Ростех Объединенная приборостроительная корпорация

«Обожаю, когда мне говорят, что я не смогу сделать это… Потому что всю свою жизнь вижу, как они ошибаются, полагая, что я даже не попытаюсь!»

Тед Тернер

Высокоскоростная коммуникационная сеть «Ангара»: от идеи до продукта

Главный конструктор Симонов А.С.

1948 1953 1968 1990 1998 2006 2010

SKB-245 NICEVT JSC NICEVT

- **STRELA**
  - Ural-1
    - M-20
      - M11
    - M-205
      - M-220
- **ES-EVM**
  - ES-1191
    - ES-1195
- Clusters
  - SCI (Dolphin)
  - IB (Mellanox)
  - TRIAD
- **ANGARA**

Содержание

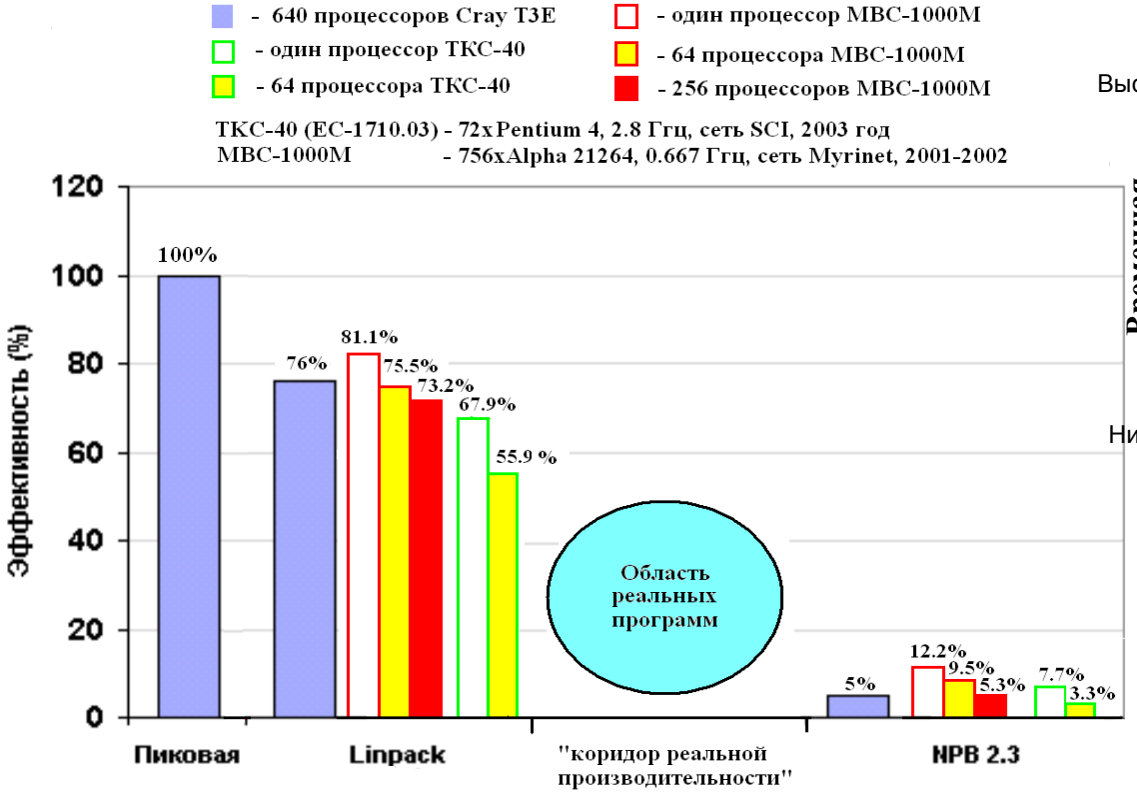
1. Формирование идеи
2. Анализ зарубежного опыта
3. Имитационное моделирование
4. Спецификация
5. Разработка
6. Верификация
7. Макетирование
8. Проектирование СБИС
9. Разработка сетевого оборудования и системного ПО
10. Оценочное тестирование
11. Перспективы

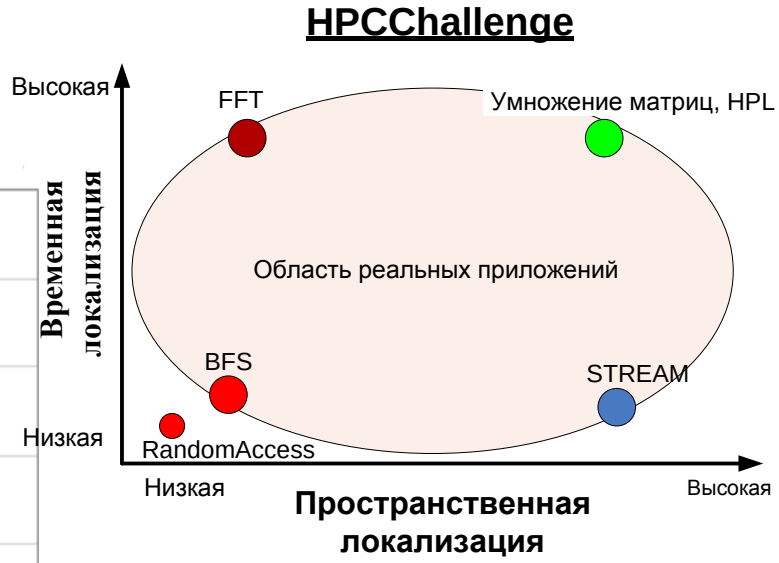
«Нет ничего более рискованного, чем не рисковать»
Ларри Эллисон (Oracle)

Формирование идеи

Формирование идеи

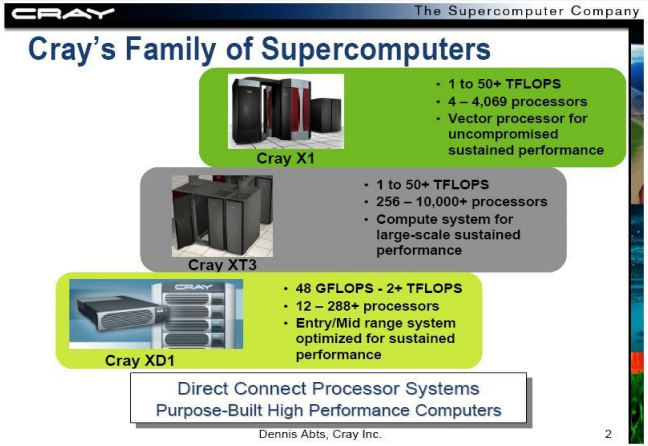
Анализ производительности суперкомпьютерных приложений

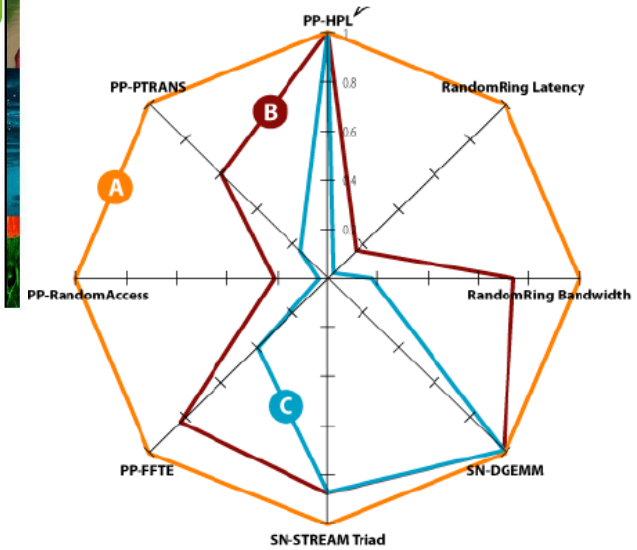


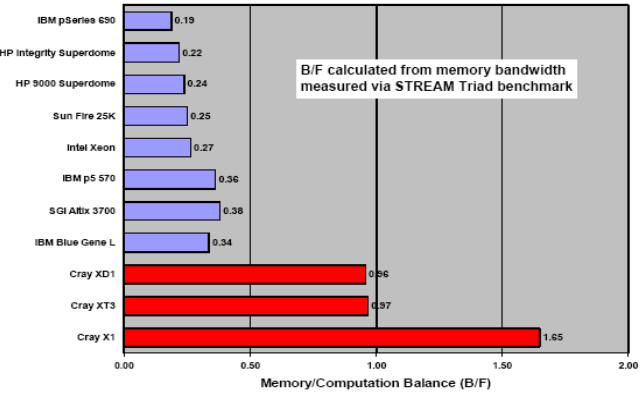


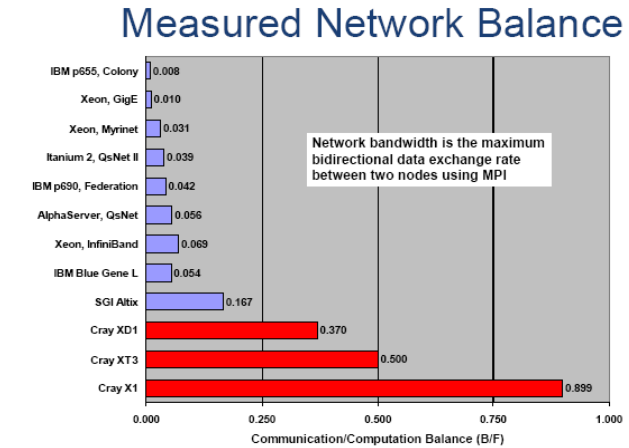
Формирование идеи

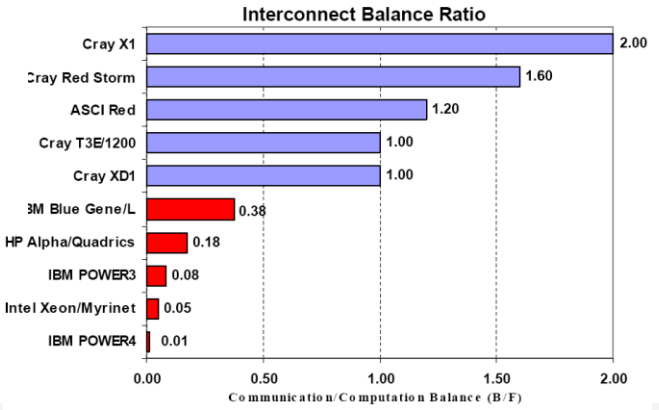
Показатели сбалансированности вычислительных систем





