

doi: <https://doi.org/10.15407/dopovidi2017.02.016>

УДК 004.8

А. Ф. Кургаев, С.Н. Григорьев

Институт кибернетики им. В.М. Глушкова НАН Украины, Киев

E-mail: afkurgaev@ukr.net, Sergey@Grigoriev.kiev.ua

Модельное проектирование процессора

Представлено академиком НАН Украины А.В. Палагиным

Предложена новая последовательностная структура распределенного сетевого устройства управления и методология модельного проектирования процессора, состоящая в последовательном наращивании структуры процессора в контексте развертывания алгоритма вычислительного процесса. Пользуясь этой методологией разработаны и защищены несколькими патентами оптимальные структуры аппаратных средств представления и интерпретации знаний.

Ключевые слова: процессор, устройство управления, операционное устройство, граф переходов, структура распределенного сетевого устройства управления, методология модельного проектирования процессора.

Постановка задачи. Согласно [1, 2] абстрактная модель компьютера представляется структурой (рис. 1), где A — автомат управления, моделирующий поведение устройства управления, B — операционный автомат, моделирующий поведение операционного устройства и C — автомат (в общем случае, конечное множество автоматов), моделирующий поведение внешних устройств, к которым относят входные, выходные устройства и устройства внешней памяти.

Абстрактная модель процессора является композицией автоматов A и B , где X и Y — множества управляющих входных и выходных сигналов автомата A , которые являются объединением множеств управляющих выходных и входных сигналов автоматов B и C соответственно: $X = X_B \cup X_C$, $Y = Y_B \cup Y_C$. Связь процессора с автоматом C осуществляется обменами векторными входными и выходными сигналами между автоматами B и C под управлением выходных сигналов Y_C автомата A и в данной работе не рассматривается.

Примем, что процесс проектирования процессора начинается с заданной спецификации алгоритма его функционирования и имеет целью получить функциональную структуру процессора максимальной производительности.

На всех этапах проектирования в качестве примера имеется в виду процессор баз знаний (ПБЗ), алгоритмом функционирования которого является алгоритм интерпретации баз знаний ПБЗ, описанный в [2, п. 5.3.4.4].

© А.Ф. Кургаев, С.Н. Григорьев, 2017

Будем рассматривать процесс проектирования процессора в составе двух последовательных этапов: проектирование структуры устройства управления и далее, базируясь на результатах этого этапа, — проектирование структуры операционного устройства. Такая последовательность процесса проектирования отвечает стратегии декомпозиции целей в направлении “от общего к частному”, поскольку при заданной спецификации алгоритма именно на управление ложится основная часть ответственности за реализацию алгоритма, — именно та, что связана с логикой, структурой алгоритма.

1. Проектирование устройства управления. В процессе проектирования устройства управления, как составной части процессора, используют семантически эквивалентные цифровые синхронные автоматы Мили и Мура. Ради конкретности примем, что следующие соображения базируются на абстракции автомата Мура:

$$A = \langle X, Y, S, \mu, \delta, S_0, F \rangle,$$

где X — конечное непустое множество входных сигналов, $X = \{x_1, x_2, \dots, x_p\}$, $|X| = p$; Y — конечное непустое множество выходных сигналов, $Y = \{y_1, y_2, \dots, y_q\}$, $|Y| = q$; S — конечное непустое множество состояний, $S = \{s_1, s_2, \dots, s_n\}$, $|S| = n$, S_0 — начальное состояние; F — возможно пустое множество финальных состояний.

Характеристические функции автомата: функция μ выходов отображает множество состояний S в множество Y выходов, $S \rightarrow Y$; функция переходов δ реализует бинарное отношение $S \times X \rightarrow S$. Итак, характеристические функции определяют в какое состояние $s \in S$ автомат перейдет в следующий $(t + 1)$ -й момент времени и каким будет значение выходного сигнала $y \in Y$ в текущий $(t + 1)$ -й момент времени:

$$s(t + 1) = \delta(x(t), s(t));$$

$$y(t + 1) = \mu(s(t + 1)).$$

1.1. Построение графа переходов. Характеристические функции представляют *табличным* и *графическим* способами. Табличный способ состоит в построении таблицы переходов и выходов конечного автомата, графический способ — в построении орграфа переходов с нагруженными дугами.

