

ЛАТВИЙСКАЯ АКАДЕМИЯ НАУК

**АВТОМАТИКА
И
ВЫЧИСЛИТЕЛЬНАЯ
ТЕХНИКА**

1994 МАРТ-АПРЕЛЬ 2

НАУЧНО - ТЕОРЕТИЧЕСКИЙ ЖУРНАЛ
ВЫХОДИТ 6 РАЗ В ГОД С ЯНВАРЯ 1967 г.

РИГА
ИНСТИТУТ ЭЛЕКТРОНИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

УДК 681.326.7

А. И. Андрюхин

МЕТОД ПАРАЛЛЕЛЬНОГО ЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ ИСПРАВНЫХ И НЕИСПРАВНЫХ ЦИФРОВЫХ УСТРОЙСТВ НА IBM PC

На этапе функционально-логического проектирования БИС создается ее функционально-логическая схема, т.е. БИС представляется на уровне базовых логических и последовательностных элементов типа И-НЕ, ИЛИ-НЕ, НЕ, триггер и т.д. На этом же этапе разрабатывается ее тестовое обеспечение. Повышение уровня интеграции современной цифровой электроники предъявляет высокие требования к параметрам инструментальных компьютеров, на которых осуществляется логическое моделирование проектных решений, ставшее неотъемлемой частью при проектировании и тестировании дискретных устройств.

Логическое моделирование, выполняемое на компьютерах «фон-неймановской» архитектуры, является принципиально медленным процессом, так как параллельные изменения в схеме имитируются последовательно. В связи с этим насущной является проблема повышения быстродействия при увеличении размеров исследуемых устройств. В настоящее время существуют следующие направления решения этой проблемы: а) построение специализированных машин логического моделирования и соединение программных и аппаратных средств [1]; б) повышение уровня моделирования путем использования многоуровневого языка описания цифровых устройств (язык VHDL [2]); в) использование аппаратных особенностей инструментальных компьютеров в моделирующих программах.

Специализированные машины логического моделирования не всегда доступны и обладают рядом ограничений. Моделирование с использованием языка VHDL является перспективным путем разрешения этой проблемы, однако и здесь имеются свои трудности. К ним можно отнести значительные затраты труда программистов; определение адекватности построенной программной модели по вход-выходным характеристикам реальному объекту; оценка качества теста, построенного с использованием этих моделей. Одним из хорошо известных способов ускорения логического моделирования является параллельное моделирование, основой которого является возможность одновременно и независимо выполнять логические операции над разрядами машинного слова — обычно в 2, 4 или в случае ЕС ЭВМ над полем — в 256 байт.

В [3,4] предложено использовать структурные особенности списка машинных команд серии IBM 370 (и совместимых с ними) для ускорения моделирования определенного типа логических схем. Здесь же приведены оценки производительности описанного способа моделирования.

Аналогично способу [3,4] предлагается выполнять моделирование на персональных компьютерах IBM, используя их возможности работы с видеопамятью.

Предварительно напомним, что экран дисплея персонального компьютера (ПК) представляется как набор отдельных точек, образующих прямоугольный растр. Число точек определяет разрешающую способность графической системы. Она отражается парой чисел, первое из которых показывает число точек в строке, а второе — число строк. Каждой

точке соответствует определенное число битов (атрибут точки) в видеопамяти ПК. Атрибут состоит из 1, 2, 4, 8 битов в зависимости от графического режима. Видеопамять является частью видеоадаптера — специального устройства, управляющего дисплеем.

Основой описываемого метода параллельного моделирования является наличие аппаратно реализованных логических битовых операций над содержимым поля оперативной памяти и буфера видеопамяти, результат которых заносится в буфер [4-7]. В число этих операций входят операции логического умножения, сложения, отрицания, «исключающего ИЛИ». Для использования их достаточно понимания работы функций putimage и getimage при программировании на языке Си [6]. Параметрами этих функций являются координаты прямоугольной области на дисплее и тип логической операции. Биты поля видеопамяти, соответствующей этой области, рассматриваются как значения выходов элементов устройства. Таким образом с помощью одного действия вывода на экран можно выполнять логическую операцию над полями памяти компьютера длиной в десятки килобайт.

