

УДК 681.3.06

Чебаненко Т.М., к.т.н.

МОДЕЛЬ ПРОГРАММНО-УПРАВЛЯЕМЫХ ОБЪЕКТОВ ДИАГНОСТИРОВАНИЯ

Chebanenko T.M. Model of the program-driven diagnostic objects. The model of specific class of devices offers in the article – the programmatic guided hardware components of the computer systems, that from the point of view of test control and diagnostics processes in a number of properties, substantially different from the objects traditionally examined within the framework of this discipline. Formal presentation of model of the program-driven objects and model of class of disrepairs on the basis of that automation of procedures of construction of effective control and diagnostic tests is possible are examined. Complication of such procedures is estimated and quality of the tests got on the basis of these models is forecast. There is formulated an approach to the construction of models and image transformations, which differs from the known fact that the latter takes into account the semantics. This allows formalized mathematical schemes for image analysis tasks based on their semantics. The formalization of the mathematical schemes for image analysis tasks creates the preconditions for the creation of a unified software framework, the use of which significantly reduces the cost of developing specialized software related to image analysis.

Keywords: diagnostics, fault-class model, computer system, tests model

Чебаненко Т.М. Модель програмно керованих об'єктів діагностування. В статті запропонована модель специфічного класу пристроїв – програмно керованих апаратних компонент комп'ютерних систем, які з точки зору тестового контролю і діагностики мають безліч властивостей, що істотно відрізняються від об'єктів, традиційно розглядаються в рамках цієї дисципліни. Розглядається формальне подання моделі програмно-керованих об'єктів і модель класу несправностей, на основі яких можлива автоматизація процедур побудови ефективних контрольних і діагностичних тестів. Оцінюється складність таких процедур і прогнозується якість одержуваних на основі цих моделей тестів.

Ключові слова: діагностування, модель класу несправностей, комп'ютерна система, модель тестів

Чебаненко Т.М. Модель программно-управляемых объектов диагностирования. В статье предложена модель специфического класса устройств – программно управляемых аппаратных компонент компьютерных систем, которые с точки зрения тестового контроля и диагностики обладают множеством свойств, существенно отличающихся от объектов, традиционно рассматриваемых в рамках этой дисциплины. Рассматривается формальное представление модели программно-управляемых объектов и модель класса неисправностей, на основе которых возможна автоматизация процедур построения эффективных контрольных и диагностических тестов. Оценивается сложность таких процедур и прогнозируется качество получаемых на основе этих моделей тестов.

Ключевые слова: диагностирование, модель класса неисправностей, компьютерная система, модель тестов

Введение

Необходимость применения моделей системного уровня для объектов диагностирования (ОД) при решении задач синтеза тестов объясняется рядом причин.

Первой из них является неприемлемая размерность модели, построенной в соответствии с известными подходами [1]. Причина этого состоит в том, что большинство современных средств вычислительной техники содержат компоненты, реализующие сложные устройства, вплоть до функционально законченных и автономных и изготовленные по технологии БИС и СБИС.

Вторая причина связана с тем, что процесс проектирования таких устройств должен включать разработку тестового обеспечения для них. Учет на ранних стадиях проектирования параметров, характеризующих контролепригодность устройства, всегда оправдывается повышением качества разработки и эффективности самого процесса проектирования.

Третья причина состоит в том, что при использовании сложных компонент в структуре проектируемого устройства разработчик как самого устройства, так и тестового обеспечения имеет их описание на системном уровне представления.

Применение моделей системного уровня для программно-управляемых ОД (ПОД) при решении задачи синтеза тестов является обязательным.

ПОД обладают рядом признаков, позволяющих выделить их в отдельный класс. Во-первых, на системном уровне ПОД не может быть представлен функциональным описанием [2], поскольку реализуемая функция определяется программой, которая априорно неизвестна. Во-вторых, объект можно описать множеством функциональных описаний, каждое из которых соответствует инструкции, принадлежащей системе команд. В-третьих, каждая из инструкций определена на конечном множестве данных.

