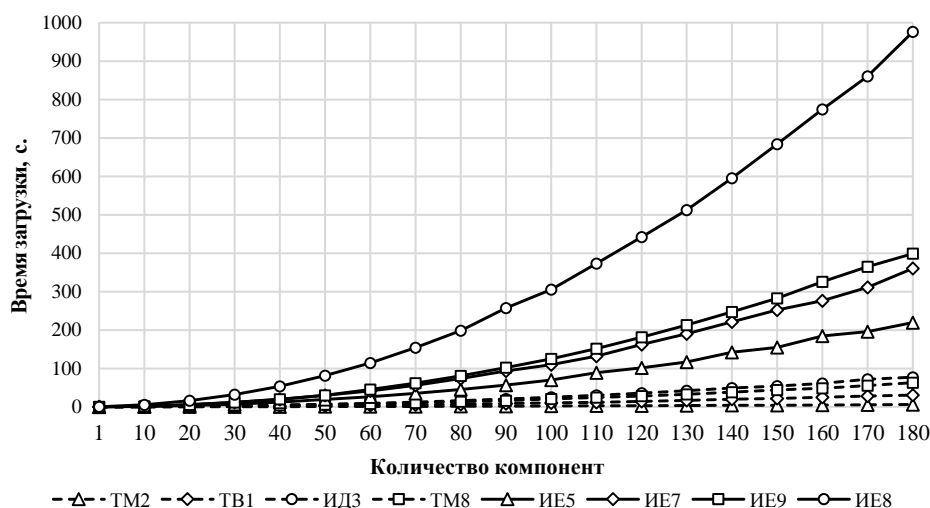


Рис. 15. Влияние количества компонент в монолитных моделях на время их загрузки
Fig. 15. Influence of quantity of components in monolithic models for the time of their loading

шого размера, показанные сплошными линиями.

Как видно из рис. 15, время загрузки у монолитных моделей с компонентами большого размера может достигать значения в 1000 с. (более 16 мин.). Это довольно большой интервал, отрицательно влияющий на эффективность применения инструментального средства имитационного моделирования. Такое большое время загрузки модели отчасти объясняется тем, что она осуществляется в два этапа. Сначала загружается сама модель, а затем ее полное визуальное отображение (см. рис. 6).

Загрузка распределенной модели тоже реализуется в два этапа. Однако ее визуальное изображение отображает уже не всю модель, как в случае с монолитной моделью, а только ту ее часть, которая находится на ее самом верхнем иерархическом уровне (см. рис. 5). При необходимости просмотра модели на более детальном (т. е. на более низком иерархическом уровне) загружается визуальное отображение только той компоненты, поведение которой пользователь хочет рассмотреть более детально. Другими словами, время загрузки распределенной модели всегда будет

заведомо меньше за счет того, что на втором этапе загружается визуальное отображение только той ее части, которая отображает изображение модели на ее самом верхнем иерархическом уровне.

На рисунке 16 приведены графики, показывающие, как изменяется время загрузки монолитных и распределенных моделей, состоящих из компонент, моделирующих поведение триггера типа TM2.

На этом рисунке представлены графики изменения времени загрузки для распределенных моделей, содержащих компоненты одного и того же типа (TM2), но с разным количеством иерархических уровней. Для сравнения здесь же показан и график изменения времени загрузки монолитной модели, представленный пунктирными линиями и обозначенный как «1 ур», указывая тем самым на то, что монолитные модели имеют только один уровень детализации. Как и ожидалось, время загрузки у распределенных моделей меньше, чем у ее монолитного аналога. Но чем больше у них количество иерархических уровней, тем больше и соответствующее время их загрузки.

