

Рензинг. Высокопроизводительные сети /DiaSoft, К. 1998. 421 с. 6. Новиков Ю.В., Кондратенко С.В. Аппаратура локальных сетей: функции, выбор, разработка. М.: ЭКОМ, 1998. 288 с. 7. Новиков Ю.В., Кондратенко С.В. Локальные сети: архитектура, алгоритмы, проектирование. М.: Издательство ЭКОМ, 2000. 312 с. 8. TechFest Ethernet Technical Summary. <http://www.techfest.com/networking/lan/ethernet3.htm>. 9. http://www.synapse.de/ban/HTML/P_LAYER2/Eng/P_lay215.html. 10. Питерсон Д. Теория сетей Петри и моделирование систем. М.: Мир, 1984. 264с. 11. 66. Автоматизация диагностирования электронных устройств/ Ю.В.Малышенко и др./ Под ред.В.П.Чипулиса. М.: Энергоатомиздат, 1986. 216с.

Поступила в редколлегию 12.07.2000

Ханько Вадим Викторович, аспирант кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: техническая диагностика компьютерных систем и сетей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26.

УДК 681.326: 519.713

В.И. ХАХАНОВ, А.С. ШКИЛЬ, Е.В. КОВАЛЕВ

МОДЕЛЬ ПРОЦЕССА ПЕРЕХОДА ОТ СОДЕРЖАТЕЛЬНОГО ГРАФА МИКРОПРОГРАММЫ К ГРАФУ АВТОМАТА

Рассматривается задача перехода от традиционной формы представления закона функционирования микропрограммного автомата в виде граф-схемы алгоритма (ГСА) к содержательному графу автомата. Полученное описание конечного автомата в виде содержательного графа используется в системе визуального ввода САПР Active-HDL. Рассматриваются особенности использования языковых конструкций VHDL при описании функций переходов и выходов конечных автоматов.

При описании проектов микропрограммно-управляемых цифровых операционных устройств существует два способа описания алгоритма их функционирования: в виде содержательной ГСА микропрограммы и в виде содержательного графа автомата [1]. В англоязычной технической литературе способ описания алгоритма функционирования автомата в виде содержательной ГСА (flow chart) часто называют ASM (Algorithmic State Mashine), а в виде содержательного графа автомата (Moore/Mealy state) – FSM (Finite State Mashine).

Указанные формы описания алгоритмов функционирования конечных автоматов (КА) используются современными САПР программируемой логики в системах визуального ввода проектов, носят конкурирующий характер и применение одного из них зависит от опыта и привычек разработчика. Большинство современных промышленных

САПР устройств программируемой логики в качестве основной системы визуального ввода закона функционирования КА используют содержательный граф автомата (FSM). В свою очередь большинство отечественных разработчиков цифровой аппаратуры привыкли оперировать описанием микропрограммных автоматов (МПА) в виде ГСА. Исходя из этого, актуальной является разработка формальной процедуры перехода от описания МПА в виде ГСА к описанию в виде содержательного графа автомата, с учетом особенностей конкретной системы визуального ввода САПР программируемых логических интегральных схем (ЛИИС), в частности Active-HDL фирмы Aldec [2].

Содержательная ГСА микропрограммы часто используется при проектировании МПА, так как алгоритмы выполнения операций в устройстве удобно описывать в графической форме. ГСА строится с использованием вершин четырех типов и дуг, связывающих вершины. Начальная вершина отмечает начало алгоритма и имеет единственный выход, из которого исходит дуга к первой выполняемой вершине графа. Для обозначения операторов и переходов используются вершины двух типов: функциональные и условные. Функциональная (основная) вершина определяет действие – совокупность функционально совместимых микроопераций, выполняемых параллельно. Микрооперации в вершине представляются в виде операторов присваивания. В функциональную вершину может входить любое, но не меньше 1, число дуг, а выходит только одна дуга. Условная вершина используется для разветвления вычислительного процесса в одном из двух возможных направлений, выбор которого осуществляется текущим значением логического условия, указанного в вершине. Если условие имеет значение 0, вычислительный процесс развивается по дуге, отмеченной символом 0, в противном случае – по дуге, отмеченной символом 1. В условную вершину может входить любое число дуг, но выходят всегда две дуги. Конечная вершина отмечает конец микропрограммы. В конечную вершину может входить любое число дуг.