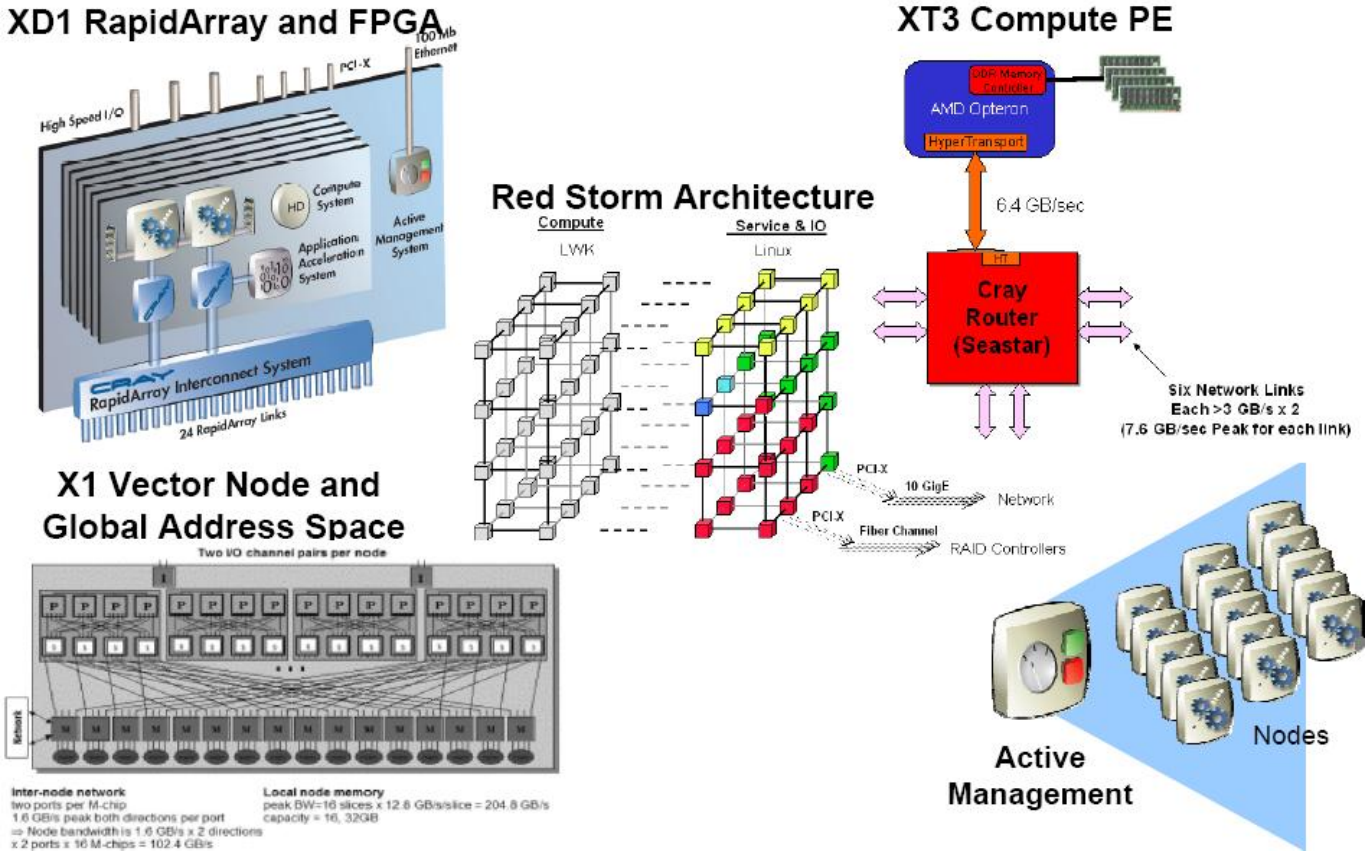






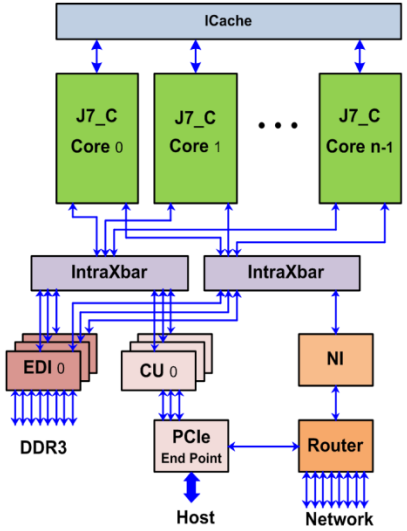
Формирование идеи

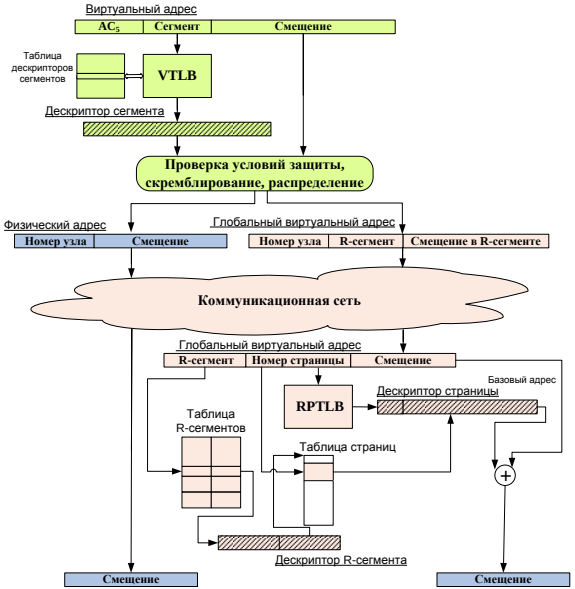
Многообразие принципов функционирования систем

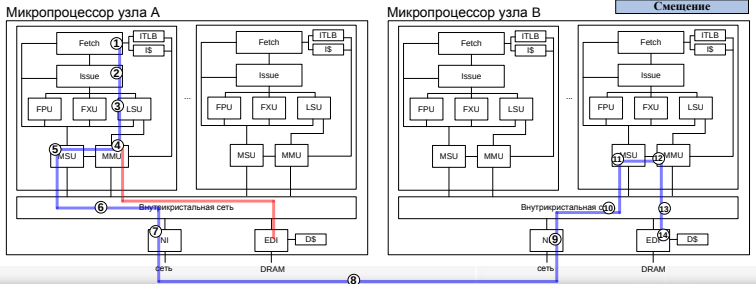


Архитектурные принципы проекта «Ангара» (Л.К. Эйсымонт):

- Мультитредовость (Multithreading)
- Глобально адресуемая память (Global Address Space)
- Поддержка мелкозернистой синхронизации
- Поддержка активных сообщений

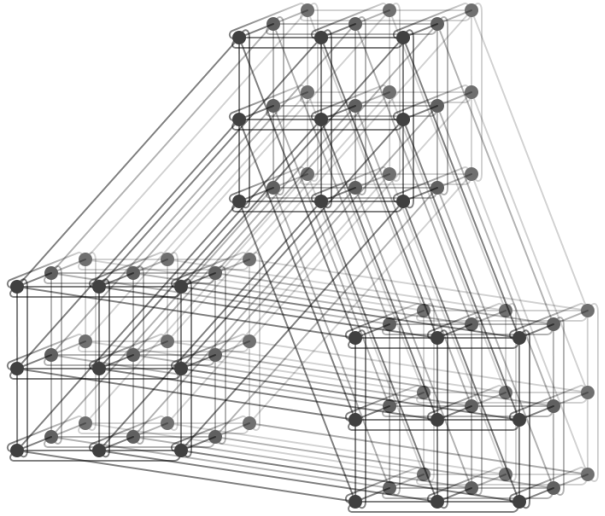






Формирование идеи

Цели создания коммуникационной сети



2006 год

- создание и возрождение российской вычислительной техники
- перспектива перекрытия каналов поставок импортной техники
- лучшее продается в Россию только тогда, когда устареет

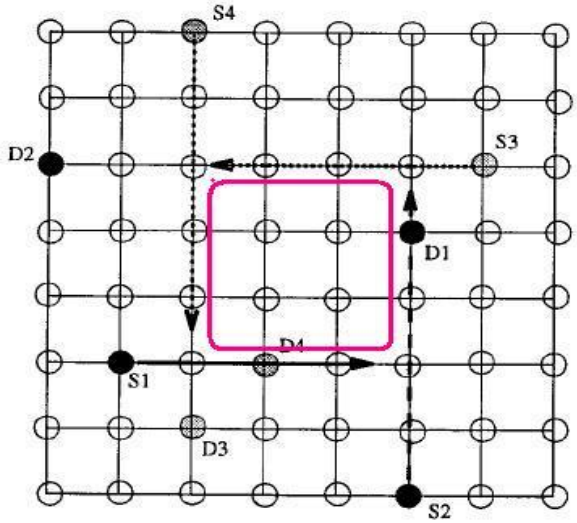
Формирование идеи

Задачи при создании коммуникационной сети

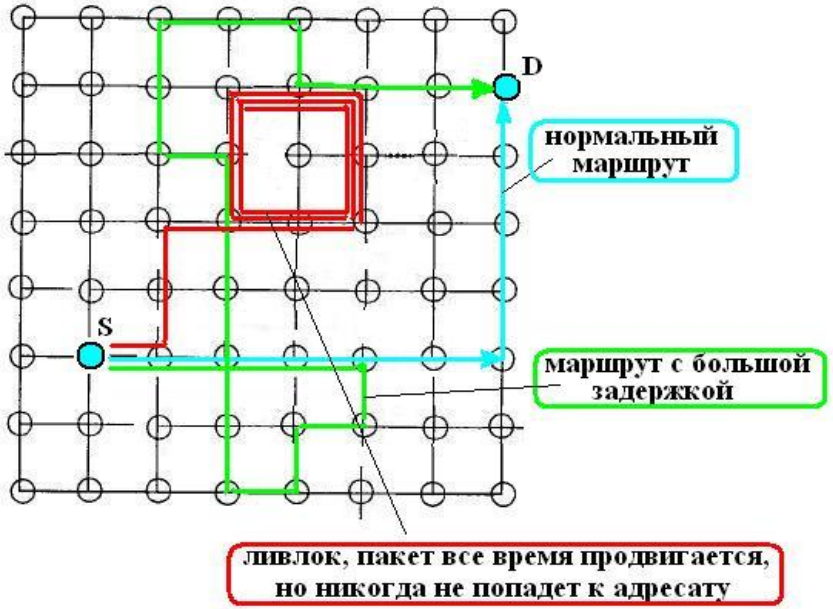
- **правильное функционирование**
  - исключение дедлоков и ливлоков (deadlock & livelock routing);
  - обход перегрузки сети (adaptive routing);
  - обеспечение отказоустойчивости (fault-tolerance),
- **эффективность**
  - низкая *latency* при передачах типа “точка-точка”;
  - высокая пропускная способность сети, *throughput*, для разных профилей взаимодействий типа “коллективных”, например:
    - полностью случайный,
    - бисекционный,
    - барьерный (синхронизация типа barrier и eurica),
    - reduce / broadcast / all-to-all,
    - специфические (shuffle, bit-wise …)