Следует отметить, что у приведенных на рис. 16 графиках максимальные значения

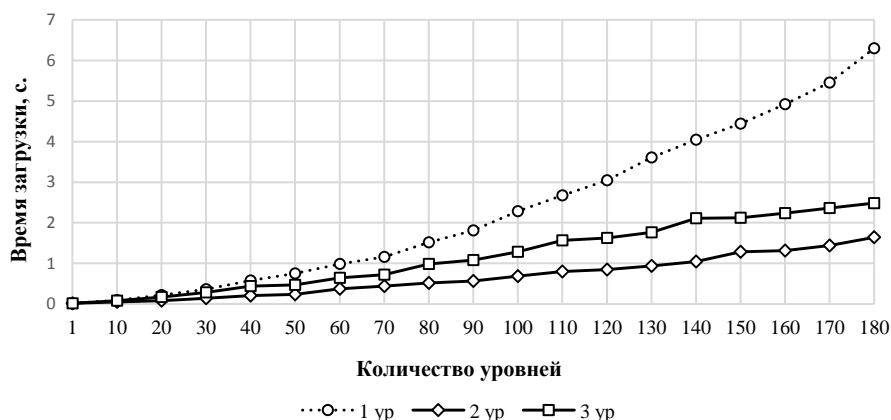


Рис. 16. Влияние размера распределенных моделей с разным количеством иерархических уровней на время их загрузки

Fig. 16. Influence of the size of the distributed models with the different number of hierarchical levels for the time of their loading

времени загрузки моделей находятся в довольно широких пределах, что несколько затрудняет объективную оценку преимущества распределенных моделей по сравнению с монолитными. С этой целью было бы полезнее использовать некоторый независимый показатель, который бы однозначно отображал преимущество (или недостаток) распределенных моделей по отношению к монолитным. С этой целью на рис. 17 приведены графики изменения показателя эффективности времени загрузки распределенных

моделей по сравнению с монолитной моделью (Кэз), представляющего собой отношение времени загрузки монолитной модели (Тзм) ко времени загрузки ее распределенного аналога (Тзмр).

Как видно из этого рисунка, график изменения эффективности загрузки распределенных моделей с двумя иерархическими уровнями заметно выше, чем у монолитных моделей, и может достигать почти в четырехкратного преимущества. А вот у трехуровневой распределенной модели максимальное значение

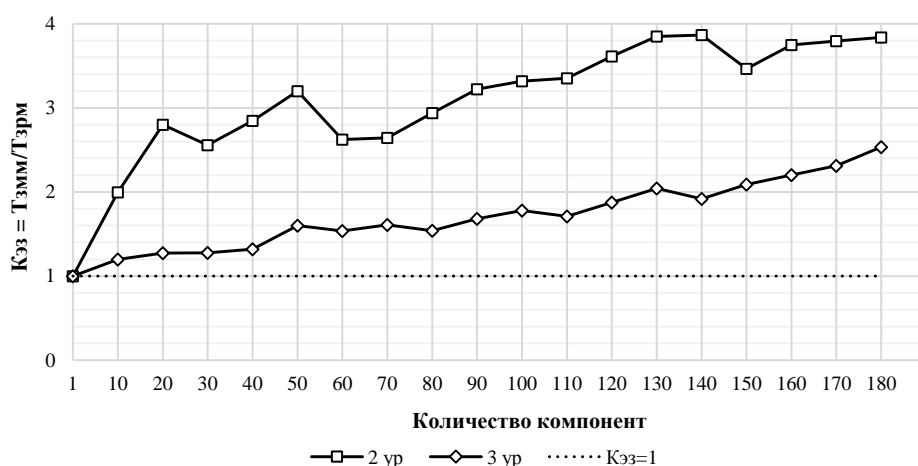


Рис. 17. Влияние количества компонент на эффективность загрузки распределенных моделей (ТМ2)

Fig. 17. Influence of quantity of components on effectiveness of loading of the distributed models (ТМ2)

ние того же показателя снижается уже до значения, равного 2,5. Объясняется это тем, что, как уже говорилось ранее, добавление в распределенную модель нового иерархического уровня приводит к дополнительному увеличению ее размера за счет введения новых представителей подчиненных компонент. А это ведет к снижению быстродействия модели в целом.

Для распределенных моделей с компонентами большего размера зависимости времени загрузки от количества содержащихся в них компонент выглядят уже несколько иначе. Например, на рис. 18 представлены графики зависимости времени загрузки монолитных и распределенных систем с компонентами, моделирующими поведение делителя частоты с переменным коэффициентом деления типа ИЕ8.

Очевидно, что увеличение размера компонент приводит к увеличению размера моделей в целом и, как следствие, должно приводить к увеличению времени их загрузки. Однако, как видно из рис. 18, при макси-

мальном количестве компонент в монолитных моделях время их загрузки увеличилось до 1000 с., а в распределенных моделях этот показатель достигает значительно меньших значений (28, 18 и 10 с.). Так же, как и у распределенных моделей с компонентами небольшого размера (рис. 16) у моделей с компонентами большого размера максимальное достижимое время их загрузки тоже меньше, чем в монолитных моделях и зависит от количества иерархических уровней, причем настолько, что они почти сливаются. Это показывает, что загрузка распределенных моделей осуществляется значительно быстрее, чем это происходит в монолитных моделях.