Кроме содержательной ГСА описание микропрограммы включает описание слов и массивов. Основным элементом информации, с которым оперирует функциональная микропрограмма, является слово. Наименование и формат слова задаются в следующем виде: $C(n_1:n_2)$, где C – идентификатор слова; n_1 и n_2 – номера старшего и младшего двоичных разрядов слова. Примем, что разряды слова нумеруются слева направо неотрицательными целыми числами, т.е. $n_1 \geq 0$, $n_2 \geq 0$, $n_1 < n_2$. В соответствии с описанием разряды слова получают номера n_1, n_1+1, \dots, n_2 . Так, описание $A(0:31)$ определяет 32-разрядное слово A , а описание $B(1:8)$ – 8-разрядное слово B . Описание одноразрядного слова состоит только из идентификатора. Так, описания $Z, ПП, ЗНАК$ определяют три одноразрядных слова.

Совокупность слов, имеющих одинаковую длину, может объединяться в массив. Массив описывается в следующем виде:

$M[m_1:m_2] (n_1:n_2)$,

где M – идентификатор массива; m_1, m_2 – границы номеров слов, составляющих массив, причем $m_1 < m_2$; n_1, n_2 – номера старшего и младшего разрядов слова. Например, описание РП[0:15] (0:31) представляет массив из 16 32-разрядных слов, а описание ОП[0:4095] (0:63) – массив из 4096 64-разрядных слов.

Чтобы спроектировать структуру операционного устройства, необходимо в функциональной микропрограмме определить способ использования значений слов и массивов, обуславливаемый типом элементов хранимой информации. В зависимости от способа использования выделяются следующие основные типы слов: 1) входные – значения присваиваются вне микропрограмм; 2) внутренние (локальные) – значения присваиваются и используются только внутри микропрограммы; 3) вспомогательные – значения присваиваются и используются только внутри микропрограммы, но существуют не постоянно, а только в течение ограниченных интервалов времени; 4) выходные – значения присваиваются внутри микропрограммы и используются вне ее. Типы слов обозначаются буквами: I – входные, L – внутренние, A – вспомогательные, O – выходные. Некоторые слова могут использоваться как элементы информации нескольких типов. Например, слово может употребляться как входное, внутреннее и выходное. Поэтому некоторые слова могут иметь следующие типы: IL, LO, IO. Последний тип соответствует слову, значение которому может присваиваться и использоваться до входа в микропрограмму и после выхода из нее.

Для синтеза МПА по содержательной ГСА необходимо выполнить отметку его состояний [1]. Заметим, что начальная и конечная вершины могут отмечаться одинаковыми метками, для автомата Мура отмечаются операторные вершины, а для автомата Мили – входы вершин, следующих за операторными. Следует также учитывать, что в автомате Мили между двумя отметками должна находиться операторная вершина. Если между отметками она отсутствует, то при переходе к графу автомата указанный (пустой) путь между отметками не учитывается.

На рис. 1 представлена ГСА микропрограммы умножения и выполнена разметка состояний для автомата Мура с различными метками для начальной и конечной вершин, а в табл. 1 приведено описание слов в указанной микропрограмме [3].

Визуальные средства ввода проектов МПА большинства современных САПР, ориентированных на проектирование устройств на базе программируемой логики, реализуют надстройки над базовым языком описания аппаратуры. В данном случае под надстройками подразумеваются программы, реализующие как правило графический ввод и затем преобразующие этот формат в грамматику выбранного языка описания аппаратуры. Одной из таких надстроек является программа ввода, редактирования и трансляции автоматов Мура и Мили (State diagram editor), используемая в пакете Active-HDL. Необходимо заметить, что эта программа реализует только трансляцию блоков данных на язык описания аппаратуры (Hardware Description Languages). Ее разработчики

не ставили перед собой задачу получить таблицу переходов и выходов автомата. Их целью было построение корректного HDL-кода, поэтому задача трансляции решалась следующим образом. Есть шаблон автомата на языке HDL. В соответствующие строки кода необходимо подставить строки из файла формата графического описания. При этом, естественно, не учитывалось, что же представляет собой полученный код. Соответствует ли он модели автомата Мура, Мили или С-автомата.