Анализ зарубежного опыта

Дедлоки



Ливлоки



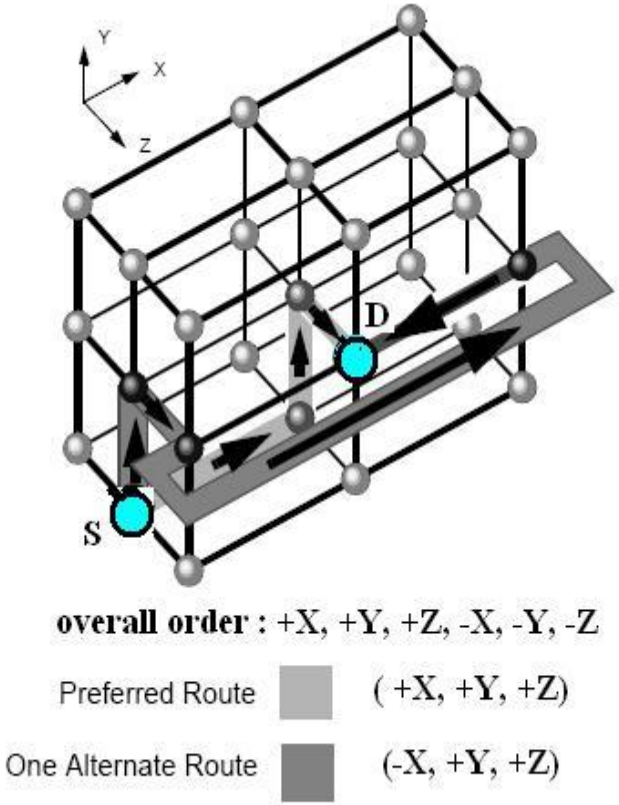
Преодоление дедлоков

dimension order routing – сначала строго по X, потом – по Y, потом – по Z

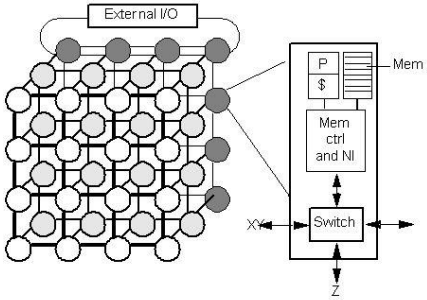
Преодоление ливлоков

минимальная маршрутизация

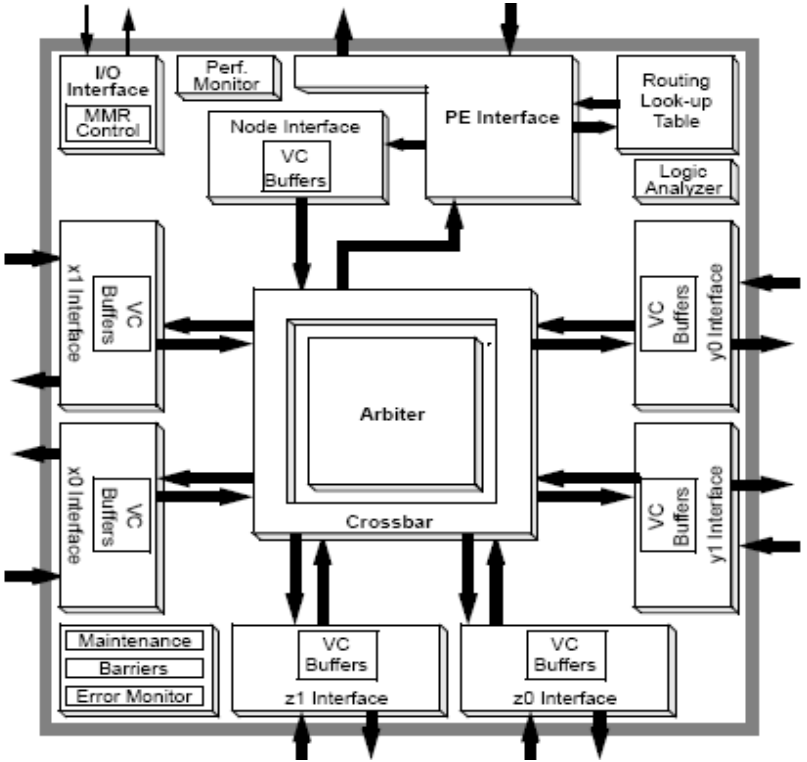
Адаптивная маршрутизация – обход перегруженных участков сети



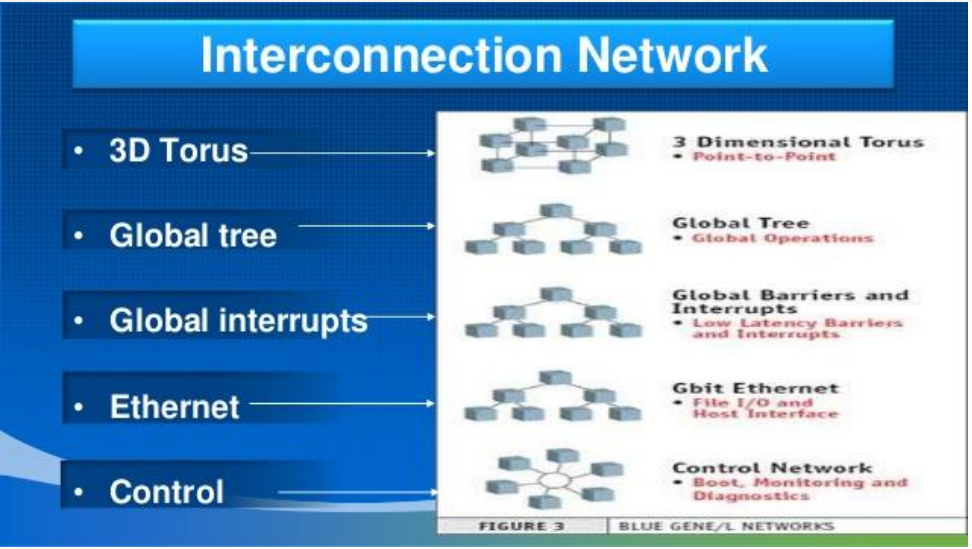
Example: Cray T3E

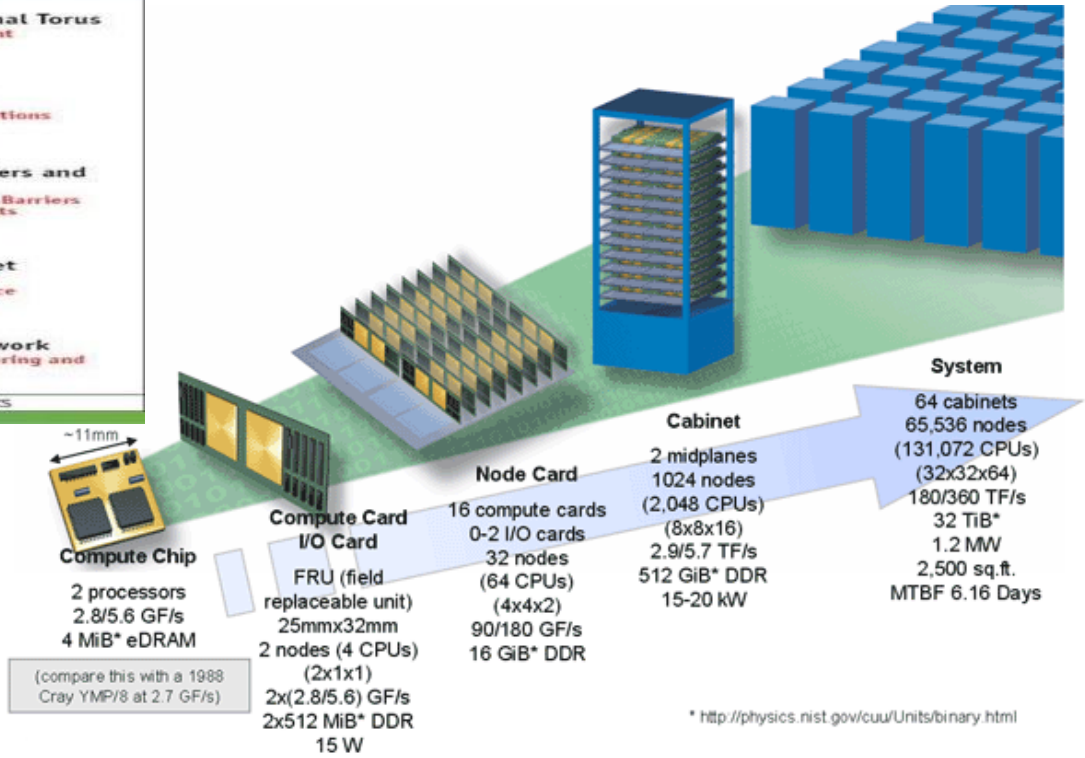


- Scale up to 1024 processors, 480MB/s links
- Memory controller generates comm. request for nonlocal references
- No hardware mechanism for coherence (SGI Origin etc. provide this)



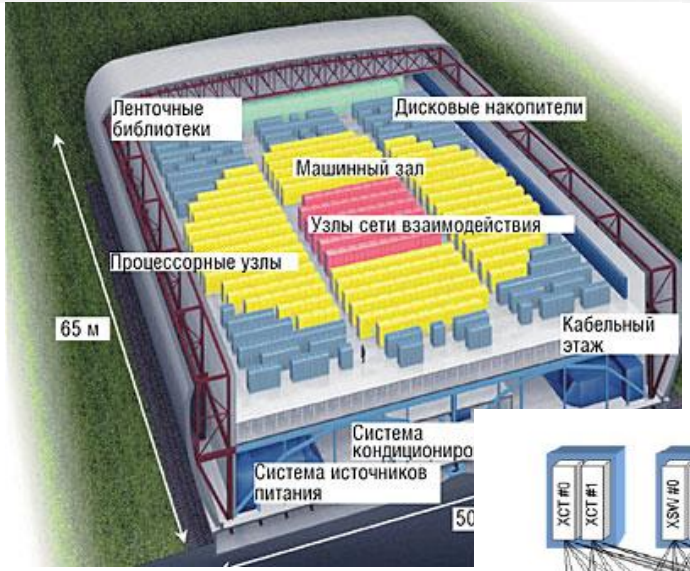
Анализ зарубежного опыта
IBM BlueGene/L





Анализ зарубежного опыта

Earth Simulator



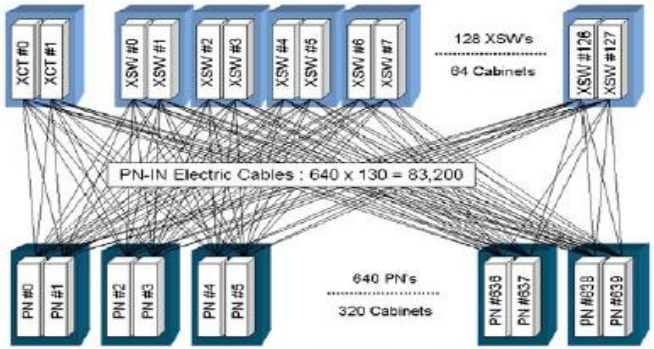


Figure 2.5. Connection between Cabinets (Courtesy JAMSTEC)