Предлагается, в отличие от обычных форм таблиц и графов переходов, базировать процесс проектирования устройства управления на представлении характеристических функций в форме орграфов (и таблиц) переходов с нагруженными не только дугами, но и вершинами.

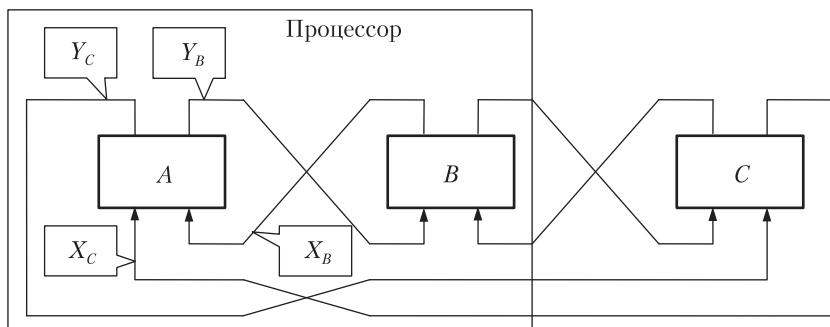


Рис. 1. Абстрактная модель компьютера

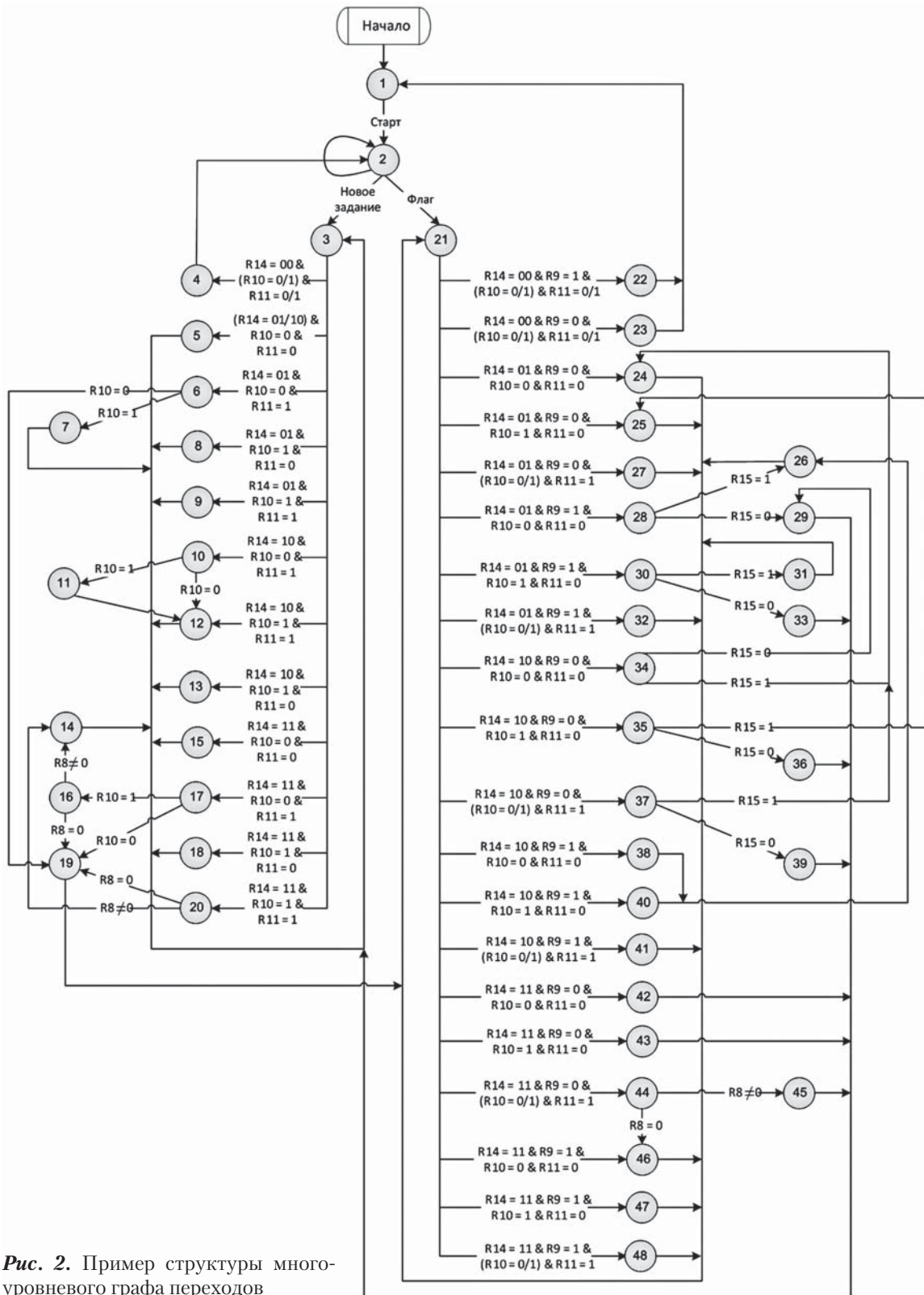


Рис. 2. Пример структуры многоуровневого графа переходов

В новой форме орграфа весом дуги, выходящей из вершины графа для состояния $s(t)$, как и обычно, остается значение входного сигнала $x(t)$, инициирующее переход в состояние $s(t+1)$, а весом вершины становится значение $\mu(s(t+1))$ функции выходов для состояния $s(t+1)$, задающее последовательность действий относительно преобразования информации в операционном устройстве согласно заданному алгоритму.

Граф переходов устройства управления должен отвечать следующим правилам.

1. Множество состояний устройства управления изображается на плоскости в виде вершин графа. Каждая вершина графа помечается именем (или номером) состояния.

2. С каждой вершиной графа связывается ее семантика, являющаяся некоторым кортежем параллельно-последовательных значений выходных сигналов соответствующего состояния устройства управления.

3. Все условные переходы между состояниями устройства управления изображают ориентированными дугами графа. Стабильное состояние изображают инцидентной петлей соответствующей вершины.

4. Каждая дуга, выходящая из некоторой вершины графа, помечается условием перехода по этой дуге в виде булевой формулы на подмножестве входных сигналов устройства управления.

5. Для каждой вершины графа обеспечивается непротиворечивость переходов в смежные вершины за счет однозначности условий перехода.