1. Определение ПОД

В рамках традиционного подхода к представлению модели ПОД на системном уровне производится его декомпозиция, в результате которой выделяют управляющий и операционный автоматы. Это связано, в первую очередь, с методологией синтеза подобных устройств. Однако, с точки зрения оценки контролепригодности и синтеза тестов для ПОД такой подход является малоэффективным. Попытки описать ПОД с помощью автоматной модели, как правило, оканчиваются неудачей. Это связано с тем, что если и существует возможность определения множества состояний, то переходы в таком автомате осуществляются по определенным последовательностям и идентификация отдельных состояний невозможна, т.к. реакция такого автомата может наблюдаться только в определенном подмножестве множества состояний ОД.

2. Описание ПОД

В данной ситуации наиболее удобным является описание ПОД тройкой вида $N = (X, Y, F)$, где X, Y – соответственно входной и выходной алфавит, F – отношение $Y = F(X)$, описывающее поведение системы.

Входной алфавит в этом случае определяется следующим образом: $X = X_{inp} \times X_{mc} \times X_r$, где X_{inp} – множество входных слов, подаваемых на ОД; X_{mc} – множество формируемых внутри ОД управляющих слов, X_r – алфавит состояний операционного автомата.

Для ПОД символы входного алфавита можно представить в виде пары $\langle x_d, x_c \rangle$, где $x_d \in X_d$ – символы алфавита входных данных, а $x_c \in X_c$ – алфавита входных управляющих воздействий (команд). Данные от команд отличаются тем, что при фиксации команды $x_c \in X_c$ для любого символа данных $x_d \in X_d$ активизируется одна и та же последовательность символов $MP_c = \langle x_{mc}^1, x_{mc}^2, \dots, x_{mc}^l \rangle$ ($x_{mc}^i \in X_{mc}, i = \overline{1, l}$), называемая микропрограммой.

Таким образом, существует взаимно однозначное отображение $X_c \longleftrightarrow MP \subseteq X_{mc}^{\max(l)}$, в котором MP соответствует множеству микропрограмм, а $\max(l)$ – максимальной длине микропрограммы. Его существование позволяет привести входной алфавит к виду $X = X_d \times X_c \times X_r$, а функционирование ОД описать следующим образом:

$$Y^* = F(X_r \times X_d), Y^* = Y \times X_r, F = \{f_i\}, DC \subseteq X_c \times F \quad (1).$$

Отношение DC ставит во взаимно однозначное соответствие элементы множеств X_c и $\{f_i\}$. Множество $F = \{f_i\}$ определяет множество команд ПОД, а соответствие DC – механизм их дешифрации.

Приведенные выше рассуждения обосновывают вывод о том, что модель ПОД на системном уровне можно представить как множество его команд и соответствующих им преобразований данных. При этом связь модели ПОД на системном, операционном и функционально-логическом уровне определяется соответствиями:

$$\begin{aligned} \{f_i\} &\xleftrightarrow{DC} X_c \longleftrightarrow MP \subseteq X_{mc}^{\max(l)}, \\ x_{mc}^i &\longleftrightarrow mc_i \subseteq MO, \quad x_{mc}^i \in X_{mc}, \\ mo_j &\longrightarrow s_j, \quad mo_j \in MO, \quad s_j \subseteq S, \end{aligned}$$

где:

- \longleftrightarrow – знак кодирующего отображения;
- mc_i – множество микроопераций, соответствующих микрокоманде x_{mc}^i ;
- MO – множество всех микроопераций;
- s_j – множество аппаратных средств, активизируемых микрооперацией mo_j (предполагается, что эти средства описаны на функционально-логическом уровне);
- S – модель функционально-логического уровня для ПОД;
- все остальные обозначения соответствуют введенным ранее.

3. Модель ПОД

Как уже отмечалось, для описания ПОД на системном уровне достаточно задать систему его команд и соответствующих им преобразований. Использование для этих целей отношений (2) является затруднительным. Это объясняется сложностью описания преобразования $Y^* = F(X_d \times X_r)$ в силу большой размерностью алфавитов X_d и X_r .

Для функционального описания ПОД возможна декомпозиция, в результате которой символ $x_r \in X_r$ представим в виде кортежа $x_r = \langle r_1, r_2, \dots, r_n \rangle$, где $r_i \in A_r = X_d$, $i = \overline{1, n}$. Здесь под A_r понимается внутренний алфавит регистров операционного автомата, а элементы r_i ассоциируются с их состояниями. С учетом вышесказанного выражение (1) преобразуется к виду:

$$Y^* = F(A_r^n \times X_d), \quad Y^* = Y \times A_r^n, \quad F = \{f_i\}, \quad DC \subseteq X_c \times F.$$

Если в соответствие каждому из элементов декартова произведения

$$\underbrace{A_{r1} \times A_{r2} \times \dots \times A_{rn}}_n \times X_d$$

поставить: для A_{ri} , $i = \overline{1, n}$ – регистровую переменную R_i , а для X_d – элемент D_i , называемый операндом, то множество $E_1 = (\bigcup_{i=1}^n R_i) \cup D_i$ является отображением входного алфавита X .