Отличительной чертой метода является параллельная обработка элементов устройства, имеющих одинаковый тип, а не данных, относящихся к одной линии схемы, что обычно имеет место при параллельном моделировании [8,9]. Для этого схема должна быть представлена в виде соединений одновыходных элементов ограниченного числа типов вида И-НЕ, ИЛИ-НЕ и т.д., которые будем считать базовым. Базовым элементом может быть широко используемый в цифровых устройствах D-триггер типа защелки и другие триггеры, если выделять несколько полей под выходы, входы и состояния элементов. В принципе любой элемент, функционирующий которого описывается булевыми функциями, может быть базовым. Для простоты рассматриваем в качестве базовых элементов стандартный функционально полный набор двухвыходных вентилей И, ИЛИ, И-НЕ и т.д.

Дискретное устройство представляется при моделировании своим структурным описанием, включающим список компонентов и список связей между ними. Проверку правильности структурного описания осуществляет семантико-синтаксический анализатор. Для представления устройства на вентильном уровне необходима программа расширения — экспандер. Ее основной функцией является получение расширенного структурного описания устройства из описания на более высоком уровне путем «раскрытия» элементов устройства (представления их на более низком уровне). Иерархическая компилятивная система моделирования (ИКСМ) [10] включает в себя семантико-синтаксический анализатор, экспандер и как автономную часть — реализованный метод.

Моделирование схем, при котором возможна параллельная обработка их элементов, опишем следующим образом. Сначала рассмотрим комбинационные устройства. Пусть схему составляют L базовых элементов, где $L = \sum L_i$ и L_i — число элементов типа i ($i = 1, K$). Обозначим через W поле оперативной памяти машины длиной $\sum (\lfloor L_i / 8 \rfloor + 1)$ байтов, в которых хранятся значения выходов всех L элементов. При этом выходы элементов одинакового типа хранятся вместе. Через W_1, W_2, \dots, W_k, R обозначим рабочие поля, в которых из поля W согласно таблицам связи Q_1, Q_2, \dots, Q_k формируем значения входов $1, 2, \dots, k$ элементов типа i . Значение элемента $Q_j(t)$ есть номер элемента, выход которого связан с j -м входным контактом элемента i . Длина полей W_k ограничена $\max (\lfloor L_i / 8 \rfloor + 1)$.

Алгоритм моделирования комбинационных устройств опишем следующим образом.

1. Установить новые значения внешних входов и установить $F = 0$.
2. Установить тип базового элемента $i = 0$.
3. Вычислить $i = i + 1$.
4. Если i больше K , перейти к п.9.
5. Сформировать поля W_1, W_2, \dots, W_k из поля W по Q_1, Q_2, \dots, Q_k .
6. Вычислить значения выходов элементов типа i в поле R .

7. Если R не равно W_i , установить $F = 1$.

8. Перейти к п.3.

9. Если $F = 1$, вычислить $W = R$ и перейти к п.2.

10. Если конец входных воздействий, то выход из алгоритма, иначе — п.1.

Алгоритм моделирования последовательностных устройств должен учитывать возможность состязаний при определении выходных реакций и необходимость введения многозначных алфавитов (хотя бы троичного алфавита [8,9]). Необходимость многозначного моделирования влечет за собой введение дополнительных областей памяти для хранения значений компонент многозначных сигналов.

Рассмотрим троичный алфавит со следующей кодировкой сигналов $0 = (1,0)$, $1 = (0,1)$ и $x = (0,0)$. Построив таблицу значений операции И-НЕ, инверсную табл.4.1 в работе [4], можем определить из нее, что значения компонент $C = (C1, C2)$ для $C = \neg(AB)$, где $A = (A1, A2)$ и $B = (B1, B2)$, выражаются булевыми функциями $C1 = A2 \cdot B2$ и $C2 = A1 \vee B1$. Здесь \neg, \cdot, \vee — обозначения булевых операций отрицания, умножения и сложения. Следовательно, над полями значений компонент поля элементов И-НЕ выполняем вычисления по этим формулам согласно алгоритму до совпадения значений на двух соседних итерациях.