Анализ зарубежного опыта

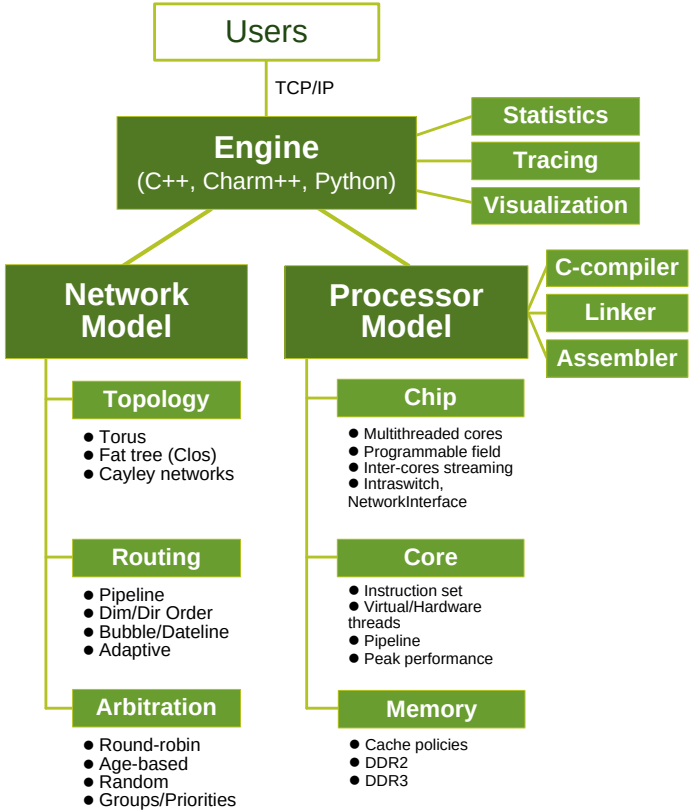
Выводы:

1. Посмотрев по сторонам можно избежать многих ошибок, уже совершенных до Вас
2. Зарубежные научные группы достаточно открыты, но информации, как правило, оказывается недостаточно
3. Важно не просто разобраться, как это сделано, а понять **почему** сделано именно так

Имитационное моделирование

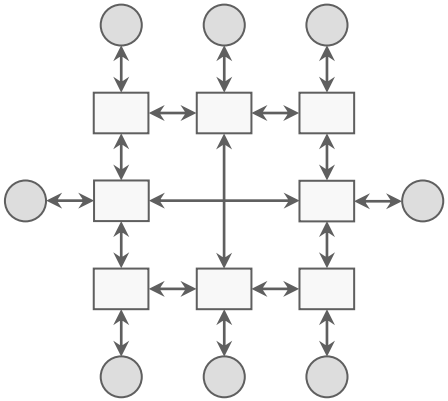
Имитационное моделирование

- Потактовая модель на языке Charm++
- Используется:
  - для оценки производительности и верификации разрабатываемой в АО «НИЦЭВТ» коммуникационной сети
  - для исследования новых архитектур
- Масштабирование производительности модели до 256 узлов суперкомпьютера «Ломоносов»



Коммуникационные сети

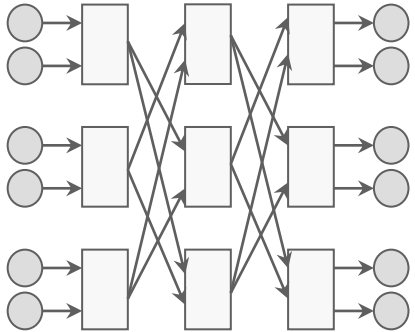
Direct Network



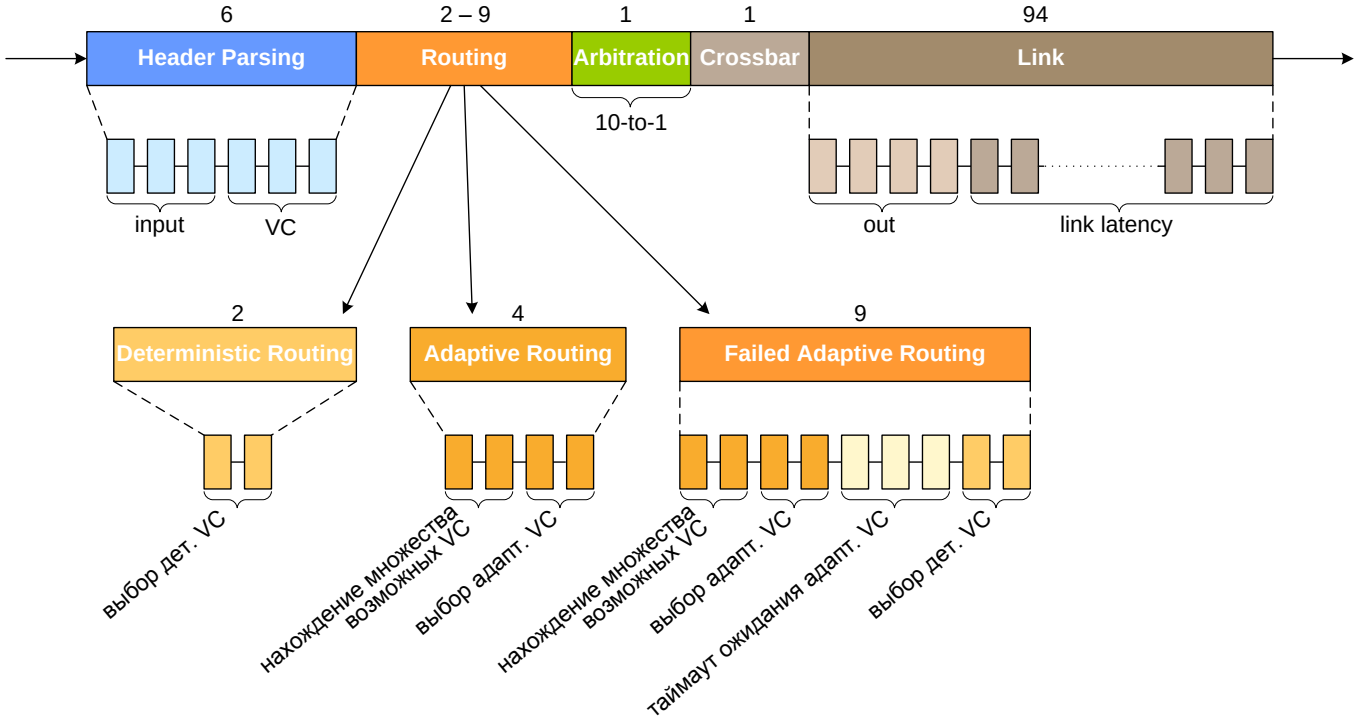
- *Решётка*
- *Тор*
- *Гиперкуб*
- *Сети Кэли*

Indirect Network

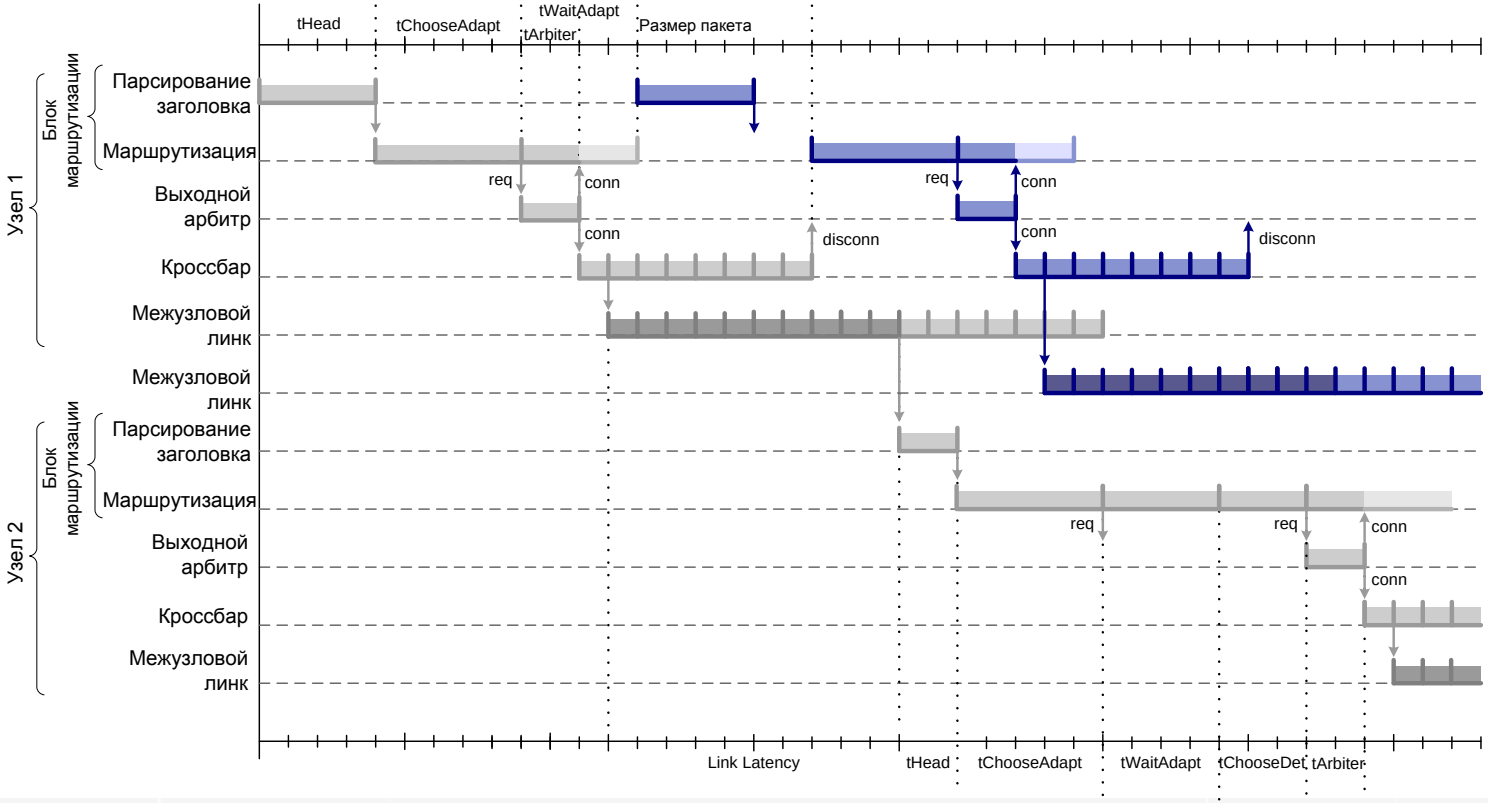
- *Fat Tree*
- *Сети Клоса*



Стадии маршрутизации

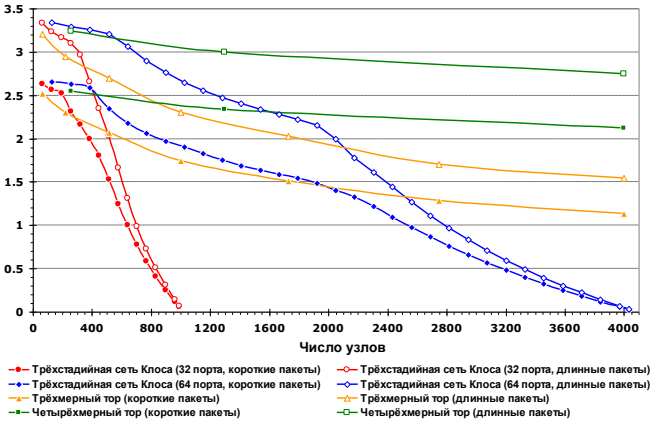


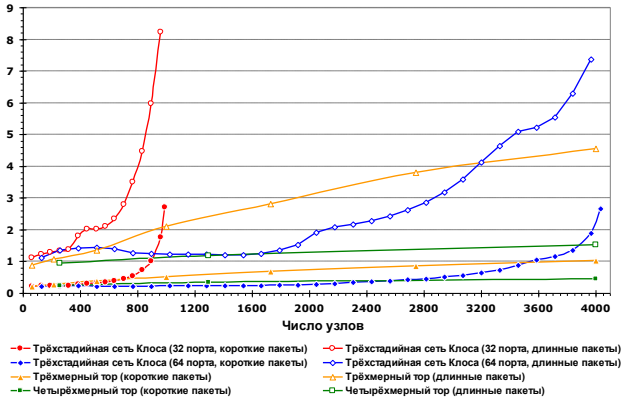
Временная диаграмма передачи адаптивного пакета