На рисунке 19 показано, насколько время загрузки монолитных моделей отличается от времени загрузки распределенных моделей.

Как видно из приведенных графиков, по мере увеличения количества компонент эффективность загрузки распределенных моделей непрерывно увеличивается и, в зависимости от количества иерархических уровней, может достигать значений в диапазоне

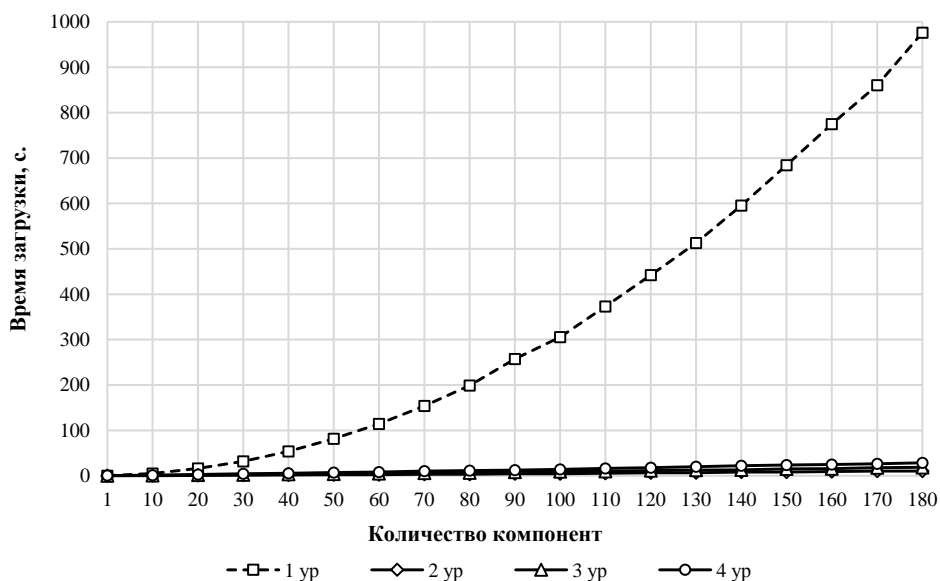


Рис. 18. Влияние количества компонент большого размера (ИЕ8) на время загрузки монолитных и распределенных моделей с разным количеством иерархических уровней

Fig. 18. Influence of quantity of components of the big size (IE8) for the time of loading of monolithic and distributed models with the different number of hierarchical levels

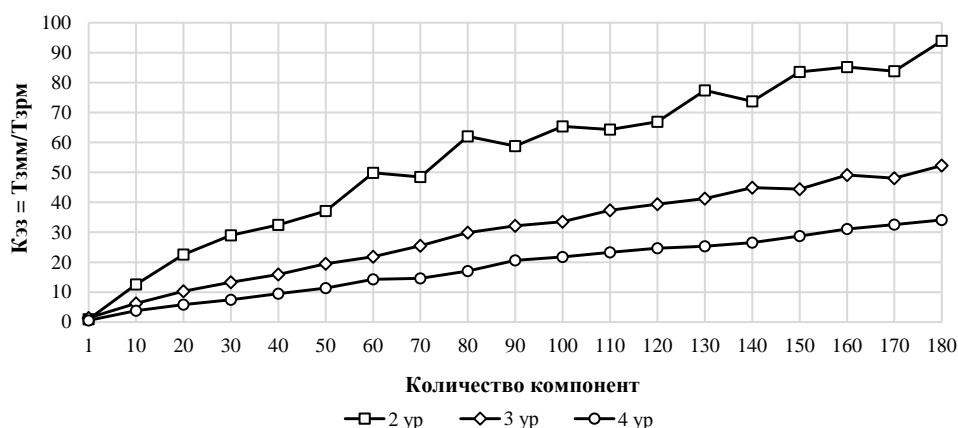


Рис. 19. Зависимости показателя эффективности по загрузке распределенных моделей с разным количеством иерархических уровней от количества компонент

Fig. 19. Dependences of an efficiency indicator on loading of the distributed models with the different number of hierarchical levels from quantity of components

от 34 до 94 раз. Например, для двухуровневых моделей этот показатель может достигать значения, равного 94. А у распределенных систем с тремя уровнями оно несколько снижается до значения чуть более 52. И наконец, у распределенных систем с четырьмя уровнями этот показатель снижается уже до 34.