6. Для обеспечения полноты переходов между вершинами графа могут быть введены дополнительные переходы, которые также помечаются соответствующими булевыми функциями.

7. Семантически эквивалентные вершины графа могут быть совмещены, если имеют тождественные переходы к смежным вершинам. Семантически эквивалентными являются вершины графа, имеющие одинаковые кортежи выходных сигналов.

Граф, удовлетворяющий этим правилам, лишен безусловных переходов, содержит лишь нагруженные дуги. В качестве примера графа переходов, удовлетворяющего сформулированным правилам, на рис. 2 [3–7] приведен граф переходов для алгоритма из [2, п. 5.3.4.4].

Процесс построения многоуровневого графа переходов устройства управления состоит в следующей последовательности событий.

1. Состояние вершины 1 зарезервировано для пассивного ожидания условия “старт”, которое инициирует переход в следующее состояние вершины 2 активизации устройства управления. В состоянии вершины 2 выполняется подготовка операционного устройства процессора к содержательной работе и ожидается значение условия наивысшего уровня, конкретизирующее задание, выполнение которого должно обеспечить устройство управления:

- при положительном значении условия “Новое задание” выполняется переход из состояния вершины 2 в состояние вершины 3, семантикой которой временно становится вся часть исходного алгоритма, инициированная положительным значением условия “Новое задание”, и является продолжением операционных действий узла предыдущей вершины 2;

- при положительном значении условия “Флаг” выполняется переход из состояния вершины 2 в состояние другой вершины, семантикой которой временно становится вся часть исходного алгоритма, инициированная положительным значением условия “Флаг”, и является продолжением операционных действий предыдущей вершины 2.