Аналогичные выражения могут быть получены и для других элементов. При N -значном моделировании l -е биты полей W^1, W^2, \dots, W^n представляют собой многозначное значение выхода l -го элемента, где $W = W^1, W^2, \dots, W^n$ и n — наименьшее целое число, не меньшее $\log_2 N$. Любое рабочее поле аналогично W состоит из n компонент. Необходимо учитывать возможность генерации сигналов на линиях устройства. Будем ограничивать число итераций достаточно большим пределом H_{\max} . Оно естественно зависит от глубины схемы, числа обратных связей, их значений, а также значений входных сигналов. По достижении H_{\max} будем устанавливать значения на несовпадающих линиях равными x . Для нашего способа кодирования это означает вычисление $C1 = A1 \cdot B1$ и $C2 = A2 \cdot B2$.

С учетом сказанного можем определить алгоритм моделирования в следующем виде.

1. Если конец входных воздействий, то выход алгоритма, иначе установить новые значения внешних входов и установить $F = 0$.

2. Установить номер итерации $T = 0$.

3. Установить тип базового элемента $i = 0$.

4. Вычислить $i = i + 1$.

5. Если i больше K , то вычислить $T = T + 1$ и перейти к п.10.

6. Сформировать поля входов W_1, W_2, \dots, W_k из поля W по Q_1, Q_2, \dots, Q_k .

7. Вычислить значения выходов элементов типа i в поле R_i .

8. Если R_i не равно W_i , установить $F = 1$.

9. Перейти к п.4.

10. Если $F = 0$, перейти к п.1.

11. Если номер итерации больше заданного H_{\max} , то установить несовпадающие линии полей W_i в неопределенное состояние и перейти к п.1.

12. Вычислить $W = R$ и перейти к п.3.

Определим объем памяти компьютера для моделирования схем, состоящих из L двухвходовых элементов. Для комбинационных схем необходимо иметь четыре поля (W, R, W_1, W_2) длиной по $\lfloor L/8 \rfloor + 1$ байтов и две таблицы связей (Q_1, Q_2) длиной по $4L$ байтов (выделяем для номера элемента четыре байта, хотя можно ограничиться и тремя байтами). Для комбинационных устройств максимальное число вентилях в схеме ограничено выражением $C/8,5$, где C — свободная память компьютера (без DOS и моделирующей

программы). Для объема памяти в 640 Кбайт имеем возможность моделировать около 60 тысяч элементов.

Для последовательных устройств ввиду необходимости введения N -значного алфавита следует иметь $4n$ рабочих полей и таблицы связей Q_1 и Q_2 . Число вентилях ограничено $C/(8 + n/2)$. При свободной памяти 15 Мбайт имеем теоретическую возможность моделировать около двух миллионов вентилях. Практически выполнять это затруднительно, так как скорость моделирования при рабочих частотах современных IBM PC является низкой для меньших схем, что видно из Табл. 1.

Т а б л и ц а 1.

Номер схемы	Число вентилей	Число входов	Время мо- делирования (мин/с)	Среднее число итераций	Время мо- делирования	Среднее число итераций	M
			$M=1$				
1	592	40	9 с	8	1,5 с	14	480
2	1076	40	18 с	11	4,7 с	23	480
3	2208	40	38 с	10	9,7 с	23	240
4	4096	80	1 мин 20 с	12	15 с	19	120
5	8160	80	3 мин	13	34 с	21	56
6	16416	80	6 мин 40 с	15	1 мин 18 с	21	24
7	24544	120	9 мин 26 с	14	2 мин 47 с	18	8
8	57344	180	21 мин	14	—	—	—
9	576	40	16 с	11	6,7 с	40	480
10	1076	40	43 с	17	10,4 с	40	448
11	2016	40	1 мин 40 с	11	25 с	40	216
12	4080	80	3 мин 8 с	19	56 с	40	104
13	8160	80	7 мин 8 с	23	2 мин 3 с	40	40
14	16416	80	15 мин	24	9 мин 22 с	36	8
15	20448	120	15 мин	19	11 мин	40	8
16	53248	180	28 мин 20 с	15	—	—	—

Можно легко повысить быстродействие описанного метода моделирования, используя параллельное моделирование на входных наборах [9]. Для последовательных устройств это возможно, если базовыми элементами являются комбинационные компоненты. Заметим, что компонент сигнала при расчетных оценках занимал 1 бит. Построение полей W_k из поля W , связанное с выделением и пересылкой битов значений сигналов по таблицам Q , требует много времени. Чтобы увеличить быстродействие предложенного метода необходимо отводить под компонент сигнала M битов (для эффективности и простоты программной реализации желательно, чтобы M было целым числом байтов). При этом уменьшается объем моделируемых схем, но существенно повышается быстродействие. Таким образом, при M , большем 1, осуществляется и параллельное моделирование элементов устройства, и параллельное моделирование устройства на M наборах.