Как изменяется эффективность загрузки распределенных моделей по сравнению с загрузкой монолитных моделей, содержащих компоненты других типов, в зависимости

от их количества и количества иерархических уровней, показано на рис. 20–22. Так же, как и ранее, на этих рисунках пунктирными линиями выделены зависимости для распределенных моделей с достаточно простыми компонентами, обладающими небольшими размерами. Для более удобного их визуального сопоставления все графики показаны в одинаковом диапазоне изменения показателя эффективности загрузки, характерном для двухуровневой распределенной модели.

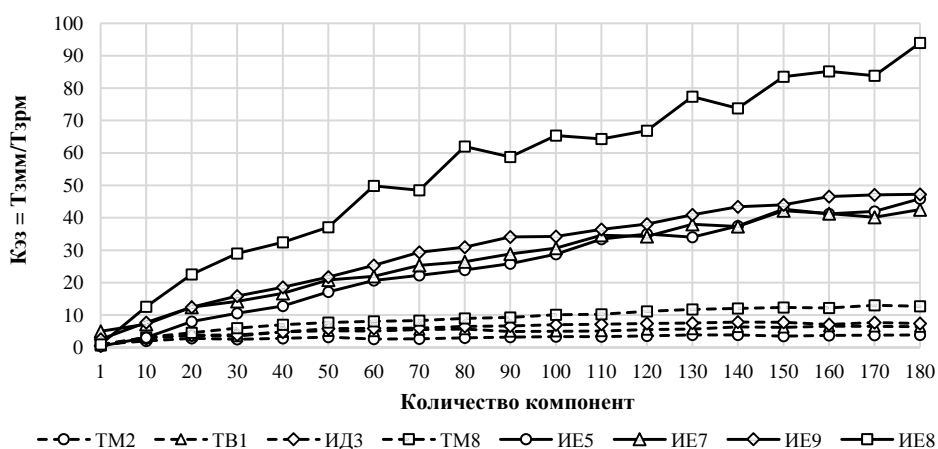


Рис. 20. Влияние количества компонент разных типов на эффективность загрузки двухуровневых моделей

Fig. 20. Influence of quantity of components of different types on effectiveness of loading of 2-level models

Из рисунка 20 видно, что эффективность загрузки двухуровневых моделей зависит не только от их общего размера, но и от размера компонент и количества иерархических уровней. Модели, состоящие из компонент малого размера, оказываются самыми малоэффективными. Их эффективность по загрузке может достигать значений только в диапазоне от 6,5 до 13 раз. Эффективность же распределенных моделей, содержащих компоненты большого размера, заметно выше и может достигать значений в диапазоне от 42 до 94.

На рисунке 21 приведены графики изменения показателя эффективности по загрузке распределенных моделей уже с тремя иерархическими уровнями. Из этого рисунка видно, что отображаемые ими зависимости в общем сохранили свой характер, но темп их изменения несколько уменьшился. Это привело к тому, что максимально достижимая эффективность по загрузке трехуровневых распределенных моделей уменьшилась до значений от 4,6 (вместо 6,5 у двухуровневых моделей) до 52 (вместо 94 у распределен-

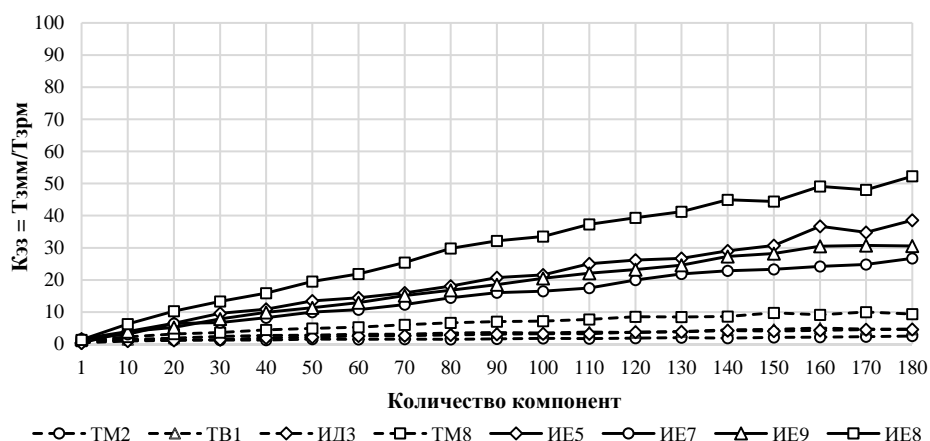


Рис. 21. Влияние количества компонент разных типов на эффективность загрузки трехуровневых моделей