2. Далее последовательно наращивается иерархия многоуровневой структуры графа переходов, исходя из заданной спецификации таким образом, что для каждого текущего уровня декомпозиции целей:

- согласно заданной спецификации алгоритма формируется логически полное множество условий перехода из каждой вершины текущего уровня декомпозиции в соответствующее множество смежных вершин следующего уровня декомпозиции;
- временной семантикой каждой вершины следующего уровня декомпозиции становится вся часть исходного алгоритма, инициированная положительным значением условия перехода в эту вершину и является продолжением операционной семантики предыдущей вершины для текущего уровня декомпозиции;
- окончательной семантикой каждой вершины текущего уровня декомпозиции становится начало ее временной семантики, заканчивающееся анализом условий перехода из этой вершины текущего уровня декомпозиции.

3. Построение графа переходов завершается или исчерпанием алгоритма, или замыканием всех циклов в результате совмещения семантически эквивалентных вершин графа.

Именно графическое описание автомата позволяет сделать наглядной логику его поведения. В частности, граф переходов на рис. 2 [3–6] является примером структуры иерархического многократно итерационного графа переходов.

1.2. Построение структуры устройства управления. Синтезированная структура графа переходов становится представлением каркаса сетевой однородной структуры устройства управления, а вершины графа переходов — представлением узлов устройства управления, разных по семантическому наполнению, но тождественных структурно (согласно рис. 3) на блочном уровне абстракции в составе подавтомата s_i для каждого состояния s_i и выходной логики $\mu(s_{ik})$.

Блоки каждой вершины s_i графа переходов (каждого узла s_i устройства управления) вписаны в контекст структуры устройства управления таким образом, что:

- входы каждого подавтомата s_i подключены к тактовому сигналу и к выходу (активизации этого состояния s_i) из логики переходов $\delta(x, s_{i-1})$ предыдущего состояния s_{i-1} ;
- выходы подавтомата s_i подключены к входам выходной логики $\mu(s_{ik})$ и к входу логики переходов $\delta(x, s_i)$ из этого состояния s_i , чьи другие входы подключены к входным условиям x ;
- выходы логики переходов $\delta(x, s_i)$ из этого состояния s_i подключены к входам активизации инцидентных вершин графа (смежных узлов устройства управления);
- выходы выходной логики $\mu(s_{ik})$ подключены к входам управления i -го фрагмента структуры операционного устройства процессора, чьи входы и выходы подключены соответственно к выходам и входам других составляющих операционного устройства процессора.

В результате подстановки вместо имен (номеров) вершин графа переходов их структур (согласно рис. 3) с наполнением конкретными значениями составляющих, получим последовательностную структуру распределенного сетевого устройства управления с непрерывным перемещением единственной активности как по уровням иерархии, так и по итерационным контурам.

2. Проектирование операционного устройства. В процессе функционирования устройства управления выстраивается определенная последовательность прохождения состояний автомата (соответственно, и вершин графа переходов), зависящая от начальных и текущих

