

ПРИМЕНЕНИЕ СТРАТЕГИИ СИМУЛЯЦИИ ОТЖИГА ДЛЯ ПОСТРОЕНИЯ ИНИЦИАЛИЗИРУЮЩИХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ЦИФРОВЫХ СХЕМ

В статье предложен новый алгоритм построения инициализирующих последовательностей для синхронных последовательностных схем. Для решения данной задачи предлагается использовать новую оптимизирующую стратегию «симуляции отжига». Преимуществом данной стратегии в отличие от детерминированных методов является возможность работы со схемами большой размерности.

1. Введение.

Разработка современных цифровых схем является сложной задачей, которая состоит из множества этапов: от проектирования дизайна до тестирования готовых образцов. При этом фаза тестирования является наиболее дорогостоящей: её стоимость может достигать до 80% стоимости всего дизайна. Тестирование заключается в подаче на внешние входы схемы последовательностей входных сигналов и наблюдении на внешних выходах реакции схемы. Часто непосредственно перед тестированием необходимо выполнить инициализацию схемы, т.е. перевести её в некоторое известное состояние. Одним из способов решения задачи является аппаратный, при котором в схему добавляются цепи, подводящие сигнал «сброс» ко всем элементам состояний. Считается, что после подачи сигнала «сброс», схема начинает работу из состояния, определённого при разработке.

Другим подходом является построение инициализирующих последовательностей. Таким образом, перед разработчиком возникает задача построения автоматизированного инструментария, который позволяет определить: существует ли для данной схемы инициализирующая последовательность и, если существует, построить такую последовательность.

Данная задача является одной из «классических» в технической диагностике. В настоящее время предложено несколько подходов к её решению. К точным методам относится, например, алгоритм, предложенный в [1]. Описанный в работе метод основан на построении дерева решений. Подобные методы хорошо зарекомендовали себя и позволяют искать решение для схем малой и средней размерности. Однако для больших схем они оказываются неприменимы из-за чрезмерных требований к ресурсам памяти. Алгоритм данного класса также описан в [2] и ему присущи те же ограничения. Задача построения инициализирующих последовательностей применительно к проблеме построения тестов исследуется в работе [3]. Развитием данных методов является работа [4]. Подход, объединяющий «жадные» алгоритмы с методом ветвей и границ предложен в [5]. Он показывает приемлемые в практическом смысле результаты для широкого класса схем, однако в некоторых случаях требует завышенных ресурсов памяти.

В последнее время распространение получили методы, основанные на моделировании [6-7]. Это связано с тем, что задача моделирования является более простой в сравнении с задачей построения входных последовательностей и, следовательно, может быть решена для больших схем. Использование данных, полученных в результате моделирования, позволяет строить входные последовательности с заданными свойствами для схем большой размерности. К данной группе методов также относятся генетические алгоритмы [8]. Авторы также успешно применяли генетический алгоритм для решения задачи построения инициализирующих последовательностей [9].

Данные методы являются приближёнными в том смысле, что они не всегда дают точный ответ о существовании инициализирующей последовательности. Однако они позволяют находить компромисс между приемлемыми в практическом смысле и ре-

альными затратами при решении той или иной задачи, например ресурсов времени или требуемой памяти.

Целью данной работы является изучение возможности применения новой стратегии «симуляции отжига» к задаче построения инициализирующих последовательностей цифровых схем.

Данная статья имеет следующую структуру. Во втором разделе описывается используемая модель последовательностных схем и производится формальная постановка задачи. В третьем разделе описывается стратегия симуляции отжига и проводится её краткое сравнение с генетическими алгоритмами. В следующем разделе описан предлагаемый алгоритм построения инициализирующих последовательностей, который использует стратегию симуляции отжига. В заключении делаются выводы и отмечаются возможные дальнейшие исследования.

2. Задача построения инициализирующих последовательностей.

