

УДК 004.27

КОНСТРУИРОВАНИЕ НЕЧЕТКОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ОБЩЕГО НАЗНАЧЕНИЯ

Краснощёк В.Е.

ОАО «Мотор Сич», г. Запорожье

Целью настоящего проекта является создание архитектуры вычислительной машины и языка программирования общего назначения, в полной мере реализующих парадигму «мягких вычислений», оперирующих в системе нечеткой логики и, вместе с тем, удовлетворяющих критерий полноты по Тьюрингу.

В процессе выполнен анализ подходов к реализации механизмов нечеткой логики в некоторых существующих инструментах и выведены требования к проектируемой вычислительной системе и компилятору.

Рассмотрен круг задач, в которых может быть применена разработка, проанализированы проблемы в реализации и моделировании на основе современной элементной базы. Обсуждаются методы оптимизации архитектуры для вычислительных систем, оперирующих в системе нечеткой логики.

Введение

В настоящее время существует большое количество специализированных нечетких вычислительных систем, находящих применение, в основном, в промышленной автоматике и радиоэлектронных товарах широкого потребления (например, система IntelliSense компании Omron [1]). Зачастую, такие системы создаются для решения какой-то одной задачи либо узкого класса задач.

Вместе с тем, на сегодняшний день не существует ни вычислительных систем в комплексе, ни, отдельно, развитых языков

программирования общего назначения, оперирующих в среде «мягких» вычислений и нечеткой логики. Единственным примером, который удалось обнаружить, является «нечеткий процессор» (fuzzy processor) HEPE98 [2] (Istituto Nazionale di Fisica Nucleare [3]), но так и осталось неясным, как разработчиками позиционируется данная разработка – как процессор общего назначения или же, всё-

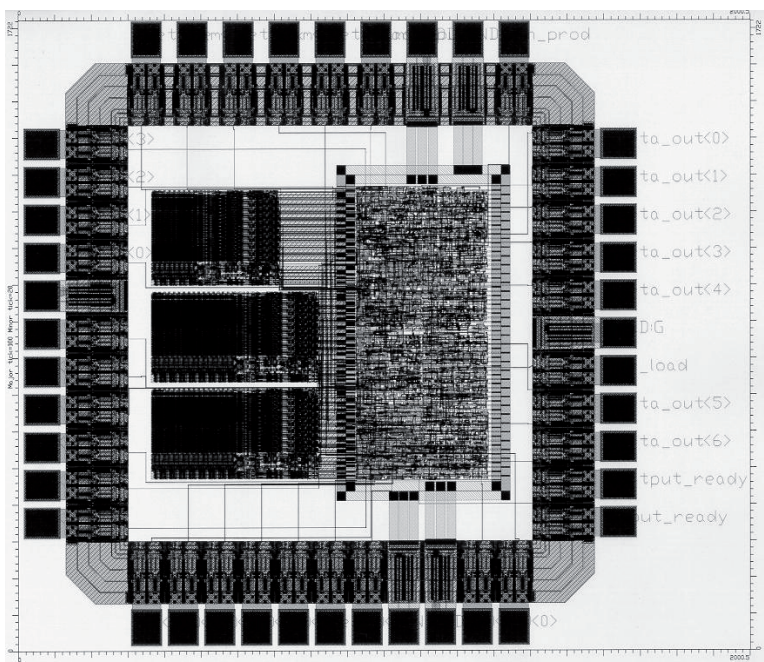


Рисунок 1 – Нечеткий процессор HEPE98

таки, как специализированный?

Основной задачей настоящего проекта является задача создания нечеткой вычислительной системы в составе:

1. Нечеткой ЭВМ общего назначения;
2. Языка программирования высокого уровня (далее – ЯВУ) с естественной поддержкой «мягких» вычислений и операций нечеткой логики, предназначенного для создания программ для вышеупомянутой ЭВМ.

Процесс создание архитектуры нечеткой ЭВМ и соответствующего языка программирования

Прежде чем рассматривать этапы процесса создания нечеткой ЭВМ нужно решить еще одну задачу. Так как вычислительная система, которая позиционируется как вычислительная система общего назначения, должна удовлетворять критерий полноты по Тьюрингу, то нужно определить, удовлетворяет ли данный критерий нечеткая вычислительная система.