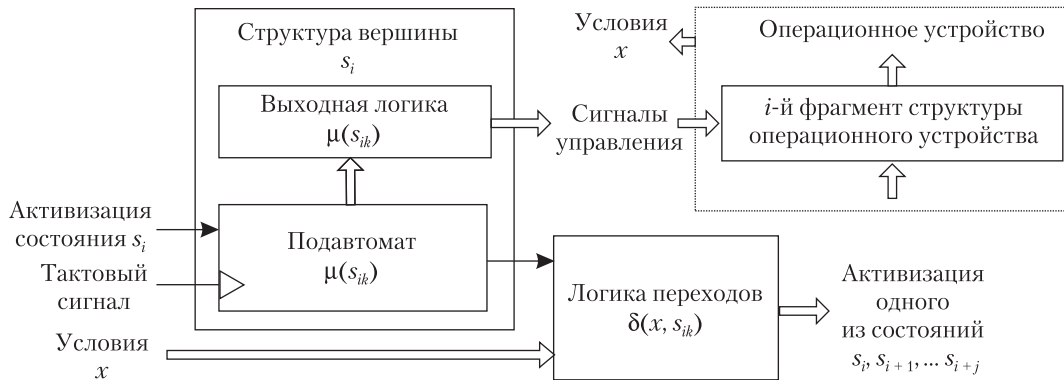


Рис. 3. Структура семантики всякой вершины графа переходов (узла устройства управления) вместе с исходящими дугами

данных, обрабатываемых операционным устройством. Следствием этого становится адекватная последовательность активизации подавтоматов состояний s_i и, в результате, — последовательность соответствующих характеристических функций $\mu(s_{ik})$, где $k \in K$ — множество состояний подавтомата для состояния s_i устройства управления.

Каждый подавтомат s_p , согласно его характеристической функции $\mu(s_{ik})$, порождает вполне определенную, постоянную последовательность выходных сигналов управления некоторым фиксированным i -м фрагментом структуры операционного устройства в процессе выполнения соответствующей конкретной части заданного алгоритма обработки информации.

Для каждого подавтомата s_i , руководствуясь выбранным целевым критерием, можно построить некоторую, в том числе оптимальную, операционную структуру выполнения известной части заданного алгоритма обработки информации. Однако это не решает задачу построения целевого операционного устройства, поскольку оставляет неопределенным процесс синтеза в единое целое этих построенных операционных i -х фрагментов.

Для решения этой задачи предлагается синтезировать структуру операционного устройства процессора согласно синтезированной структуре устройства управления.

Этот шаг является логически вполне оправданным, поскольку, базируясь на спецификации алгоритма функционирования процессора, задача проектирования устройства управления решена именно воспроизведением структуры этого алгоритма в структуре устройства управления. Текущий шаг является продолжением предыдущего: теперь необходимо структуру устройства управления воссоздать в структуре операционного устройства. Этот процесс разлагается на следующие этапы.

1. Подготовительный этап:

- каждой переменной сопоставляется конкретное значение формата данных;
- каждой переменной сопоставляется регистр и значение его структуры;
- со структурой каждого регистра сочетается операционная структура, выполняющая преобразование соответствующей переменной;
- формируется множество и структура разновидностей памяти.

2. Для каждого узла однородного сетевого устройства управления (для каждой вершины графа) строится в имеющемся базисе (регистров, операционных структур, разновидно-

стей памяти) свой i -й фрагмент структуры целевого операционного устройства, который должен выполнять соответствующую часть заданного алгоритма обработки информации.

3. На каждом следующем шаге для очередного i -го узла сети выполняется контекстная вставка i -го фрагмента структуры операционного устройства в структуру операционного устройства, построенную к началу этого шага.

4. Процесс построения структуры операционного устройства завершается полным обходом всех вершин графа.

Для принятого примера алгоритма из [2, п. 5.3.4.4], согласно предложенному процессу синтеза, построены и даны в [3–7] варианты структуры операционного устройства процессора.

Проектирование устройства управления и операционного устройства процессора можно объединить в едином процессе, последовательно наращивая их структуры по этапам декомпозиции начального алгоритма.

Таким образом, разработана новая последовательностная структура распределенного сетевого устройства управления с непрерывным продвижением единственной активности как по уровням иерархии, так и по итерационным контурам.

Сущность предложенной методологии модельного проектирования процессора состоит в последовательном наращивании структуры процессора в контексте развертывания алгоритма вычислительного процесса. Пользуясь этой методологией разработаны и защищены тремя патентами на полезные модели и двумя патентами на изобретения оптимальные структуры аппаратных средств представления и интерпретации знаний.