При решении поставленной задачи используется модель синхронной последовательностной схемы (рис.1). В ней выделяют комбинационные блоки и блоки элементов состояний, которые представлены D-триггерами. Комбинационные блоки представляют собой в общем случае сеть, вершинами которой являются логические элементы, входы и выходы. Схема характеризуется следующими числовыми параметрами: $\#V_x$ – число внешних входов схемы, определяет размерность вектора входных наборов v ; $\#V_{yx}$ – число внешних выходов схемы, определяет размерность вектора выходов Y ; $\#Tr$ – число элементов состояний, размерность вектора T ; $\#Эл$ – число элементов схемы.



Рис.1. Модель синхронной последовательностной схемы.

Входной вектор v – упорядоченное множество двоичных значений, которое подаётся на вход цифровой схемы в определённый такт времени. Последовательность s_i заданной длины l_i – упорядоченное множество из l_i векторов, которые подаются на вход схемы в последовательные такты времени. Обозначение v_{ij} говорит, что мы рассматриваем в последовательности s_i вектор с номером j ($j = 0, \dots, l_i - 1$).

Моделирование работы синхронной схемы происходит путём преобразования её в псевдокомбинационный эквивалент путём удаления элементов состояний [Барашко], при этом входы элементов состояний (вектор T_i на рис.1) называются псевдовыходами, а их выходы (вектор T_{i-1} на рис.1) – псевдовходами.

Далее выполняется итеративное моделирование полученного эквивалента, которое заключается в следующем. В момент подачи очередного импульса синхронизации на входы схемы подаются новые входные значения v_i для момента времени i , на псевдовходы подаются значения псевдовыходов схемы в предыдущий такт времени T_{i-1} , после чего путём моделирования комбинационной части схемы для такта времени i формируются выходные сигналы схемы Y_i и сигналы псевдовыходов T_i .

Возможно несколько различных постановок задачи инициализируемости схемы. Мы в данной работе будем рассматривать задачу логической инициализируемости схемы. Задача состоит в нахождении одиночной последовательности s , которая переводит схему из неопределённого состояния в некоторое известное состояние.

Дадим более формальное определение. Пусть Q - множество всех состояний последовательностной схемы, тогда $Q = \{0,1,X\}^{\#Tp}$, где $0,1,X$ – элементы трёхзначного алфавита моделирования схемы. Пусть S - множество всех возможных определённых состояний схемы, состоящей из $\#Tp$ триггеров: $S = \{0,1\}^{\#Tp}$. Пусть также Σ - множество всех возможных входных последовательностей s_i . Очевидно также, что для выбранного трёхзначного моделирования $S \subset \Sigma$. Пусть $x \in Q$ начальное неопределённое состояние. Функция $F: Q \times \Sigma \rightarrow Q$ обозначает все состояния, достижимые схемой при поступлении на её вход последовательностей из Σ при использовании трёхзначного алфавита моделирования.

Определение. Последовательность $s_i \in \Sigma$ называется инициализирующей для заданной схемы, если в финальном состоянии $q = F(x, s_i)$ все состояния определены, т.е. $q \in S$.

Известно также, что не все функционально инициализируемые схемы являются логически инициализируемыми.

3. Стратегия симуляции отжига.

3.1. Алгоритм симуляции отжига.

Стратегия симуляции отжига была впервые описана в [11], а её применение к решению больших комбинаторных задач в [12]. Данный метод моделирует физический процесс восстановления, который заключается в нагреве и последующем контролируемом охлаждении некоторой субстанции. В результате следует ожидать появления упорядоченной кристаллической структуры, которая по свойствам отличается от структуры с дефектами, которая образуется в результате быстрого охлаждения.

Кратко опишем суть стратегии симуляции отжига. Для поиска решения вводится понятие конфигурации K_i , которая является потенциальным решением поставленной комбинаторной задачи. Также выбирается кодирование для представления точек в пространстве поиска в виде конфигураций. С каждой конфигурацией K_i ассоциируется функция стоимости $C_i = C(K_i)$. Данная функция показывает, как хорошо выбранная конфигурация может решить поставленную задачу. Оценивание решения (конфигурации) состоит из его декодирования и собственно вычисления оценочной функции, показывающей целесообразность конфигурации для решения данной проблемы. Цель оптимизации – найти конфигурацию, которая минимизирует функцию стоимости.