Аналогичным образом можно показать, что $E_2 = (\bigcup_{i=1}^n R_i) \cup D_o$, где D_o обозначает результирующие данные, является отображением для Y^* . Необходимо также определить множество базовых операций

$$O = \{o_j\}, \quad o_j \subseteq E_1 \times E_2, \quad j = \overline{1, m},$$

которое является отображением для $Y^* = F(X)$ в выражении (2).

Целью выполненных отображений является получение множества

$$E = E_1 \cup E_2 \cup O = \left(\bigcup_{i=1}^n R_i \right) \cup D_I \cup D_O \cup O ,$$

достаточного для задания функционального описания ПОД на системном уровне представления модели в виде:

$$F = \{f_i\}, X_c \xleftarrow{DC} f_i, f_i \subseteq E \times E . \quad (2)$$

Таким образом, в соответствие каждой команде $I_i \in I$, где I – множество команд ПОД, может быть поставлен граф

$$G_i = \{E_i, R_i\} (E_i \subseteq E, R_i \subseteq E_i \times E_i).$$

В дальнейшем G_i будет называться абстрактным графом выполнения (АГВ) команды I_i .

Следует отметить, что $G = \bigcup_{i=1}^{|I|} G_i$ является моделью ОД в целом и может быть принят в

качестве модели ПОД. В качестве примера на рис. 1 приведен АГВ некоторых команд МП 1810ВМ86. Такое описание ПОД является обобщением моделей, описанных в [3].