В табл. 1 представлены результаты пробных расчетов. Моделирование проводилось на IBM PC AT/386 с тактовой частотой 40 МГц. Число входных наборов равнялось 100. Схемы 1-8 были комбинационными и моделировались в двоичном алфавите. Последова-

тельные схемы, начиная с девятой, моделировались в троичном алфавите. Число итераций ограничивалось 40.

Можно сделать вывод, что описанный метод моделирования является сплошным методом моделирования, использующим аппаратные особенности IBM PC. Характеристики его быстродействия высоки, он позволяет обрабатывать большие логические схемы. Значение скорости моделирования обратно пропорционально числу итераций и числу вентилях. Топология устройства влияет на скорость моделирования увеличением или уменьшением числа итераций, которое зависит также от значения входных сигналов. Подкупает простота программной реализации. Легко реализуется моделирование в различных алфавитах. Недостатком метода является то, что приходится выполнять лишнюю работу по переносу неизменившихся значений, что присуще сплошному методу в отличие от событийного. Однако невзирая на это, основной событийный метод моделирования в ИКСМ уступает ему при моделировании схем, представленных на вентиляльном уровне, в 2-6 раз по скорости моделирования при случайных воздействиях. Изложенные алгоритмы синхронного моделирования легко распространяются на случай моделирования с единичными задержками [9].

Рассмотрим параллельное моделирование неисправностей, основой которого является описанный выше метод моделирования исправных схем. Определим классы неисправностей, которые будем моделировать. Под неисправностью на линии J понимаем изменение ее значения V_j под воздействием факторов $X = (x_1, \dots, x_m)$, т.е. $V_j = V_j(X)$. В частности, если $V_j = 0(1)$, мы имеем известный класс одиночных константных неисправностей.

Группу неисправностей будем описывать массивом данных $(l, N_l, V_l(X))$, $l = (1, M)$, где N_l — номер линии проявления неисправности, V_l — функция неисправности, l — номер неисправности в группе. Внесение неисправности означает замену для линии N_l в l -м разряде поля из M битов значения линии N_l значением $V_l(X)$. При параллельном моделировании легко реализуется моделирование одиночных константных неисправностей (замена значения на линии константным значением 1 или 0) и моделирование неисправностей типа «короткое замыкание» (в зависимости от технологии, применяемой при изготовлении БИС, значение неисправных линий может быть вычислено с помощью логической суммы/произведения их значений в исправном состоянии [11]).

Основное отличие моделирования неисправностей от параллельного моделирования исправного устройства на входных наборах заключается в том, что значения входных полюсов всегда будут одинаковы для всей группы из M неисправностей (исключение составляют неисправности на входах), т.е. все M битов поля для каждого входа равны 1 или 0. Необходимо также осуществлять замену значений на линиях проявления неисправностей согласно данным $(l, N_l, V_l(X))$.

Алгоритм моделирования группы неисправностей для комбинационных устройств может быть описан следующим образом.

1. Если конец входных воздействий, то выход из алгоритма, иначе установить новые значения внешних входов и установить $F = 0$.
2. Установить тип базового элемента $i = 0$.
3. Вычислить $i = i + 1$.
4. Если i больше K , перейти к п.10.
5. Сформировать поля входов W_1, W_2, \dots, W_k элементов типа i .
6. Вычислить значения выходов элементов типа i в поле R_i .
7. Установить значения на линиях проявления неисправностей N_1, N_2, \dots, N_m в поле R_i согласно функциям неисправностей.

8. Если R_i не равно W_i , установить $F = 1$.
9. Перейти к п.3.
10. Если $F = 1$, вычислить $W' = R$ и перейти к п.2.
11. Перейти к п.1.

Опишем алгоритм моделирования для последовательных устройств.