Дополнительно вводится понятие окружения конфигурации K_i : множество потенциальных конфигураций, которые могут быть получены из K_i путём применения некоторых возмущающих операций.

Тогда общая схема стратегии симуляции отжига может быть описана следующим образом (рис.2).



Рис.2. Общая схема алгоритма симуляции отжига.

- 1) В начале работы формируется начальная конфигурация K_0 и происходит вычисление её оценки $C_0 = C(K_0)$. Начальная конфигурация принимается в качестве текущей конфигурации $K_i = K_0$. Определяется текущая начальная температура $T_i = T_0$. Далее итеративно вплоть до выполнения условия остановки выполняются следующие шаги.
- 2) Формируется некоторое окружение текущей конфигурации K_i с помощью операции возмущения.
- 3) Для всех конфигураций, входящих в окружение, вычисляется оценка и выбирается наилучшая конфигурация $K_{пром}$.
- 4) Вычисляется изменение функции стоимости:

$$\Delta C_i = C(K_{пром}) - C(K_i).$$

- 5) Произведённые возмущением изменения либо принимаются, либо отвергаются: если изменение функции стоимости отрицательное, то промежуточная конфигурация заменяет текущую. В противном случае такая замена происходит на основании распределения Больцмана:

$$K_{i+1} = \begin{cases} K_{пром}, & \text{если } \Delta C_i < 0; \\ K_{пром} & \text{с вероятностью } P = \exp(-\Delta C_i / kT_i), & \text{если } \Delta C_i > 0; \end{cases}$$

где k - эвристическая константа.

Такой способ приёма изменений конфигурации позволяет при большей температуре чаще принимать плохие решения, чем отбрасывать их. При снижении температуры также снижается вероятность принятия худших решений.

- 6) Изменяется текущая температура:

$$T_{i+1} = \text{обновить}(T_i).$$

- 7) Переход к шагу 2.

Завершение работы алгоритма происходит либо на шаге 3, когда в окружении найдена конфигурация, удовлетворяющая решению, либо на шаге 6 по достижении конечной температуры (некоторого максимального числа итераций). В этом случае в качестве приближённого решения принимается последняя исследуемая конфигурация.

Из описания алгоритма видно, что для его применения к решению некоторой конкретной задачи необходимо задать компоненты алгоритма симуляции отжига:

- кодирование конфигураций;
- операцию построения окружения конфигурации (операцию возмущения);
- вид оценочной функции; сложность её вычисления определит общую сложность алгоритма;
- параметры, связанные с температурой: начальная и конечная температура (последняя называется температурой заморозки), изменение температуры, число итераций для каждого значения температуры;
- распределение вероятности для принятия худших решений. Наиболее часто используют распределение Больцмана, хотя в общем случае оно может быть иным.
- критерии остановки алгоритма.

3.2. Сравнение симуляции отжига и генетических алгоритмов.

Проведём краткое сравнение двух стратегий поиска решений: симуляции отжига и генетических алгоритмов [13]. Данные подходы объединяют сразу несколько моментов.

- 1) Обе стратегии разработаны для решения NP-полных задач.
- 2) Оба подхода используют эволюцию свойств некоторых предварительных решений. От итерации к итерации переходят, в основном, особи с лучшей оценкой. При этом в стратегии симуляции отжига происходит улучшение свойств одного реше-

ния, которое называется конфигурацией. Генетические алгоритмы работают с группой решений.