Fig. 21. Influence of quantity of components of different types on effectiveness of loading of 3-level models

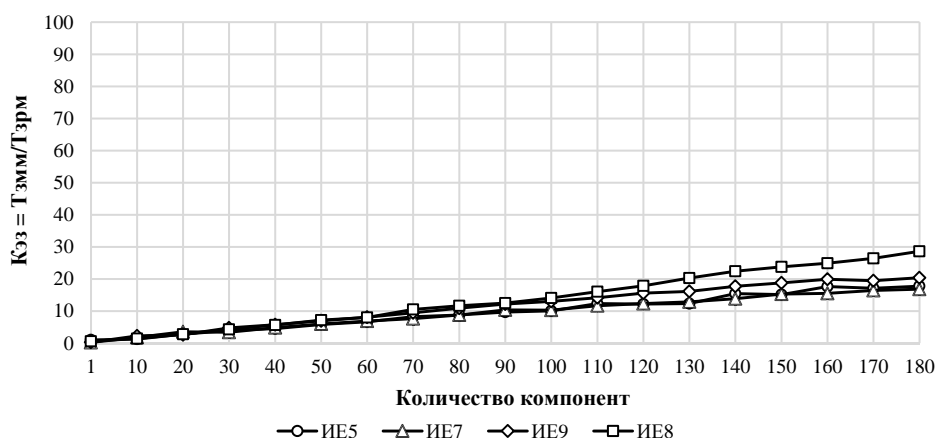


Рис. 22. Влияние количества компонент разных типов на эффективность загрузки четырехуровневых моделей

Fig. 22. Influence of quantity of components of different types on effectiveness of loading of 4-level models

ных моделей с двумя уровнями), то есть более, чем в 1,5 раза.

На рисунке 22 отсутствуют графики для распределенных систем, содержащих компоненты малого размера. Это, как уже говорилось, связано с невозможностью представления компонент с малыми размерами и, что еще более важно, простой структурой и с большим количеством иерархических уровней. Оставшиеся же графики показывают ту же тенденцию, которая была выявлена при анализе распределенных моделей с тремя иерархическими уровнями (рис. 21). В данном случае максимально достижимая эффективность загрузки моделей снизилась до еще более низких значений от 17 до 29.

Анализ зависимостей, приведенных на рис. 20–22, позволяет сделать вывод о том, что распределенные модели обеспечивают заметно меньшее время загрузки, чем у монолитных моделей. Причем для моделей большого размера это преимущество может достигать значений в несколько десятков раз.

Размеры моделей

Как показано в предыдущих разделах, оценка эффективности распределенных моделей проводилась в зависимости от их размеров, определяемых как количество компо-

нент в их составе. Но это очень приближенная оценка, поэтому в качестве более точной оценки размера моделей была использована оценка в виде суммарного количества базовых элементов сети Петри, таких как позиции, переходы и связи между ними. Это позволило установить более точную оценку размера моделей и показать, что их размер зависит не только от количества компонент на их верхних иерархических уровнях, но и от размера самих компонент и количества представляемых ими иерархических уровней (уровней детализации).

На рисунке 23 приведены результаты измерения размера моделей, имитирующих поведение компонент разного типа, и с разным количеством иерархических уровней. Как видно из диаграмм, представленных на этом рисунке, модель компоненты, имитирующей поведение функционального элемента триггера типа ТМ2, имеет самые маленькие размеры. А наибольшими размерами среди рассмотренных компонент обладает компонента, имитирующая поведение функционального элемента делителя частоты с переменным коэффициентом деления типа ИЕ8.

В дальнейшем будем ссылаться на подобные компоненты, соответственно, как на компоненту типа ТМ2 и компоненту типа ИЕ8.

