

Создание адаптивной параллельной вычислительной архитектуры

Авторы работы:

Дания Юнусова, Юрий Герасимов, Арслан Юсифов, Артур Атласов - студенты кафедры

«Автономные робототехнические системы» Высшей Школы ИТИС, КФУ.
Александр Чичигин- научный руководитель рабочей группы, ассистент кафедры

«Автономные робототехнические системы» Высшей Школы ИТИС, КФУ.

Тезисы:

Целью работы является создание нового вида вычислительной архитектуры, пригодного для оптимизации и синтеза программ по спецификации. В рамках этого проекта мы реализуем работу с цифровыми схемами. Для достижения поставленной цели мы должны выполнить следующие задачи:

- Построение и моделирование цепей из логических элементов
- Синтез схем на основе заданной задачи
- Создание IDE

Примечание [1]: Ну и где список задач?

Для работы с логическими схемами необходимо научиться их эмулировать. Позже планируется перенести эмуляцию на микроконтроллеры. Переход от эмуляции к оптимизации будет выполнен с помощью генетических алгоритмов. Следующий этап – синтез логических схем по заданной спецификации.

Для выполнения проекта был выбран язык C++, так как он позволяет перенести приложение на микроконтроллеры с минимальными усилиями. В качестве среды разработки используется Eclipse, поскольку сотрудники используют разные операционные системы. Так же, для создания кроссплатформенных приложений используется фреймворк Qt. Для моделирования логических схем мы остановились на библиотеке DEVS.

DEVS – Discrete EVent Systems или Дискретные Событийные Системы.

Поведение системы в формализме DEVS описывается при помощи задания множества событий.

X - мн-во входов

Y - мн-во выходов

S - мн-во состояний системы

$\tau_a: S \rightarrow T$ - функция, определяющая время жизни каждого состояния

δ_{ext} - функция внешних воздействий, определяет влияние входных данных на состояние системы

δ_{int} - функция внутренних воздействий. Определяет изменения внутри системы (при достижении временной отсечки);

L - функция вывода. Определяет, как система генерирует выходные события (при достижении временной отсечки)

Логические схемы моделируются в виде DEVS-систем по следующим правилам.

1) В системе событиями являются только изменения уровней входных сигналов схемы. Всё остальное время состояние системы остаётся неизменным.

2) Изменения уровней сигналов дискретны и мгновенны. То есть если изменения происходят, они происходят моментально.

3) Все элементы схемы рассматриваются как тактируемые. Этот общепринятый принцип применяется для более удобного моделирования схем.

4) Следовательно, изменение состояния компонентов происходит тогда и только тогда, когда приходит тактирующий сигнал. Это следует из предыдущих 3 правил.

5) Комбинационные элементы моделируются в виде компонент с задержкой 0. Это было сделано нами для удобства моделирования этих элементов

На первом этапе для упрощения реализации нами была выполнена классификация логических элементов. Также мы перенесли эти схемы в среду симулирования Logically

Примечание [2]: Этот пассаж я оставляю на Вашей совести.

Примечание [3]: Не совсем понял...