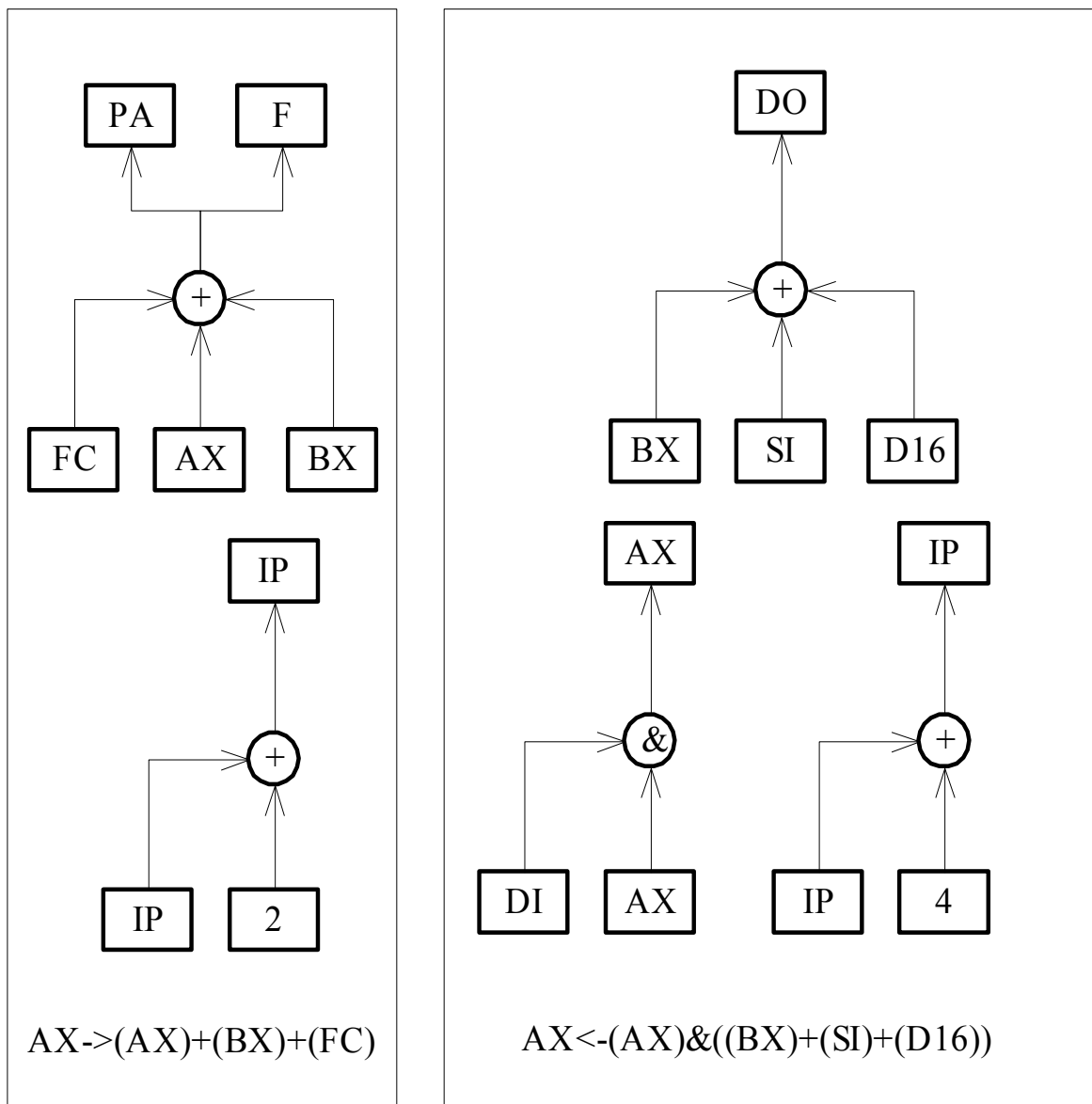


Рис.1. АГВ некоторых команд МП 1810ВМ86.

4. Функциональные неисправности схем

В традиционном подходе к синтезу тестов в качестве неисправности рассматриваются модели неисправности операторов. Это неисправности, связанные только внешними полюсами схем, реализующих эти операторы.

Наиболее часто приходится к ним относят такие классы неисправностей операторов: константные неисправности P_0 , короткие замыкания (отождествления) P_1 и перепутывания P_2 .

Как ясно из названий, к классу константных неисправностей относят неисправности, вследствие которых состояние одного или нескольких полюсов имеют фиксированное состояние, к классу коротких замыканий полюсов относят неисправности, вследствие которых некоторые подмножества полюсов могут находиться в одинаковом состоянии; класс перепутываний определяется неисправностями, характеризующимися перестановкой состояний полюсов.

Заключение

В работе сформулирован подход к построению моделей и преобразований изображений, который отличается от известных тем, что учитывает семантику последних. Это позволяет получать формализованные математические схемы решения задач анализа изображений с учетом их семантики. Формализация математической схемы решения задачи анализа изображений создает предпосылки к созданию унифицированного программного каркаса, использование которого существенно снижает затраты на разработку специализированных программных средств связанных с анализом изображений.

В качестве примера применения сформулированного подхода приведено решение задачи анализа изображений схем в технологических процессах автоматизированного проектирования.

Литература

1. Малышенко Ю.А. Автоматизация диагностирования электронных устройств / Ю.А.Малышенко, В.В.Чипулис, С.Г.Шаршунов ; под ред. В.П.Чипулиса. – Москва : Энергоатомиздат, 1986. – 216 с.
2. Бельхадри Н.Обеспечение надежности компьютерных систем / Н. Бельхадри, А.А. Болдак, Т.М. Чебаненко ; под ред. О.В. Бузовского. – Київ : ТОО «ВЕК», 2007. – 160 с.
3. Андерсон Р.Н. Методы контроля микропроцессорных устройств / Р.Н. Андерсон // Электроника. – 1996. – № 8. – С. 82-89.

Автор статті

Чебаненко Тетяна Михайлівна – кандидат технічних наук, доцент кафедри обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут», Київ, Україна. Тел. +38 067 185 11 85. E-mail: obuza@i.ua

Authors of the article

Chebanenko Tetyana Mykhaylivna – candidate of science (technical), associate professor of Department of Computer Engineering, National Technical University of Ukraine «Kyiv Polytechnic Institute», Kyiv, Ukraine. Tel.: +38 067 185 11 85. E-mail: obuza@i.ua

Дата надходження в редакцію: 29.02.2016 р.

Рецензент: д.т.н., проф. В.П. Тарасенко