1. Если конец входных воздействий, то выход алгоритма, иначе установить новые значения внешних входов и установить $F = 0$.
2. Установить номер итерации $T = 0$.
3. Установить тип базового элемента $i = 0$.
4. Вычислить $i = i + 1$.
5. Если i больше K , то вычислить $T = T + 1$ и перейти к п.11.
6. Сформировать поля входов W_1, W_2, \dots, W_k из общего поля выходов $W = (W^1, \dots, W^n)$.
7. Вычислить значения выходов элементов типа i в поле R_i .
8. Установить значения на линиях проявления неисправностей N_1, N_2, \dots, N_m в поле R согласно функциям неисправностей.
9. Если R_i не равно W_i , установить $F = 1$.
10. Перейти к п.4.
11. Если $F = 0$, перейти к п.1.
12. Если номер итерации больше заданного H_{\max} , то установить несовпадающие линии полей W_i в неопределенное состояние и перейти к п.3.
13. Вычислить $W = R$ и перейти к п.3.

Необходимо сказать, что в описанных алгоритмах моделируются неисправности на выходах элементов. Если же рассматривать неисправности на входах элементов, то внесение неисправностей (пп. 7, 8 алгоритмов для комбинационных и последовательных устройств соответственно) необходимо производить после подготовки полей входов элементов (пп. 5, 6 соответствующих алгоритмов).

Т а б л и ц а 2.

Номер схемы	Число вентилей	Число входов	Константные неисправности		Неисправности короткого замыкания		M
			число неисправнос- тей	время мо- делирования (мин)/итера- ции	число неисправнос- тей	время мо- делирования (мин)/итера- ции	
1	544	24	1088	13/12	544	14/12	480
2	1024	24	2048	48/13	1024	31/13	480
3	4096	48	2048	161/14	1080	80/14	120
4	10096	64	1008	220/14	1008	231/14	48
5	22736	64	1008	680/14	1008	450/14	16
6	576	24	960	21/22	480	12/23	480
7	1072	32	736	38/30	368	21/30	368
8	4224	48	968	320/41	440	150/41	88
9	8432	96	480	620/61	480	640/61	40

В табл.2 представлены результаты моделирования неисправных схем. Комбинационные схемы 1-5 моделировались в двоичном алфавите, последовательностные схемы — в троичном алфавите. Количество входных наборов равно 50. Другие параметры аналогичны данным при исправном моделировании.

СПИСОК ЛИТЕРАТУРЫ

1. Шмид А.В., Святский А.В. Автоматизация логического проектирования ЭВМ// Итоги науки и техники. Сер. Техническая кибернетика. — М.: ВИНТИ, 1986. — Т.20. — С. 136-204.
2. Армстронг Дж.Р. Моделирование цифровых систем на языке VHDL. — М.: Мир, 1992. — 175 с.
3. Андрухин А.И. Об одном способе параллельного моделирования логических схем// АВТ. — 1990. — №6. — С. 87-88.
4. Андрухин А.И. Возможности параллельного логического моделирования на ЕС ЭВМ// Моделирование и диагностика управляющих систем.— Киев: Наукова думка. 1991. — С. 5 - 8.
5. Джордс Р. Справочник программиста персональных компьютеров типа IBM PC, XT и AT. — М.: Финансы и статистика. 1992. — 544 с.
6. Прокофьев Б.П., Сухарев Н.Н., Храмов Ю.Е. Графические средства Turbo C и Turbo C++. — М.: Финансы и статистика, 1992. — 150 с.
7. Чертова В.Г. Как поставить точку// В мире ПК. — 1993. — №1. — С. 115-125.
8. Киносита К., Асада К., Карацу О. Логическое проектирование СБИС. — М.: Мир, 1988. — 309 с.
9. Бадулин С.С., Барнаулов Ю.М., Бердышев В.А. и др. Автоматизированное проектирование цифровых устройств. — М.: Радио и связь, 1981. — 240 с.
10. Андрухин А.И., Сперанский Д.В. ИКСМ — иерархическая компилятивная система моделирования и построения тестов цифровых устройств// Методы и средства технической диагностики: Сб. материалов XI Межвуз. школы-семинара «Методы и средства технической диагностики». — Ивано-Франковск, 1992. — С. 99-102.
11. Fault-tolerant computing. — Prentice-Hall, 1986. — Vol.1. — 415 p.

Поступила в редакцию 15.11.93