В подсистеме визуального ввода (State Diagram Editor) САПР Active-HDL используется описание содержательного графа С-автомата с дополнительными функциями состояний, перечисленными ниже. В данном содержательном графе функциональные вершины ГСА (список микроопераций) могут быть заданы 6 различными способами, первые 4 из которых связаны со способом задания содержательного графа: 1) функциональная вершина для перехода; 2) функциональная вершина входа в состояние; 3) функциональная вершина петли и выхода из состояния; 4) функциональная вершина выхода из состояния; 5) функции автомата; 6) функции диаграммы.

Функциональная вершина перехода 1 по функции выхода полностью совпадает со способом задания функции выхода автомата Мили. Функциональная вершина петли и выхода из состояния 3 по своему назначению полностью совпадает с функцией выхода автомата Мура. Функциональная вершина входа в состояние 2 может использоваться для минимизации, так как ее можно представить без потери информации как вершину 1, связанную со всеми дугами, входящими в соответствующее состояние. Функциональная вершина выхода в состояние 4 может использоваться для минимизации, поскольку ее можно представить без потери информации как вершину 1, связанную со всеми дугами, кроме дуги сброса, выходящими из соответствующего состоя-

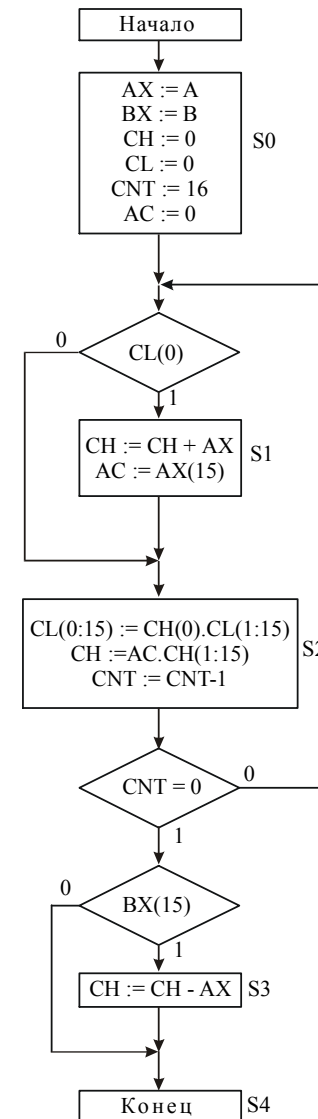


Рис. 1. Содержательная ГСА микропрограммы умножения с разметкой состояний автомата Мура

Таблица 1
Описание слов в микропрограмме умножения

Тип	Слово	Пояснения
I	A(0:15)	Множимое
I	B(0:15)	Множитель
L	AX(0:15)	Промежуточные данные для множимого
L	BX(0:15)	Промежуточные данные для множителя
L	CNT(0:4)	Счетчик циклов
L	AC	Перенос
LO	CL(0:15)	Младшая часть произведения
LO	CH(0:15)	Старшая часть произведения

Функции 5 и 6 выполняются параллельно друг другу и являются внешними по отношению к функциям автомата 1, 2, 3 и 4.

Возможно решение задачи приведения функциональных вершин типа 1, 2, 3, 4 к типу 1. Для этого необходимо:

а) для петли объединить функции 3 и 1; б) для перехода из состояния объединить функции 3, 4, 1, 2; в) для перехода по сигналу сброса объединить функции 1, 2.