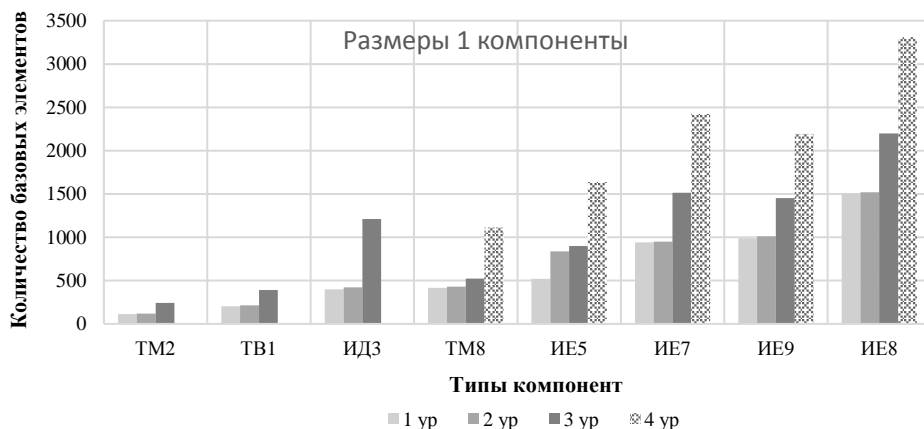


Рис. 23. Размеры компонент в зависимости от их типа и количества иерархических уровней

Fig. 23. The sizes of components depending on their type and the number of hierarchical levels

Поскольку функциональные модули каждой из компонент, в свою очередь, также могут быть представлены как распределенная модель, для каждой из них отдельными столбиками показаны размеры, при разных количествах иерархических уровней. Компоненты, представленные как монолитные модели, показаны как модели с одним иерархическим уровнем. Вполне очевидно, что увеличение количества компонент одного типа в монолитных или в распределенных моделях будет приводить к линейному увеличению их размера в соответствии со следующим выражением:

$$S = K * n + S_0;$$

где S — размер модели;

K — коэффициент пропорциональности (темп изменения размера), равный размеру компоненты, используемой в составе модели; n — количество компонент на верхнем уровне; S_0 — некоторое начальное значение, равное размеру, например, генератора входных воздействий (см. рис. 5).

Как и следовало ожидать, увеличение количества иерархических уровней в компонентах должно приводить и к увеличению их размеров. Однако, как показали исследования,

это увеличение носит уже явно выраженный нелинейный характер. На рисунке 24 приведены диаграммы, в которых цифры над столбцами указывают, насколько размеры компонент с разным количеством иерархических уровней отличаются от размеров тех же компонент, представленных монолитными моделями. Анализ этих диаграмм показывает, что увеличение иерархических уровней в распределенных моделях может приводить к увеличению их размера в 2,5–3 раза.

По-видимому, такое нежелательное увеличение размера и является той ценой, которую приходится платить за более высокую скорость выполнения распределенных моделей и уменьшение времени их загрузки.

Заключение

Как показали проведенные исследования, использование распределенных моделей значительно более эффективно по сравнению с их монолитными аналогами. При этом следует заметить, что быстроедействие распределенных моделей увеличивается не на десятки процентов и не в несколько раз, а более чем в 100 раз. Причем чем больше размер моделей, тем больше быстроедействие распределенных моделей. Например, трехуровне-

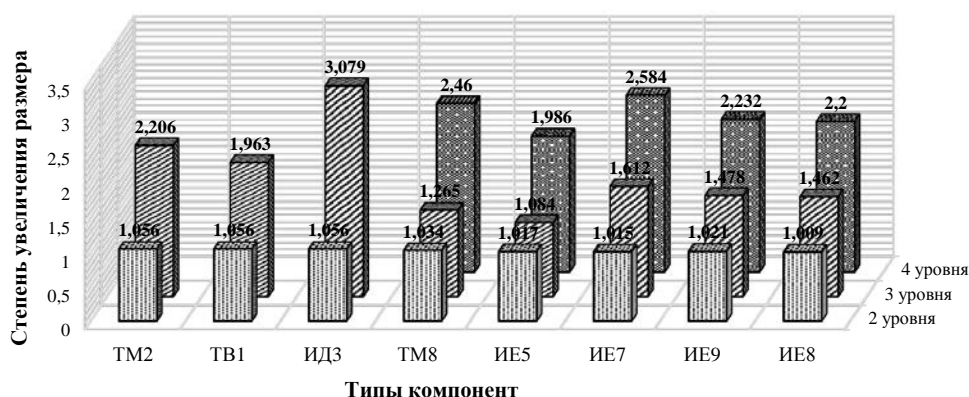


Рис. 24. Степень увеличения размеров распределенных моделей в зависимости от типа компонент и количества иерархических уровней