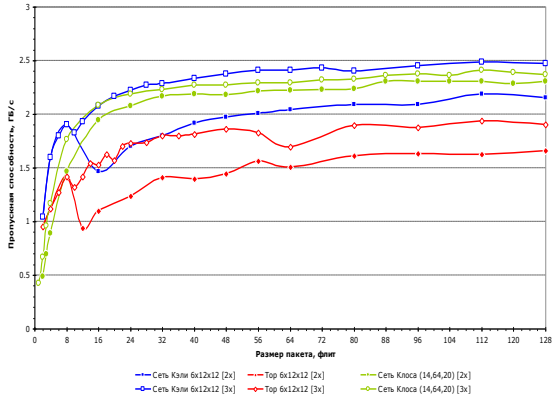


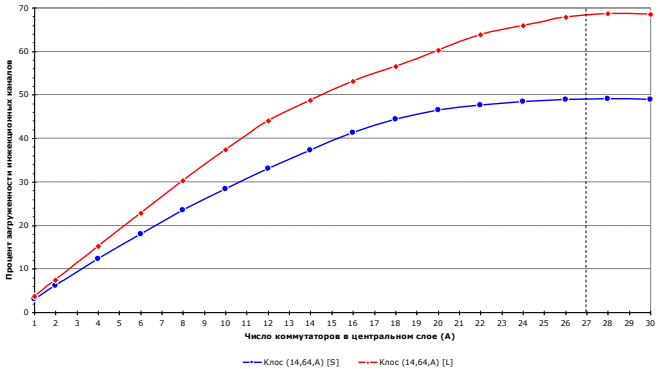
Имитационное моделирование

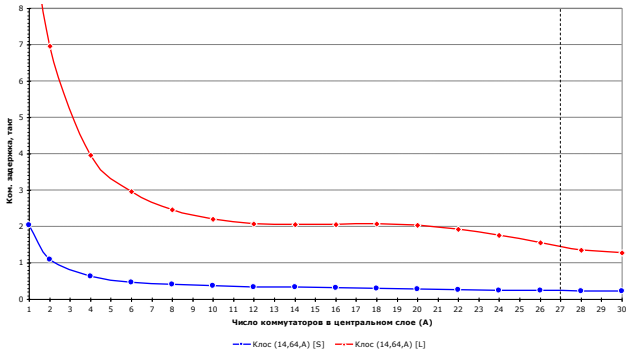
Результаты имитационного моделирования

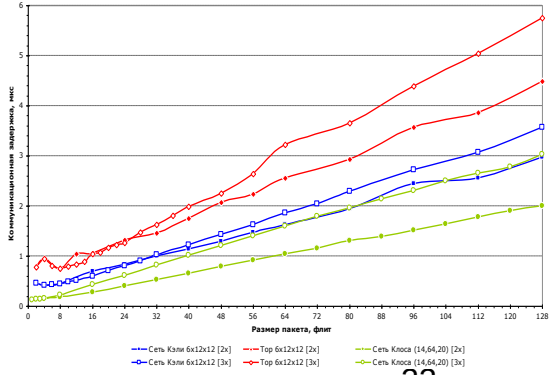












Выводы:

1. Сложность современных систем столь велика, что их создание «с нуля» - нетривиальная задача, которую нельзя решить устаревшими методами
2. Моделирование позволяет глубже понять тонкости работы системы, оценить различные аспекты взаимодействия составных частей
3. Хорошо спроектированная система моделирования позволяет оценить характеристики системы до ее создания, выявить и устранить «узкие» места
4. В идеальном случае имитационная модель должна стать эталонной моделью в системе верификации

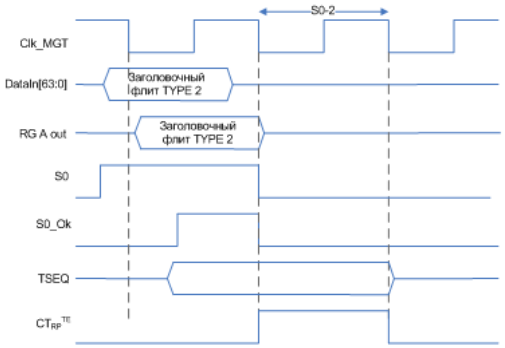
Спецификация

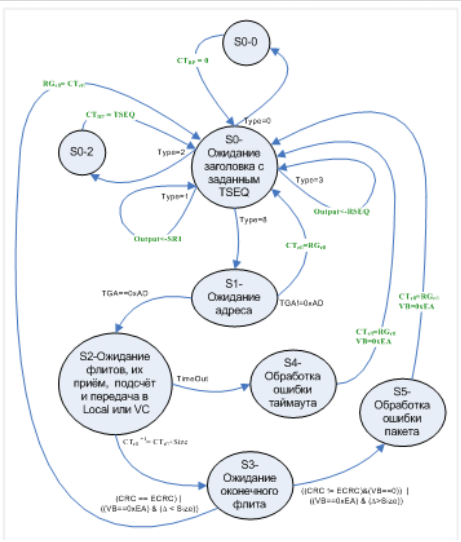
Спецификация (технический проект)

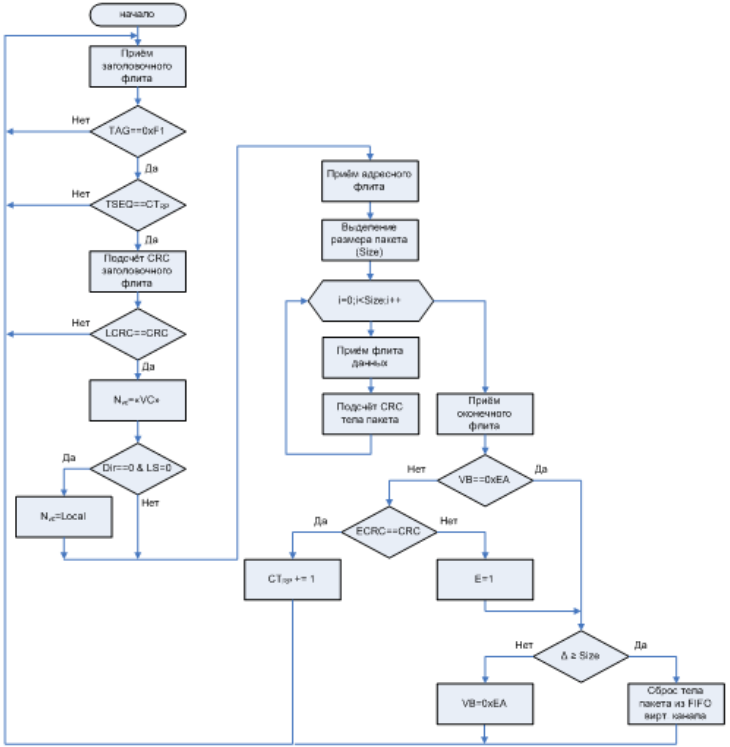
Что требовалось

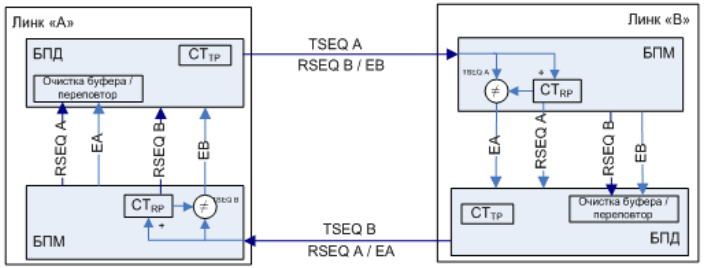
- По аппаратуре:
  - Общая структурная схема со связями
  - Описание функциональных возможностей блоков, в т.ч. алгоритмы работы, диаграммы состояний управляющих автоматов и пр.
  - Требования к интерфейсам между блоками
- По программному обеспечению:
  - Программная модель (перечень программно доступных ресурсов с описанием порядка доступа, управляющих значений и пр.)
  - Требования по реализации, стек ПО, временные параметры и пр.

Например, так:









Что получили:

a) При $j_{in} = 0$ (инжекция).

$$(r_{j,esc} \leftarrow (j = j_{det}), [(r_{j,esc} \leftarrow (j = j_{det})].$$

$r_{j,hp}$	0	$r_{j,hp}$	0
$r_{j,adapt}$	aa	$r_{j,adapt1}$	aa
			aa
		$r_{j,adapt2}$	

b) При $v_{in} = esc$.

$$(r_{j,esc} \leftarrow (j = j_{det}), [(r_{j,esc} \leftarrow (j = j_{det})].$$

$r_{j,hp}$	0	$r_{j,hp}$	0
$r_{j,adapt}$	aa	$r_{j,adapt1}$	aa
			aa
		$r_{j,adapt2}$	

c) При $v_{in} = adapt$ [$v_{in} = adapt1$ или $v_{in} = adapt2$].

$$(r_{j,esc} \leftarrow (j = j_{det}), [(r_{j,esc} \leftarrow (j = j_{det})].$$

$r_{j,hp}$	0	$r_{j,hp}$	0
$r_{j,adapt}$	1	$r_{j,adapt1}$	1
			1
		$r_{j,adapt2}$	

Программисты

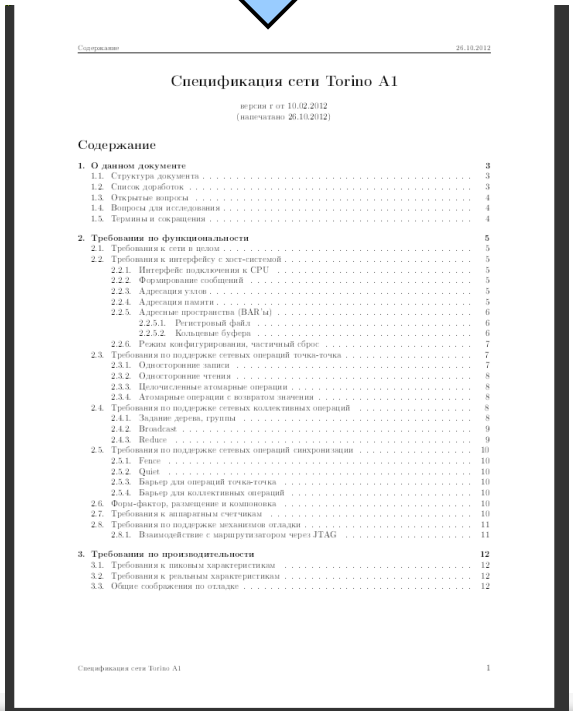
Прикладные программисты

Системные программисты

Архитекторы:

1. **Различные виды операций**
2. **Коллективные операции**
3. **Синхронизация**
4. **Виртуализация**
5. **Быстро**

Главный конструктор



Содержание 26.10.2012

Спецификация сети Torino A1

версия r от 10.02.2012
(напечатано 26.10.2012)

Содержание

Спецификация сети Torino A1 1

Аппаратчики

Эксплуатанты

Верификаторы

Разработчики:

1. **Тогда придется увеличить пакет**
2. **Тогда придется отказаться от большого числа процессов**
3. **Мы не уложимся в план**
4. **Это технически нереализуемо**

- Каждая группа специалистов понимает задачу по своему, в собственной «системе координат»
- Налаживание интерфейса между специалистами различного профиля – задача нетривиальная
- Формирование спецификации специалистами одного профиля с последующей трансляцией в терминологию специалистов других профилей приведет к перекосам и противоречиям, т.к. будут учтены требования одной группы в ущерб требованиям других групп
- Требования к изделию со стороны специалистов разного профиля, как правило, противоречивы

Выводы:

1. Создание спецификации нельзя доверять какой-то одной группе инженеров, необходимо создавать совместную группу из инженеров разных специальностей (аппаратчиков, программистов, верификаторов, системщиков, прикладников, эксплуатантов…)
2. Спецификация – это всегда баланс противоречивых требований, при этом важно выделить один или несколько ключевых параметров, определяющих конечные характеристики продукта, и вытягивать именно их, остальные параметры должны уйти в ограничения
3. Найти оптимальный баланс между интересами различных групп специалистов - ключевая задача главного конструктора

Самый главный вывод:

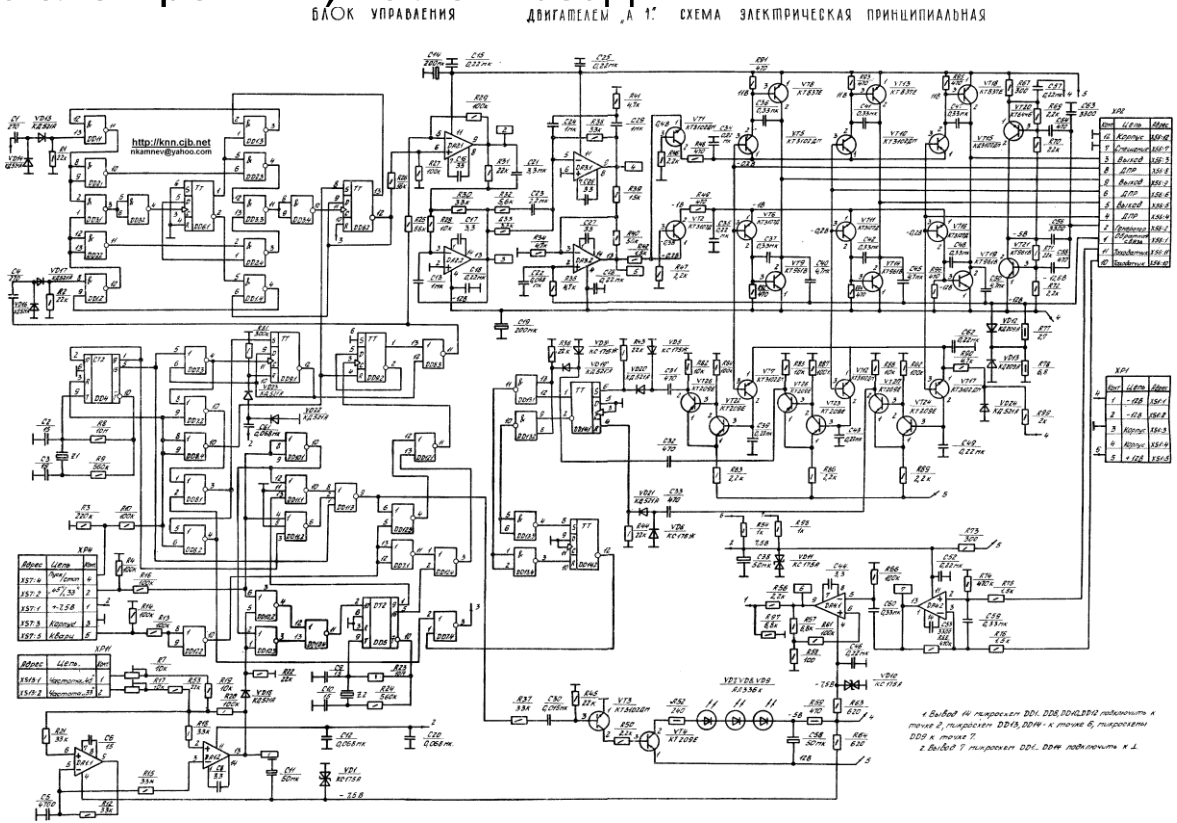
*«Никогда не позволяйте изобретателю управлять компанией:
Вы просто не сможете заставить его остановить разработки
и выпустить продукт на рынок»*

Роял Литтл (Textron)

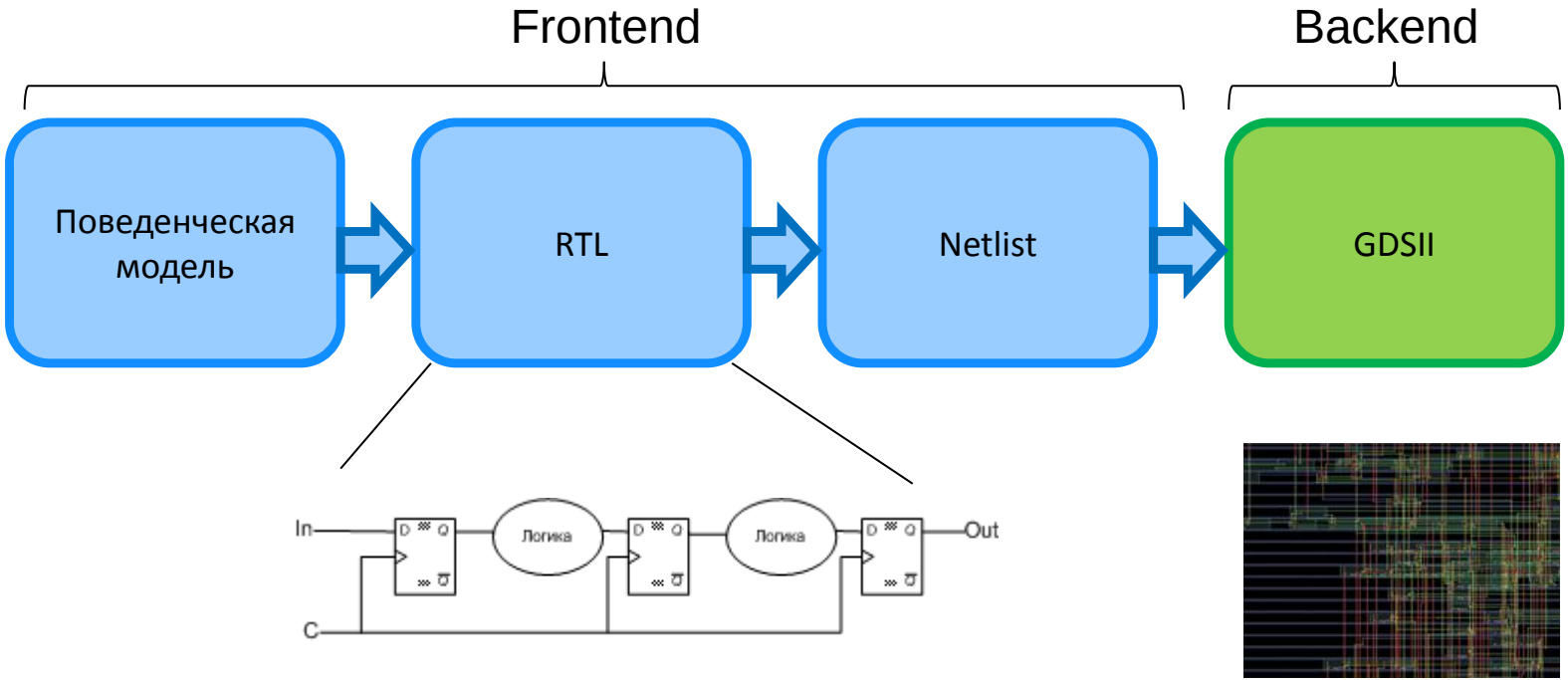
«Сделай настолько просто, насколько это возможно, но не проще!»
Альберт Эйнштейн

Разработка

Разработка электроники, 20 лет назад



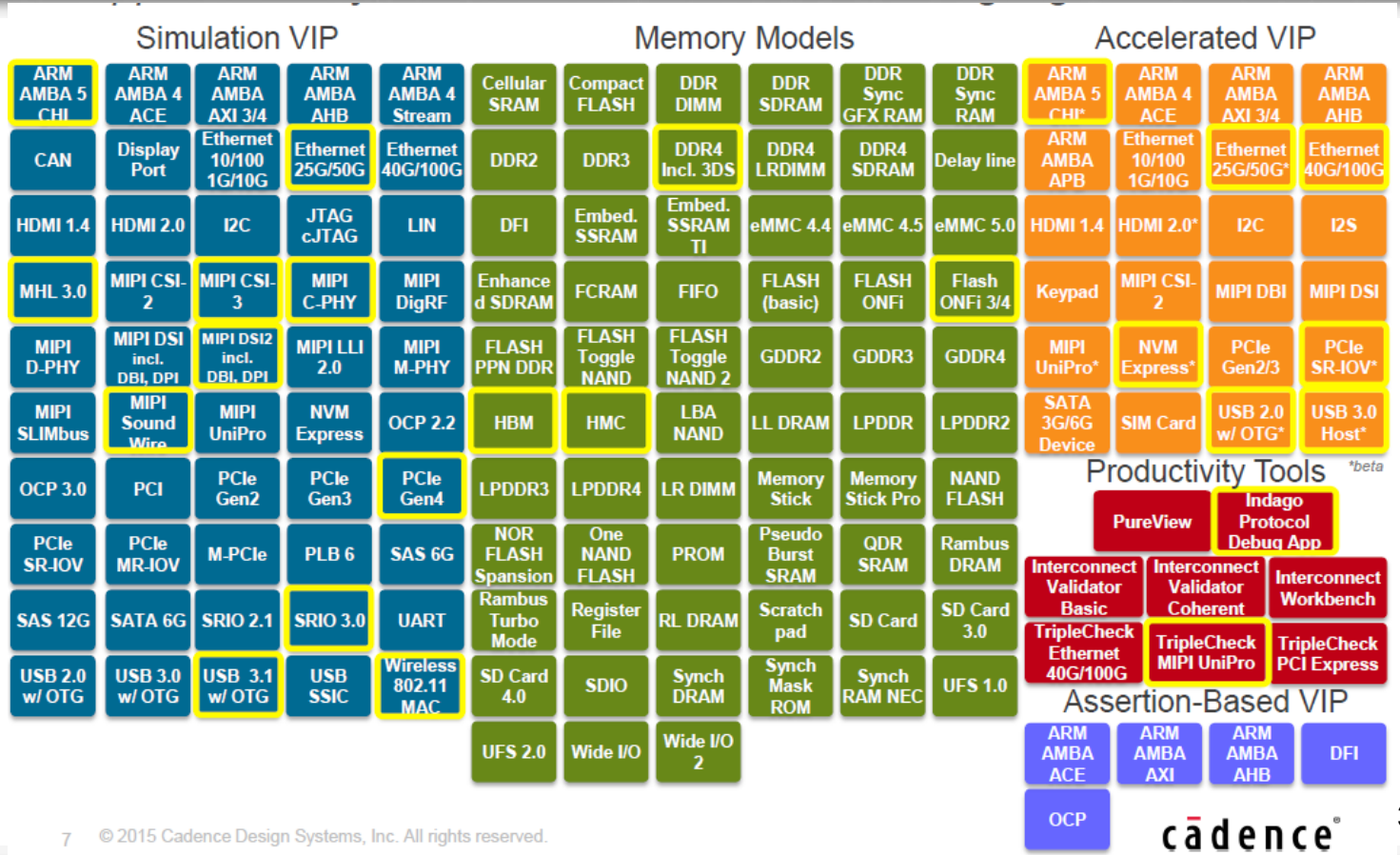
Разработка электроники сейчас (очень упрощенный flow)