Однако решение этой задачи может существенно изменить объем обрабатываемых данных, так как функция типа 2 распадается на количество переходов в соответствующее состояние, функция типа 3 распадается на количество переходов плюс количество петель из соответствующего состояния, функция 4 распадается на количество переходов из соответствующего состояния. Если принять во внимание то, что для всех состояний исходящие и входящие переходы существуют, то можно сделать вывод о том, что объем данных будет заметно увеличен.

Визуальная система ввода State Diagram Editor формирует файл в специальном формате ASF, который с помощью транслятора переводится в файл на языке VHDL или Verilog. Данный файл содержит один или более содержательных графов автомата, одно общее описание сигналов (входных, выходных, внутренних, двунаправленных) и отдельное для каждого автомата описание его внутренних переменных. Все указанные типы функций записываются с помощью языковых конструкций VHDL или Verilog в форме поведенческого шаблона, который однозначно соответствует прямой структурной таблице автомата Милли.

Кроме перечисленных функций граф имеет два дополнительных типа дуг: дугу сброса (Reset transition) и дугу без начального состояния (Global transition). Последнюю можно представить как множество дуг, ведущих в конечное состояние из каждого состояния. Также в графе может быть выделено состояние по умолчанию (Default transition), в

которое переходит автомат, если все условия переходов из текущего состояния не выполняются.

Каждая дуга, кроме дуг сброса и дуг без начального состояния может иметь приоритет, который необходим в том случае, когда несколько условий перехода принимают значение "истина". В этом случае выполняется переход с наибольшим приоритетом. Приоритет задается числом от 0 до 16. Нулю соответствует максимальный приоритет, 16 – минимальный. Таким образом, при проверке функций активизации перехода МПА проверяется переход с приоритетом 0, далее с приоритетом 1, 2, ..., 16.

Описание данных является частью графического описания автомата. По сравнению с данными микропрограммы (табл. 1) включено 2 дополнительных сигнала CLK и RESET. Сигнал CLK служит при последующих процедурах синтеза для обозначения цепи синхросигнала, а сигнал RESET – для обозначения сигнала сброса.

В качестве примера продемонстрируем выполнение перехода от содержательного графа микропрограммы умножения, представленного на рис. 1, к содержательному графу автомата Мура.

Вначале выполняется разметка состояний в соответствии с алгоритмом разметки для автомата Мура. Полученные отметки состояний являются вершинами содержательного графа автомата.

Далее анализируются переходы из каждой метки состояний. Для минимизации условий переходов возможно применение следующего алгоритма для каждого отмеченного состояния:

- 1) текущий номер приоритета равен 0;
- 2) выбираем переход (здесь под переходом понимается путь из состояния в состояние, возможно, через несколько условных вершин) с наименьшим числом условных вершин, заносим в таблицу;
- 3) увеличиваем номер приоритета на 1;
- 4) из последнего рассмотренного пути находим последнюю условную вершину, для которой переход по инверсному условию еще не занесен в таблицу;
- 5) выбираем не занесенный в таблицу переход, содержащий найденную условную вершину и проходящий до нее через те же вершины, что и предыдущий выбранный, с наименьшим числом условных вершин, заносим в таблицу;
- 6) при формировании условий перехода условные вершины, для которых все переходы, содержащие инверсное условие, уже есть в таблице, не учитываются;
- 7) если не обработанные переходы из рассматриваемой вершины еще есть, то переходим к пункту 3;

На рис. 2 приведен содержательный граф МПА, ГСА которого приведена на рис. 1. Все его состояния отображаются в виде вершин графа (на рис. 2 – большие окружности с точкой), малые окружности на дугах перехода соответствуют приоритетам дуг при выходе из состояния.