Опорной точкой служит теорема об универсальном нечетком аппроксиматоре [4]. Согласно данной теореме, любая математическая система может быть аппроксимирована системой на нечеткой логике. В таком случае процесс обработки данных, протекающий в вычислительной системе можно представить в виде функции отображения значений входных переменных на значения выходных:

$$f : X \rightarrow Y \quad (1)$$

В качестве аппроксимируемой вычислительной системы возьмем систему, заведомо удовлетворяющую критерий полноты – машину Тьюринга. Любую программу, выполняющуюся на данной машине можно представить как последовательность

$$\begin{aligned} f_1 : X_1 &\rightarrow Y_1 \\ f_2 : X_2 &\rightarrow Y_2 \\ &\dots \\ f_i : X_n &\rightarrow Y_n \end{aligned} \quad (2)$$

где f_i – i -я функция преобразования данных – одна из операций, реализуемых вычислительной системой; $i = 1, 2 \dots m$, где m – число операций в операционном алфавите системы. При такой постановке задача аппроксимации процесса преобразования данных сводится к аппроксимации каждой функции преобразования из операционного алфавита, т.е. нахождения соответствующих конструкций нечеткой логики, реализующих f_i . Так как алфавит выбранной вычислительной системы удовлетворяет критерию полноты, то аппроксимирующая его нечеткая вычислительная система также будет удовлетворять

данный критерий. Более строгое доказательство планируется привести в отдельной статье.

Под созданием архитектуры нечеткой ЭВМ общего назначения понимаются следующая последовательность:

1. Создание иерархической математической модели устройства. На данном этапе будет проверяться корректность и реализуемость операций нечеткой логики с учетом требований и ограничений, заложенных в проект:
 - разрядность представления опорных точек функций принадлежности;
 - методы дефаззификации;
 - методы выполнения базовых арифметических операций;
 - представление нечеткого числа;
 - представление функции принадлежности.
2. Описание архитектуры при помощи одного из формальных методов (TLA, связанные FSM, описание архитектуры на одном из функциональных языков программирования). Включение данного этапа в цикл разработки позволит сократить число возможных логических ошибок и ошибок планирования на верхнем уровне, а также упростит отладку на этапе реализации при помощи HDL.
3. Решение платформенно-ориентированных задач, т.е. задач реализации архитектурных конструкций с учетом особенностей выбранной платформы прототипирования.
4. Реализация созданной архитектуры на базе выбранной платформы, отладка.

В качестве платформы прототипирования рассматриваются два варианта: FPGA и PSoC.

Описание грамматики нечеткого ЯВУ представляет собой более простую задачу. Имея базовый набор операций, реализовываемых вышеупомянутой архитектурой ЭВМ задача сводится к:

1. Описанию грамматических конструкций языка и их верификации (что может быть реализовано автоматически

при условии выполнения описания на одном из функциональных языков, вроде Haskell). В качестве основы для построения грамматики может быть выбран один из существующих проблемно-ориентированных языков, напр. FCL либо входные языки математических пакетов, напр. FuzzyJess или FISLAB;

2. Построение транслятора.

Применения настоящей разработки

Сферы применения вычислительной системы, разрабатываемой в рамках настоящего проекта:

1. Системы автоматического управления. Нечеткие вычислительные системы позволяют ускорить процесс разработки систем управления за счет более быстрого и удобного переноса знаний экспертов в управляющие алгоритмы без промежуточного построения точной математической модели [5] (например, на ранних этапах прототипирования). При чем, в отличие от специализированных нечетких контроллеров, настоящую систему можно будет применять в широком классе задач управления, а за счет полноты по Тьюрингу, функциональность реализованных систем управления может быть расширена для совместного выполнения других, вспомогательных, задач;
2. Задачи идентификации и управления в условиях неполноты параметров, отсутствия точной модели или в условиях плохой наблюдаемости. Так как нечеткая вычислительная система может оперировать не самими параметрами, а их оценками (и алгоритмы для нее также могут быть составлены на основе оценок входных данных), то процессы управления могут быть реализованы с приемлемой точностью даже в таких условиях;
3. Задачи искусственного интеллекта и моделирования процессов мышления.

Также открывается достаточно широкое поле для

исследования в области анализа и построения алгоритмов.

Литература

- [1] Omron Healthcare – FAQ. – Режим доступа: http://www.ims.com.ua/faq_ton.shtml#16
- [2] HEPE98: The first 133 MHz Fuzzy Processor. – Режим доступа: <http://www.bo.infn.it/~falchier/hepe98.html>
- [3] Istituto Nazionale di Fisica Nucleare. – Режим доступа: <http://www.infn.it/indexen.php>
- [4] Kosko B. Fuzzy Systems as Universal Approximators // IEEE Trans. on Computers. 1994. Vol. 43. №11. P.1329 1333.
- [5] Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2003. 736с.