Fig. 24. Extent of increase in the sizes of the distributed models depending on type of components and the number of hierarchical levels

вая распределенная модель со 180 компонентами, моделирующими поведение функциональных элементов типа ИЕ8, работает почти в 200 раз быстрее, чем ее монолитный аналог. И это еще не предел. Процессы загрузки распределенных моделей осуществляются в несколько десятков раз быстрее, чем это происходит у монолитных моделей. Однако все указанные преимущества распределенных моделей перед монолитными достигаются за счет почти 2–3-кратного увеличения их размеров.

Результаты проведенных исследований явно указывают на то, что переход к распределенным моделям позволит существенно расширить область практического применения имитационных моделей, и надеемся, что подобные результаты могут быть получены для моделей, построенных не только на основе сетей Петри, но и на базе использования других математических аппаратов.

Следует отметить, что проведенные исследования были бы невозможны без помощи инструментального средства, позволяющего создавать и редактировать модели цифровых устройств электронной техники и проводить с ними экспериментальные исследования [9]. В процессе выполнения данной работы этот инструмент показал очень хорошую работоспособность, он прост и удобен и может быть рекомендован для более широкого использования. Однако на данный момент его все-таки следует рассматривать, скорее всего, как удачный опытный образец или рабочий макет. Для доведения его до уровня промышленного образца или коммерческого продукта, по-видимому, все же потребуются некоторая дополнительная работа.

Список литературы

1. *Fujimoto R. M.* Parallel and Distributed Simulation Systems. USA: Wiley, 2000. — 300 p.
2. *Таненбаум Э., ван Стен М.* Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. — 877 с.
3. *Бродский Ю. И.* Распределенное имитационное моделирование сложных систем. М.: ВЦ РАН, 2010. — 156 с.
4. *Jensen K., Christensen S., Wells L.* Coloured Petri Nets and CPN Tools for Modeling and Validation of Concurrent Systems // *Intern. J. on Software Tools for Technology Transfer*. 2007. Vol. 9. P. 213–254.
5. *Окольнишников В. В.* Разработка системы распределенного имитационного моделирования для различных операционных сред // Сборник докладов Второй Всероссийской научно-практической конференции по имитационному моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» (ИММОД 2005). Санкт-Петербург, 2005. — Т. 1. — С. 256–260.
6. *Смелянский Р. Л., Бахмутов А. Г., Костенко В. А.* Среда моделирования DYANA: синтез, анализ и оптимизация вычислительных систем реального времени // Сб. докл. I Междунар. конф. «Цифровая обработка сигналов и ее применения». МЦНТИ. 1998. Т. 4. С. 152.
7. *Веселов А. А.* Моделирование устройств цифровой электронной техники D-расширением сетей Петри. Монография. Тверь: ТГТУ, 2006. С. 103.
8. *Веселов А. А.* Распределенная модель устройств цифровой и вычислительной техники на основе сетей Петри // Интернет-журнал «Науковедение». Т. 7. № 3. 2015. М.: Науковедение, 2015. URL <http://naukovedenie.ru/PDF/124TVN315.pdf>.
9. *Веселов А. А.* Система распределенного моделирования устройств цифровой электроники на основе сетей Петри. «Приборы и системы. Контроль, Измерение, Диагностика». 2016. № 2. С. 11–22.