- 3) Новые конфигурации решений получаются в результате модифицирующих операций. В стратегии симуляции отжига это единственная операция, называемая возмущением. В генетических алгоритмах ей соответствует оператор мутации. Дополнительно в генетических алгоритмах используется оператор скрещивания, что даёт возможность получать улучшающие свойства от двух особей.
- 4) В обеих стратегиях начальные конфигурации строятся случайно.
- 5) Обе стратегии для эффективной работы требуют хорошо построенной функции оценки особей. Качество построения данной функции во много определяет эффективность алгоритмов. С другой стороны сложность вычисления оценочной функции определяет скорость работы всего алгоритма. В практических задачах часто необходимо определить баланс между видом оценочной функции и сложностью её вычислений.
- 6) Проблема кодирования конфигураций решений и восстановления свойств решения из кодирования.
- 7) Оба этих основанных на самоорганизации процесса имеют тот недостаток, что они вырождаются в неполиномиальные.
- 8) Обе стратегии используют большое количество эвристик: предельное число итераций, критерий остановки эволюции, вероятности применения модифицирующих операций и т.д.

Такое краткое сравнение двух парадигм, заимствованных из разных областей человеческих знаний, позволяет надеяться на их примерно равные свойства применительно к поиску оптимальных решений. Однако более простая парадигма в симуляции отжига должна в этом случае давать преимущество.

4. Предлагаемый алгоритм.

Как было указано в предыдущем разделе, для построения алгоритма симуляции отжига для решения конкретной задачи необходимо задать его следующие компоненты: кодирование конфигурации, вид функции стоимости, отображение возмущения и определить численные значения эвристических констант.

Пространство поиска Σ состоит из всех возможных входных последовательностей произвольной длины.

Конфигурацией K_i (точкой в пространстве поиска Σ) является входная последовательность s_i (рис.3). Отметим, что вследствие близости подходов, такое представление совпадает с представлением особи в генетических алгоритмах построения тестовых входных последовательностей.

Окружение конфигурации строится на основе следующих операций:

- вертикальное (горизонтальное) возмущение: в конфигурации, представленной в виде на рис.3, случайно выбирается и изменяется столбец (строка);
- добавление (удаление) строки: в случайном месте в конфигурации добавляется (удаляется) строка.

Видно, что эти операции являются подобными операциям мутации, которые авторы использовали в генетических алгоритмах [9].

Для каждой конфигурации K_i необходимо поставить в соответствие функцию стоимости $C_i = C(K_i)$. В нашем алгоритме оценочная функция имеет следующий вид:



Рис.3. Кодирование конфигурации.

$$C(K_i) = f(s_i) = f(n_1, n_2, n_3) = (c_1 * n_1 + c_2 * n_2) * c_3^{n_3},$$

где $n_1, n_2, n_3, c_1, c_2, c_3$ – следующие параметры:

- n_1 - отношение числа инициированных триггеров к их общему числу; чем больше триггеров схемы перешло из неопределённого состояния в определённое, тем выше качество заданной последовательности;
- n_2 - активность схемы или число событий моделирования; чем выше число событий при моделировании схемы на заданной последовательности, тем выше вероятность, что ещё неустановившиеся триггеры перейдут в определённое состояние.
- n_3 - длина последовательности; из двух последовательностей при прочих равных условиях необходимо предпочесть более короткую;
- c_1 и c_2 константы, которые уравнивают влияние на функцию стоимости числа триггеров, перешедших в известное состояние, и активность вентилей в схеме. Очевидно, что для построения эффективной функции оценки недостаточно одного из параметров n_1 или n_2 , поскольку это может завести алгоритм в тупик, например в случае, когда число установленных триггеров не изменяется, а активность схемы близка к нулю;
- c_3 позволяет искать более короткие входные последовательности; если данную константу выбрать близкой к единице $c_3 < 1$, то из двух входных последовательностей при прочих равных условиях (число активизированных триггеров и активность схемы) большую оценочную функцию будет иметь та последовательность, длина которой меньше.

Параметры n_1, n_2 вычисляются на основании результата моделирования работы исправной схемы. Для этого использовался алгоритм моделирования работы синхронных последовательностных схем в трёхзначном алфавите [14].

Описанный выше алгоритм в виде псевдокода представлен на рис.4.

5. Выводы.