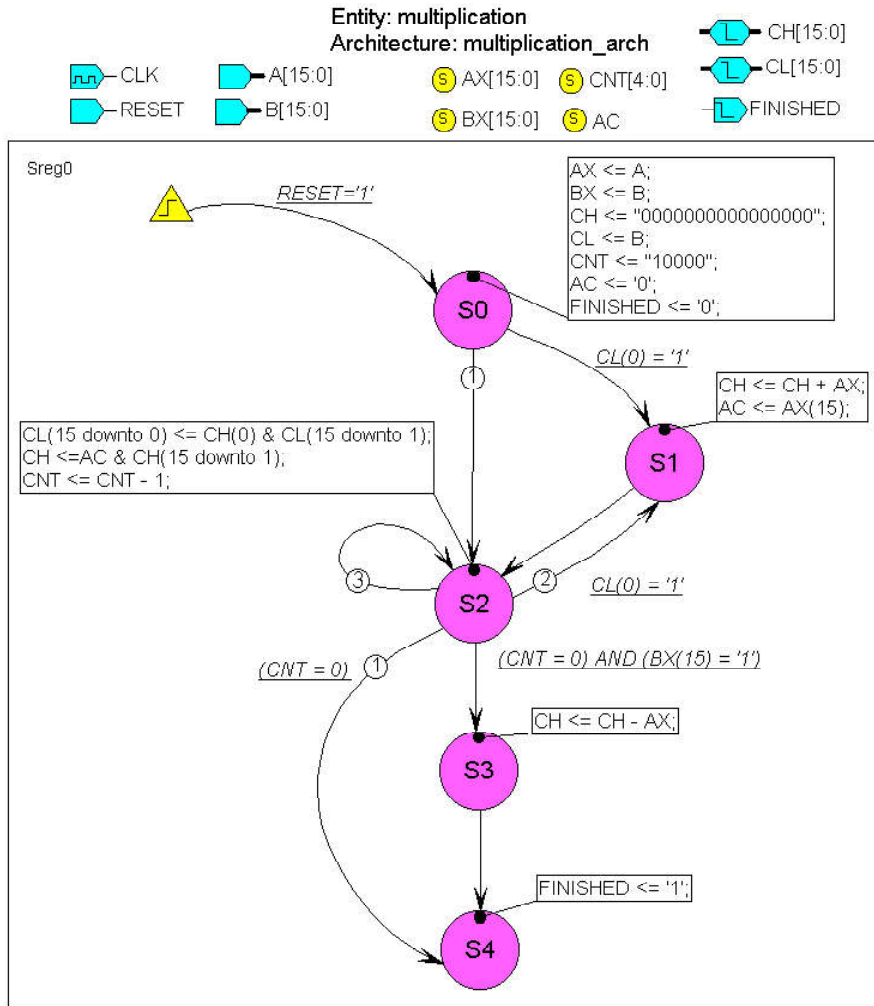


Рис. 2. Содержательный граф автомата, реализующего микропрограмму умножения применительно к САПР

Содержательная ГСА микропрограммы представляет собой описание функции устройства без привязки к состояниям автомата, который реализует данную ГСА. Для перехода от абстрактной модели к структурной удобно пользоваться таблицей переходов. В табл.2 представлена таблица переходов, соответствующая содержательной ГСА микропрограммы умножения, без выполнения кодирования условий перехода булевыми переменными и кодирования состояний автомата.

Таблица переходов для содержательного графа автомата

Начальн. состоян.	Условие перехода	Приоритет	Конечн. состоян.	Функция выхода
-	RESET='1'	-2	S0	AX <= A; BX <= B; CH <= "0000000000000000"; CL <= B; CNT <= "10000"; AC <= '0'; FINISHED <= '0';
S0	CL(0) = '1'	0	S1	CH <= CH + AX; AC <= AX(15);
S2	CL(0) = '1'	2		
S0	-	1	S2	CL(15 downto 0) <= CH(0) & CL(15 downto 1); CH <= AC & CH(15 downto 1); CNT <= CNT - 1;
S1	-	0		
S2	-	3		
S2	(CNT=0) AND (BX(15) = '1')	0	S3	CH <= CH - AX;
S2	CNT=0	1	S4	FINISHED <= '1';
S3	-	0		

Данная таблица является аналогом прямой таблицы переходов-выходов автомата Мили, но вместо входного сигнала в табл.2 находится условие перехода, а вместо значений выходов – функции на языке VHDL (Verilog).