References

1. *Fujimoto R. M.* Parallel and Distributed Simulation Systems. USA, Wiley, 2000, 300 p.
2. *Tanenbaum Andrew S., van Steen Maarten.* Distributed systems. Principles and Paradigms. Saint-Peterburg, Piter, 2003, 877 p.
3. *Brodskey U. I.* *Raspredeleennoe imitacionnoe modelirovanie slozhnih sistem.* [The distributed simulation modeling of difficult systems]. Moscow, Computer Center of the Russian Academy of Sciences, 2010, 156 p.
4. *Jensen K., Christensen S., Wells L.* Coloured Petri Nets and CPN Tools for Modeling and Validation of Concurrent Systems. *Intern. J. on Software Tools for Technology Transfer*, 2007, vol. 9, pp. 213–254.
5. *Okolnishnikov V. V.* *Razrabotka sistemi raspredelennogo imitacionnogo dlia razlichnih operacionnih sred* [Development of the system of the distributed

- simulation modeling for different operating environments]. *Sbornic dokladov Vtoroi Vserossiiskoi nauchno-prakticheskoi konferencii po imitacionmu modelirovaniu i ego primeneniu v nauke i promishlennosti «Imitacionnoe modelirovanie». Teoria i practica* [The Collection of reports of the Second All-Russian scientific and practical conference on simulation modeling and its application in science and the industry «Simulation modeling. Theory and practice»] (IMMOD 2005). Saint-Petersburg, 2005, vol. 1, pp. 256–260.
6. Smelianskii R. L., Bahmurov A. G., Kostenko V. A. *Sreda modelirovaniya DYANA: sintez, analiz i optimizatsia vychislitel'nykh sistem realnogo vremeni* [Environment of simulation DYANA: synthesis, analysis and optimization of computing systems of real time]. *Sbornic dokladov I mezdunarodnoi konferencii «Cifrovaia obrabotka signalov i ee primeneniia»* [Collection of reports of I International conference «Digital signal processing and its applications»]. MCNTI, 1998, vol. 4, p. 152.
 7. Veselov A. A. *Modelirovanie ustroistv cifrovoi elektronnoi tekhniki D-rasshireniem setei Petri. Monographia* [Simulation of devices of a digital electronic equipment D-extension of Petri nets. Monograph. Tver. TSTU]. Tver, TGTU, 2006, p. 103.
 8. Veselov A. A. *Raspredelelnaia model ustroistv cifrovoi i vychislitel'noi tekhniki na osnove setei Petri*. [The distributed model of devices of a digital and ADP equipment on the basis of Petri nets]. *Inert-gurnal «Naukovedenie»* [The Online magazine «Naukovedeniye»]. 2015, vol. 7, no. 3. Moscow, Science of science. Available at: <http://naukovedenie.ru/PDF/124TVN315.pdf>.
 9. Veselov A. A. *Sistema raspredelenogo modelirovaniia ustroistv cifrovoi elektroniki na osnove setei Petri* [The system of the distributed simulation of devices of digital electronics on the basis of Petri nets]. *Pribori i sistemi. Kontrol, Izmerenie, Diagnostika* [Instruments and systems. Monitoring, Measurement, Diagnostics]. 2016, no. 2, pp. 11–22.

A. Veselov, Department of Electronic Computer, Tver State Technical University, Tver, Russia, evm.tstu@gmail.com

Experimental studies of a distributed model based on Petri nets

The paper presents the results of experimental studies of the effectiveness of the application of distributed models in comparison with their monolithic analogs. As a basic model, models based on the mathematical apparatus of Petri nets are used. In particular, special expansion of ordinary, temporary Petri nets with inhibitory arcs, oriented to modeling of digital devices of automation and computer facilities. A description of the indicators used to evaluate the effectiveness of monolithic and distributed models is given. The advantages and disadvantages of these types of models are discussed. The short description of this model, indexes for assessment of effectiveness and conditions of carrying out an experiment is given. Analysis of effectiveness of the distributed models allowed to select two feature. The first is that, in process of increase in their size, the efficiency factor of the distributed models continuously increases. At the same time, models with the smaller number of hierarchical levels are more efficient. The second feature is that, in process of further increase in the sizes of the distributed models, the effectiveness on speed of models with a large number of hierarchical levels becomes not only equal, but to exceed effectiveness of models with their smaller quantity.

It is shown, that high-speed performance of the distributed models can be two orders more, and time of their loading is tens times less, than similar indices at monolithic models.

Keywords: distributed and monolet model, Petri nets, component, functional module, proxy, hierarchical reflection, effectivity by speed, loading time, models size.

About author:

A. Veselov, Dr of Technique

For citation:

Veselov A. Experimental studies of a distributed model based on Petri nets. *Prikladnaya Informatika — Journal of Applied Informatics*, 2018, vol. 13, no. 5 (77), pp. 123–143 (in Russian).