

<http://model.exponenta.ru/k2/>

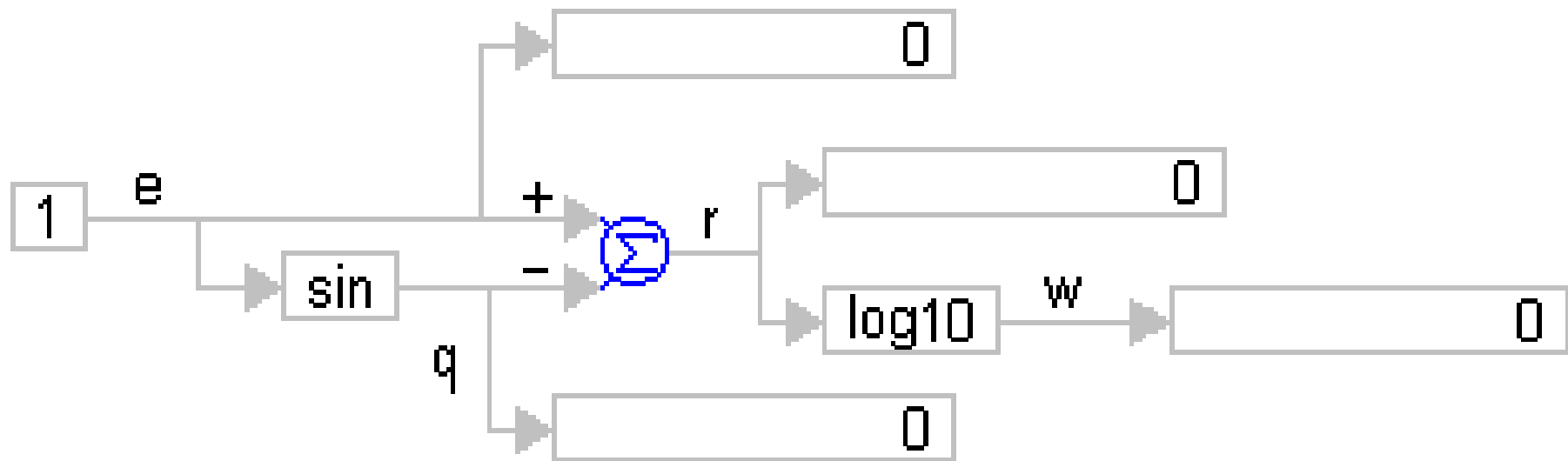
О возможности  
распараллеливания вычислений  
при мультидоменном физическом  
моделировании на базе  
бинаправленных графов

Клиначёв Николай Васильевич,  
к.т.н, каф. «Электротехника»  
и каф. «Информатика» ЮУрГУ;  
Клиначёва Наталья Васильевна,  
аспирант каф. «Электротехника» ЮУрГУ

# Архитектура математических ядер (Simulink, VisSim, EWB, **K2.SimKernel**)

- **Полиморфный класс** с виртуальной математической функцией
- **Классы потомки** полиморфного класса, индивидуально реализующие математическую функцию — математическая библиотека ядра
- **Массив объектов** порожденных от классов математической библиотеки — модель пользователя — **MathBlock[i]**
- **Алгоритмы** для работы с массивом объектов — численные методы

# Принцип поточного управления ИСПОЛНЯЕМЫМ КОДОМ



Наблюдаем 4 каскада исполняемых функций

# Процедура симуляции модели

## *Вызов потока (шаг симуляции)*

```
for (i=0; i < amntBlocks; i++) {  
    MathBlock[i]->Calc(); // вызов математических функций  
}
```

## *Аспекты итерационных вызовов потока*

- Блоки вызываемые один раз за процедуру симуляции: **const**.
- Блоки вызываемые один раз за шаг симуляции: **Delta(t-dT)**, **1(t-dT)**, **squareWave**, **prbs**, **noise**, **gaussian**, **pulsTrain**; + все серверы визуализации.
- Блоки вызываемые один раз за субшаг симуляции (не участвующие в итерациях): **v(t-dT)**, **a(t-dT)**, **sawtoothWave**, **triangleWave**, **sinWave**, **SimKernelVar**.
- Все остальные блоки, как правило, вызываются в каждой итерации.




# Технологии моделирования

- **Моделирование на основе направленных графов** — блок-схемы; типовые динамические звенья — Simulink
- **Моделирование на основе бинаправленных графов** — схемы физические принципиальные; модели RLC и EJ-элементов — EWB
- **Моделирование на основе гибрида из направленных и бинаправленных графов**



# Возможности распараллеливания вычислений

- Распараллеливание некоторых многостадийных методов интегрирования —  $\times 2$
- Распараллеливание при разделении математического ядра (разделении модели) —  $\times 10.. \times 30$
- Распараллеливание каскадов функций одной модели —  $\times 50.. \times 500$



# Аспекты распараллеливания каскадов функций модели


- Масштаб модели из RLC и EJ-элементов не влияет на количество каскадов функций — 14..25
- Количество каскадов функций увеличивается при последовательном включении элементов
- Внедрение в схему физическую принципиальную элементов направленного графа, как правило, не увеличивает количество каскадов в модели



# Требования к многоядерному процессору и его драйверу

- Каскадная организация групп (или переключений) вычислительных ядер
- Доступ к настройкам синхронизации работы каскадов
- Доступ к настройкам синхронизации работы ядер в каскаде
- Доступ к настройкам режима энергопотребления отдельных ядер
- Наличие ведущих по вычислительной способности ядер (50 / 500 — LC / REJ)
- Допустима узкая специализация ядер





# Виды требуемых вычислительных ядер ЦПУ

- Специализированные под операцию и универсальные ядра
- Ядра, переключающиеся и непереклюющиеся между модулями кода (каскадами)
- Ядра, адаптирующиеся к повторяемому циклу ветвлений
- Ядра, обслуживающие один и разные потоки процесса

Документация математического ядра **K2.SimKernel**

Видеозапись работы с библиотекой **SimLib4Visio**



Видеозапись