В статье предложен новый подход к решению задачи построения инициализирующих последовательностей цифровых схем. Для этого предложено использовать стратегию симуляции отжига. Простота основных принципов построения стратегии и основанный на моделировании способ вычисления функции стоимости позволяют данному алгоритму работать с цифровыми схемами большой размерности. Предложены основные компоненты алгоритма симуляции отжига: кодирование конфигураций, способы построения окружения конфигураций, вид функции стоимости; а также приведён псевдокод алгоритма

В качестве дальнейших исследований можно отметить следующие направления:

- исследование зависимости эффективности алгоритма от значений эвристических констант;
- исследование возможности применения алгоритма симуляции отжига к другим задачам построения входных последовательностей.

Список литературы

1. C. Pixley, S. Jeong, G. Hatchel, "Exact Calculation of Synchronization Sequences based on Binary Decision Diagrams", IEEE Trans. on CAD, Vol.13, pp.1024-1034, 1994.
2. J.-K. Rho, F. Somenzi, C. Pixley, "Minimum Length Synchronizing Sequences of Finite State Machine", Proc. ACM Design Automation Conf., 1993, pp.463-468.
3. J.A. Wehbeh, D.G. Saab, "Initialization of Sequential Circuits and its Application to ATPG," Proc. IEEE VLSI Test Symp., 1996, pp. 246-251.
4. R.E. Bryant "Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams", ACM Computing Surveys, Vol.24, Nr 3, 1992, pp. 293-318.

```

симуляция_отжига (описание_схемы, параметры)
{
  K0=построение_начальной_конфигурации ();
  Ti=Tнач;
  пока ( Ti>Tкон )
  {
    Kпром=выбрать_конф._из_окружения (Ki);
    Cпром=C (Kпром);
    если ( цель_достигнута (Cпром) )
    {
      решение=Kпром;
      возврат;
    }
    ΔCi=Cпром-Ci;
    если ( ΔCi>0 )
      Ki+1=Kпром;
    иначе
      Ki+1=Kпром с вер. P (ΔCi)=exp (-ΔCi/kTi);
    Ti+1=обновить (Ti);
  }
  решение=Ki+1;
}

```

Рис.4. Алгоритм симуляции отжига построения инициализирующих последовательно-

5. M. Keim, B. Becker, B. Stenner, "On the (Non-)Resettability of Synchronous Sequential Circuits," Proc. IEEE VLSI Test Symp., 1996, pp. 240-245.
6. P. Camurati, F. Corno, P. Prinetto, M. Sonza Reorda, "A simulation-based approach to test pattern generation for synchronous circuits", Proc. IEEE VLSI Test Symposium, 1992, pp.263-267.
7. V.D. Agrawal, K.-T. Cheng, P. Agrawal, "CONTEST: A concurrent Test Generator for Sequential Circuits", Proc. 25-th Design Automation Conference, 1988, pp.84-89.
8. P. Prinetto, M. Rebaudengo, M. Sonza Reorda, "An Automatic Test Pattern Generator for Large Sequential Circuits based on Genetic Algorithms" In Proc. Int. Test Conf., 1994, pp. 240-249.
9. Д.Е. Иванов, Ю.А. Скобцов, А.И. Эль-Хатиб Построение инициализирующих последовательностей синхронных цифровых схем с помощью генетических алгоритмов Проблемы інформаційних технологій.-2007.-№1.-с.158-164.
10. Барашко А.С., Скобцов Ю.А., Сперанский Д.В. Моделирование и тестирование дискретных устройств. – Киев:Наукова думка, 1992. – 288 с.
11. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, "Equation of State Calculation by Fast Computing Mashines", J. of Chem.Phys., Vol.21, No.6, pp.1087-1092, 1953.
12. S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, "Optimization by simulating annealing", Science, 220, pp.671-680, 1983.
13. Ю.А. Скобцов. Основы эволюционных вычислений.– Донецк: ДонНТУ, 2008.- 326с.
14. Иванов Д.Е., Скобцов Ю.А. Система моделирования и генерации тестов цифровых схем // Наукові праці Донецького державного технічного університету, серія "Обчислювальна техніка та автоматизація», випуск 12. – Донецьк. – 1999. – С.143-150.