Для дальнейших преобразований, связанных с решением задачи синтеза модели устройства, а также решения задач генерации тестов, необходимо преобразовать строки данной таблицы в конструкции языка VHDL (Verilog). Такое описание является результатом функционального синтеза и может использоваться для функционального моделирования и построения тестов.

Предложенная модель перехода от содержательной ГСА к содержательному графу автомата может оказаться полезной начинающим специалистам-проектировщикам, работающим с пакетом САПР ActiveHDL, а также студентам направления "Компьютерная инженерия", изучающим указанный пакет САПР ПЛИС. Разработанная процедура

построения прямой структурной таблицы с использованием языковых конструкций VHDL может быть использована при проектировании САПР для построения визуальной системы ввода МПА на основе описания в виде GSA (ASM) .

Список литературы: 1. Баранов С.И. Синтез микропрограммных автоматов. Л.: Энергия. 1979. 232 с. 2. Active-VHDL Series. Book #1 - #4. Reference Guide. ALDEC Inc. 1998. 206 p. 3. Майоров С.А., Новиков Г.И. Структура электронных вычислительных машин. Л.: Машиностроение. 1979. 384 с.

Поступила в редколлегию 17.07.2000

Хаханов Владимир Иванович, д-р техн. наук, профессор кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: техническая диагностика вычислительных устройств, систем, сетей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26. E-mail: Hahanov@kture.kharkov.ua

Шкиль Александр Сергеевич, канд. техн. наук, доцент кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: диагностика компьютеров. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26.

Ковалев Евгений Викторович, аспирант кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: техническая диагностика компьютерных систем и сетей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26.

РЕФЕРАТИ

УДК 681.323

Розрахунок кутової похибки спецпроцесора у системі візуалізації / В.М. Гусятин, А.М. Бугрій // АСУ та прилади автоматики. 2000. Вип.112. С. 4-10.

Викладено порядок застосування мажорантної оцінки для розрахунку кутової похибки проектуемого спецпроцесора (СП) у системі візуалізації (СВ) тренажерів транспортних засобів. Для конкретного прикладу показана методика одержання аналітичного співвідношення, що зв'язує кутівий дозвіл СВ із технічними характеристиками СП (розрядність операційних пристроїв, таблиць обчислення функцій), і системи відображення (екрана). Підбираючи параметри цього співвідношення, можна мінімізувати мажоранту і вирішувати задачу оптимізації системи в цілому.

Табл. 1. Бібліогр.: 5 назв.

UDC 681.323

Calculation of angle error criterion of the specially designed processor in a visualization system / V.M.Gusyatin, A.N. Bugriy // Management Information System and Devices. All-Ukr. Sci. Interdep. Mag. 2000. N 112. P. 4-10.

The order of application of majorant evaluation for account of an angle error projected specially designed processor (SDP) in system of visualization (SV) of simulator vehicles, is proposed. For a concrete example is shown the technique of reception of an analytical correlation, which connects the angle sanction SV to characteristics SDP (capacity of operational devices, tables of function evaluation) and system of display (screen). Varying parameters of this correlation it is possible to minimize of majorizing function and to solve a problem of optimization of system as a whole.

Tab. 1. Ref.: 5 items.

УДК 519.7

Про метод та задачі теорії інтелекту. V / З.В. Дударь, О.В. Калініченко, С.Ю. Шабанов-Кушнарєнко // АСУ та прилади автоматики. 2000. Вип. 112. С.11-34.

Запропоновано біонічний підхід до проблеми побудови штучного інтелекту. Описано розвинення спеціалізованого математичного апарату для ефективного моделювання роботи механізмів людського інтелекту.

Табл. 5. Бібліогр.: 5 назв.

UDC 519.7

About a method and problems of the theory of intellect. V / Z.V. Dudar, O.V. Kalinichenko, S.Yu. Shabanov-Kushnarenko // Management Information System and Devices. All-Ukr. Sci. Interdep. Mag. 2000. N 112. P. 11-34.

It is offered bionic approach to a problem of construction of an artificial intelligence. The specialized mathematical instrument for effective simulation of activity of mechanism of human intellect develops.

Tab. 5. Ref.: 5 items.