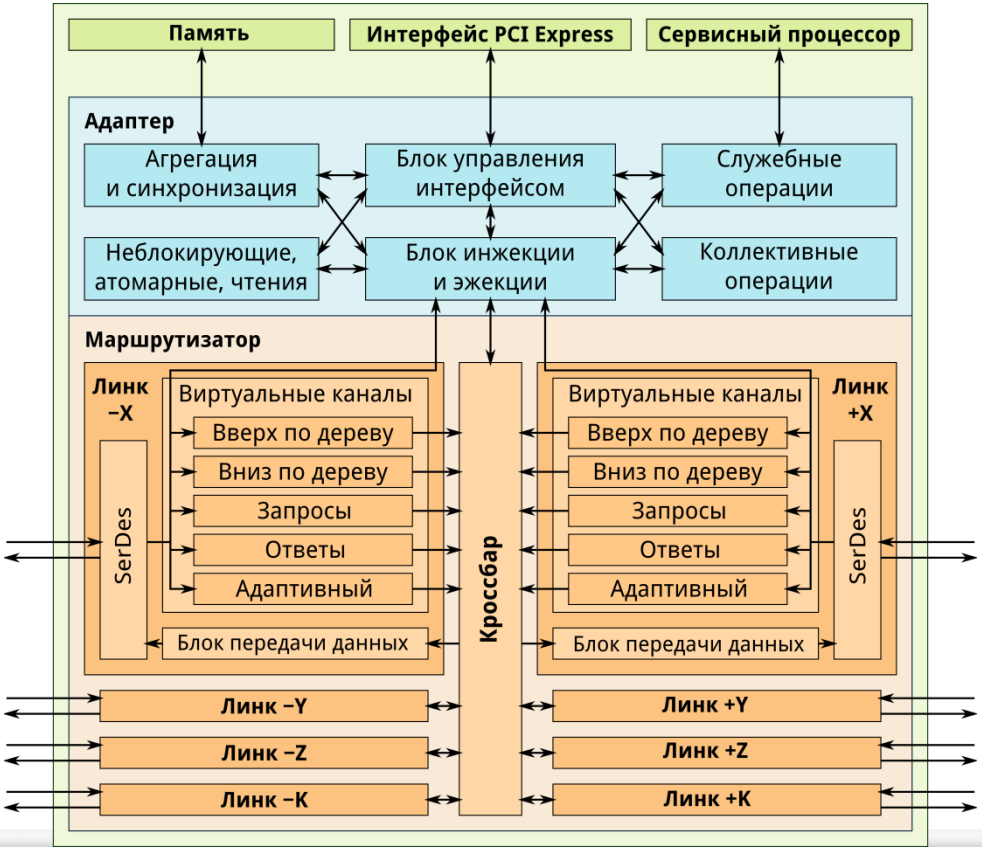
Исходный текст на уровне RTL:

```
always @ (posedge clk or negedge rst_n) begin
 if(!rst_n) begin
  rrd_rg <= {RGDBW{1'b0}};
 end
 else begin
  if (rrd_inc_en_rg && rrd_dec_en_rg)
   rrd_rg <= rrd_rg + rrd_inc_val_rg - rrd_dec_val_rg;
  else if (rrd_inc_en_rg)
   rrd_rg <= rrd_rg + rrd_inc_val_rg;
  else if (rrd_dec_en_rg)
   rrd_rg <= rrd_rg - rrd_dec_val_rg;
 end
end
```

Наборы IP:



Микроархитектура:



Выводы:

1. Современная аппаратура – это сотни миллионов и миллиарды транзисторов на кристалле
2. Сложность разработки очень высока
3. Там, где уместно – надо применять верифицированные IP-блоки
4. Современный разработчик СБИС – это образование инженера электронной техники, понимание физических процессов, навыки программиста, владение инструментами разработки, симуляции, синтеза, физдизайна, инженерного анализа
5. По возможности, надо повышать уровень абстракции описания и давать как можно больше свободы инструментарию

*«Мне трудно заставить себя думать об ошибках.
Не то, чтобы я их не делал, просто меня так воспитали:
смотреть только на светлую сторону жизни»*

Ричард Брэнсон

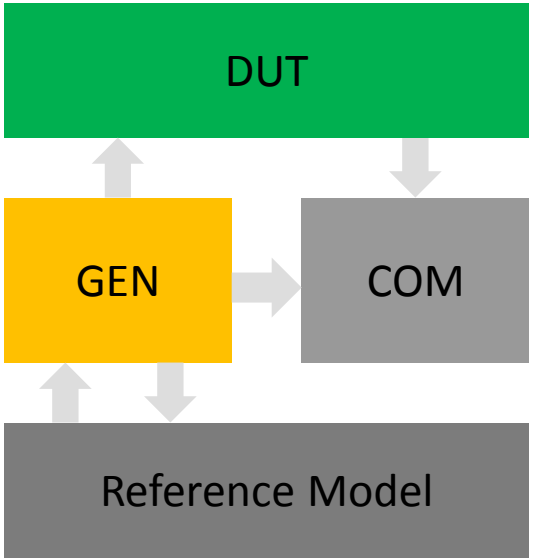
Верификация

1. Задачи верификации
2. Цена ошибки
3. Система верификации
4. Методы верификации (ASSERT, Coverage, сравнение)
5. Уровни верификации

1. Получение работоспособного кристалла с первой итерации
2. Контроль качества на каждом этапе разработки
3. Построение автоматической системы верификации

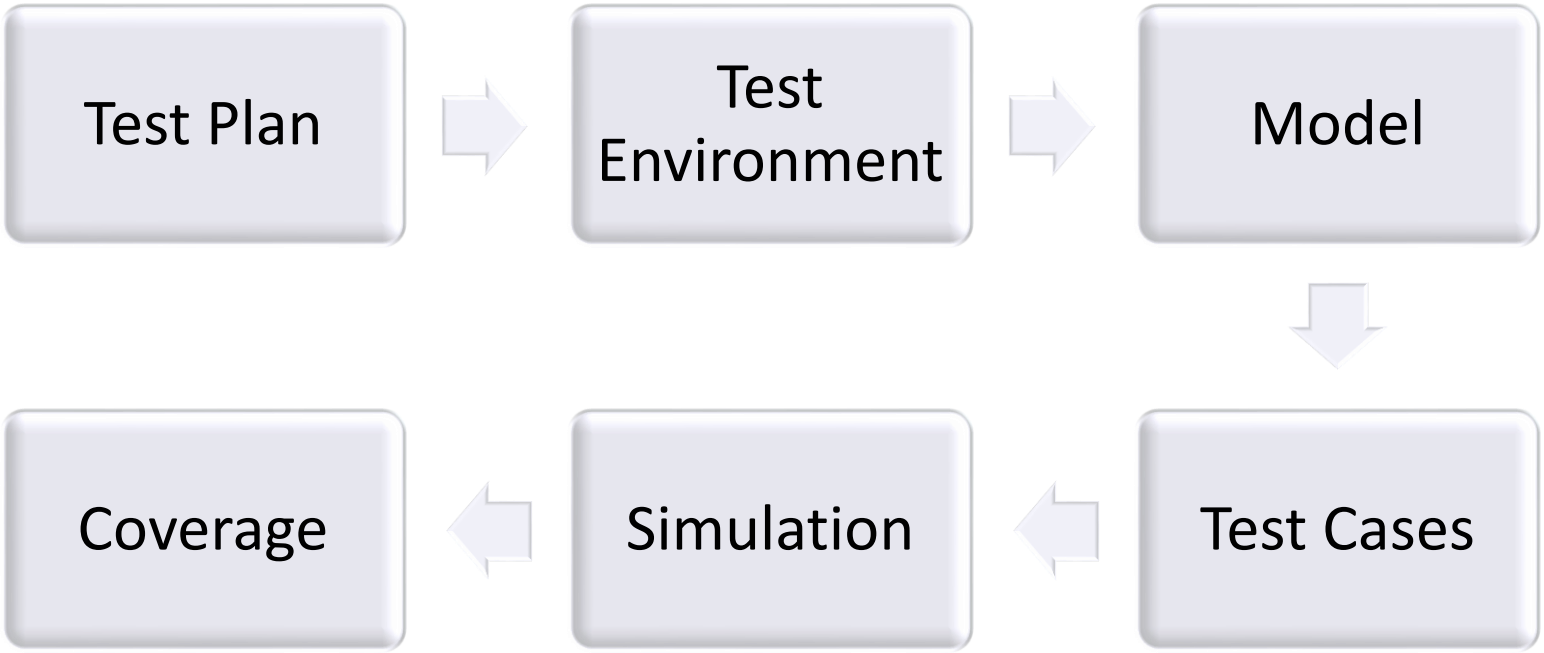
1. Тестовое окружение
2. Эталонная модель схемы
3. Среда моделирования
4. Система сбора и анализа результатов

DUT - Design Under Test - проверяемый блок на любом языке
GEN – генератор воздействий на блок
Reference Model – эталонная модель
COM – блок сравнения и анализа результата