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

1. Глушков В.М. Теория автоматов и вопросы проектирования структур цифровых машин. *Кибернетика*. 1965. № 1. С. 3–11.
2. Кургаев А.Ф. Проблемная ориентация архитектуры компьютерных систем. Киев: Сталь, 2008. 540 с.
3. Пристрій баз знань: пат. 89644 Україна. МПК G06F15/00; заявл. 05.12.2013. Опубл. 25.04.2014.
4. Комп'ютер баз знань: пат. 89651 Україна. МПК G06F15/00; заявл. 05.12.2013. Опубл. 25.04.2014.
5. Інформаційна машина: пат. 90043 Україна. МПК G06F15/00; заявл. 05.12.2013. Опубл. 25.04.2014.
6. Пристрій баз знань: пат. 109322 Україна. МПК G06F15/18, G06N5/04, заявл. 05.12.2013. Опубл. 10.08.2015.
7. Комп'ютер баз знань: пат. 109325 Україна. МПК G06F15/00, G06N5/04, заявл. 05.12.2013. Опубл. 10.08.2015.

Поступило в редакцию 03.11.2016

REFERENCES

1. Glushkov, V. M. (1965). Automata Theory and problems of designing structures of digital machines. *Cybernetics*. No 1, pp. 3-11 (in Russian).
2. Kurgaev, A. F. (2008). Problem orientation of computer systems architecture. Kyiv: Stal, (in Russian).
3. Pat. 89644 UA, IPC G06F15/00, Knowledge base device, Kurgaiev, O. P., Grygoryev, S. M. Publ. 25.04.2014 (in Ukrainian).
4. Pat. 89651 UA, IPC G06F15/00, Knowledge base computer, Kurgaiev, O. P., Grygoryev, S. M. Publ. 25.04.2014 (in Ukrainian).
5. Pat. 90043 UA, IPC G06F15/00, Data-processing machine, Kurgaiev, O. P., Grygoryev, S. M. Publ. 25.04.2014 (in Ukrainian).
6. Pat. 109322 UA, IPC G06F15/18, G06N5/04, Knowledge base device, Kurgaiev, O. P., Grygoryev, S. M. Publ. 10.08.2015 (in Ukrainian).
7. Pat. 109325 UA, IPC G06F15/00, G06N5/04, Knowledge base computer, Kurgaiev, O. P., Grygoryev, S. M. Publ. 10.08.2015 (in Ukrainian).

Received 03.11.2016

О.П. Кургаев, С.М. Григор'єв

Інститут кібернетики ім. В.М. Глушкова НАН України, Київ
E-mail: afkurgaev@ukr.net, Sergey@Grigoriev.kiev.ua

МОДЕЛЬНЕ ПРОЕКТУВАННЯ ПРОЦЕСОРА

Запропонована нова послідовнісна структура розподіленого мережного пристрою керування та методологія модельного проектування процесора, яка полягає в послідовному нарощуванні структури процесора в контексті розгортання алгоритму обчислювального процесу. Користуючись цією методологією розроблено й захищено декількома патентами оптимальні структури апаратних засобів подання й інтерпретації знань.

Ключові слова: процесор, пристрій керування, операційний пристрій, граф переходів, структура розподіленого мережного пристрою керування, методологія модельного проектування процесора.

A. F. Kurgaev, S.M. Grygoryev

V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kiev
E-mail: afkurgaev@ukr.net, Sergey@Grigoriev.kiev.ua

THE MODEL PROCESSOR DESIGN

A new sequential structure of the distributed network control unit and a methodology of model processor design, which consists in the successive building-up of the processor structure in the context of the deployment of a computational algorithm, is presented. Using this methodology, the optimal structures of a hardware for the knowledge representation and interpretation were developed and patented.

Keywords: processor, control unit, operating unit, transition graph, structure of a distributed network control unit, methodology of model processor design.