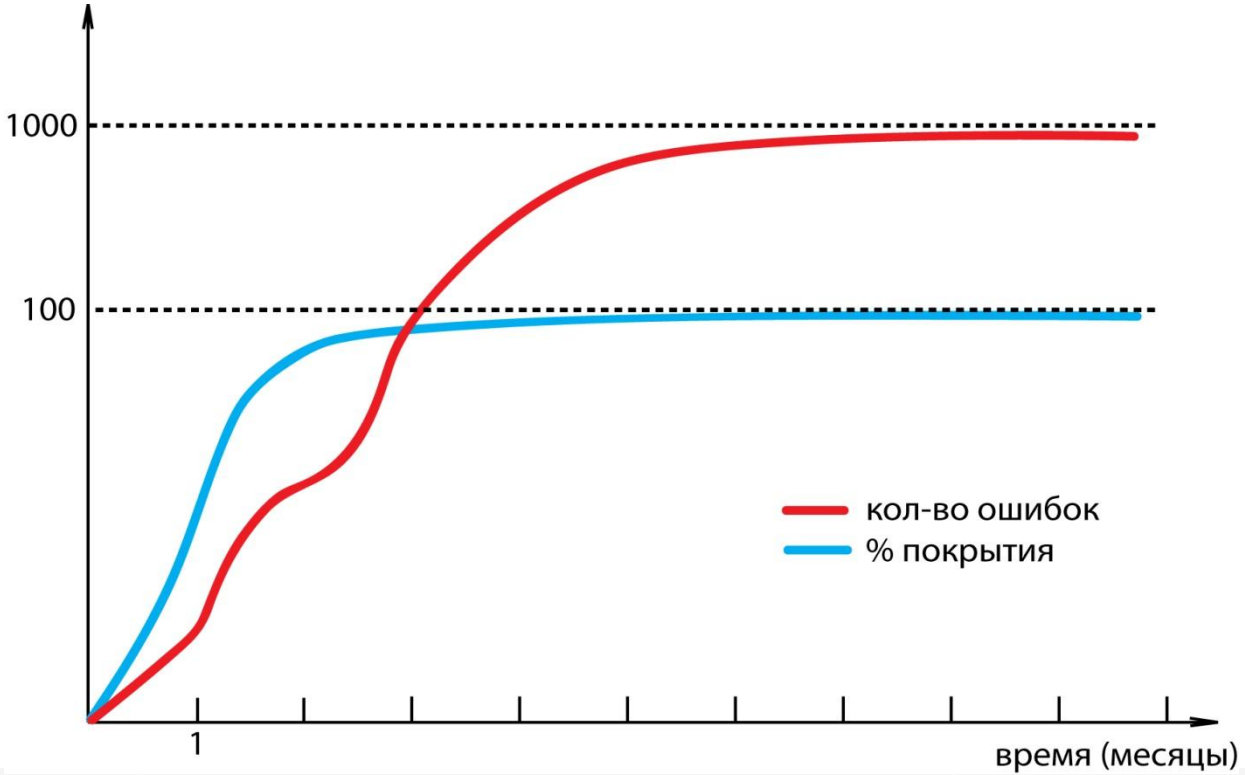


Верификация

Этапы верификации



Кривая кол-ва ошибок и покрытия от времени на реальном проекте



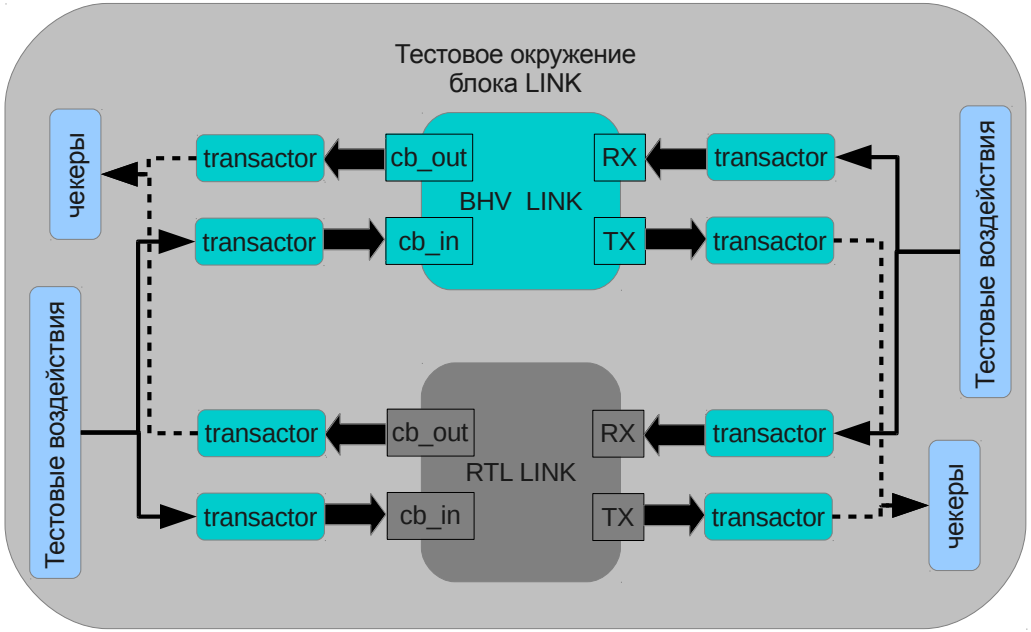
Coverage

1. Line – каждая ли строка кода была задействована
2. Branch – задействованы ли все ветки переходов
3. Condition – анализ условий переходов
4. Expression – все ли строчки таблицы истинности были задействованы
5. FSM – все ли состояния конечных автоматов были задействованы

Уровни верификации:

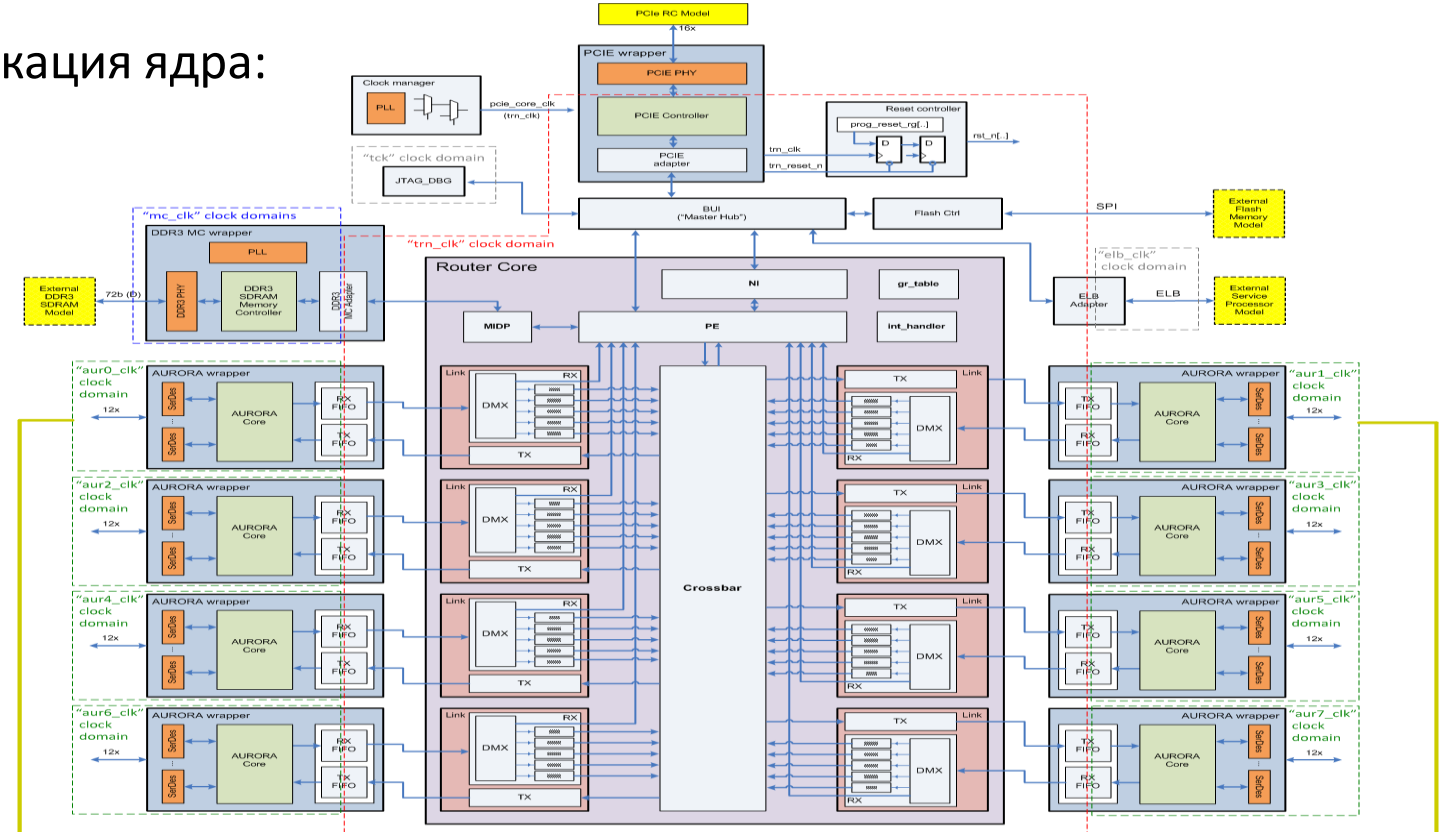
- Крупные блоки
- Ядро
- Маршрутизатор
- В составе вычислительной системы

Верификация на уровне крупных блоков

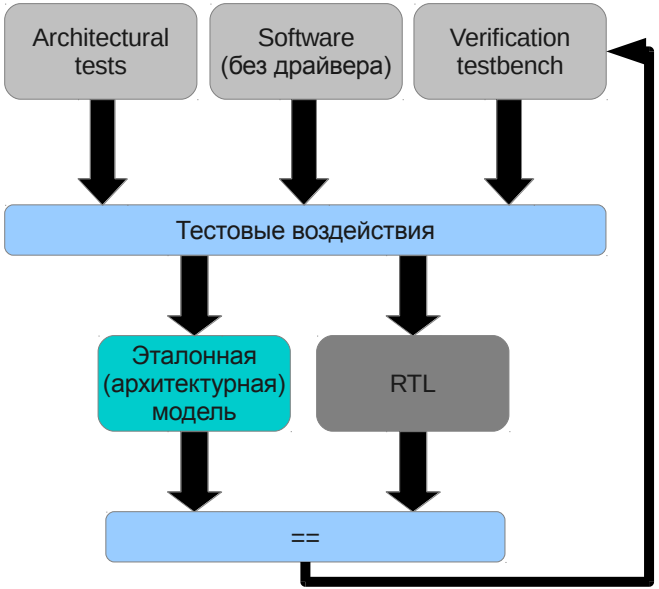


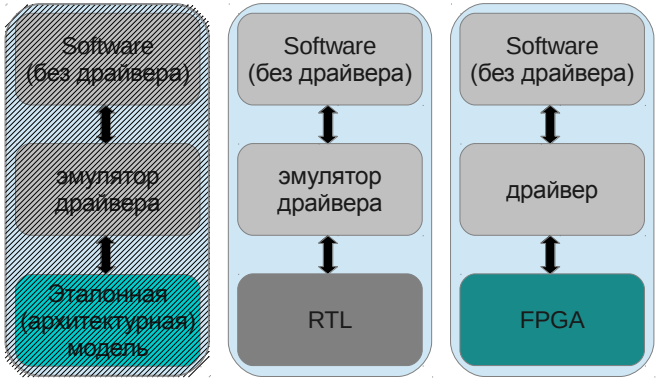
Верификация

Верификация ядра:

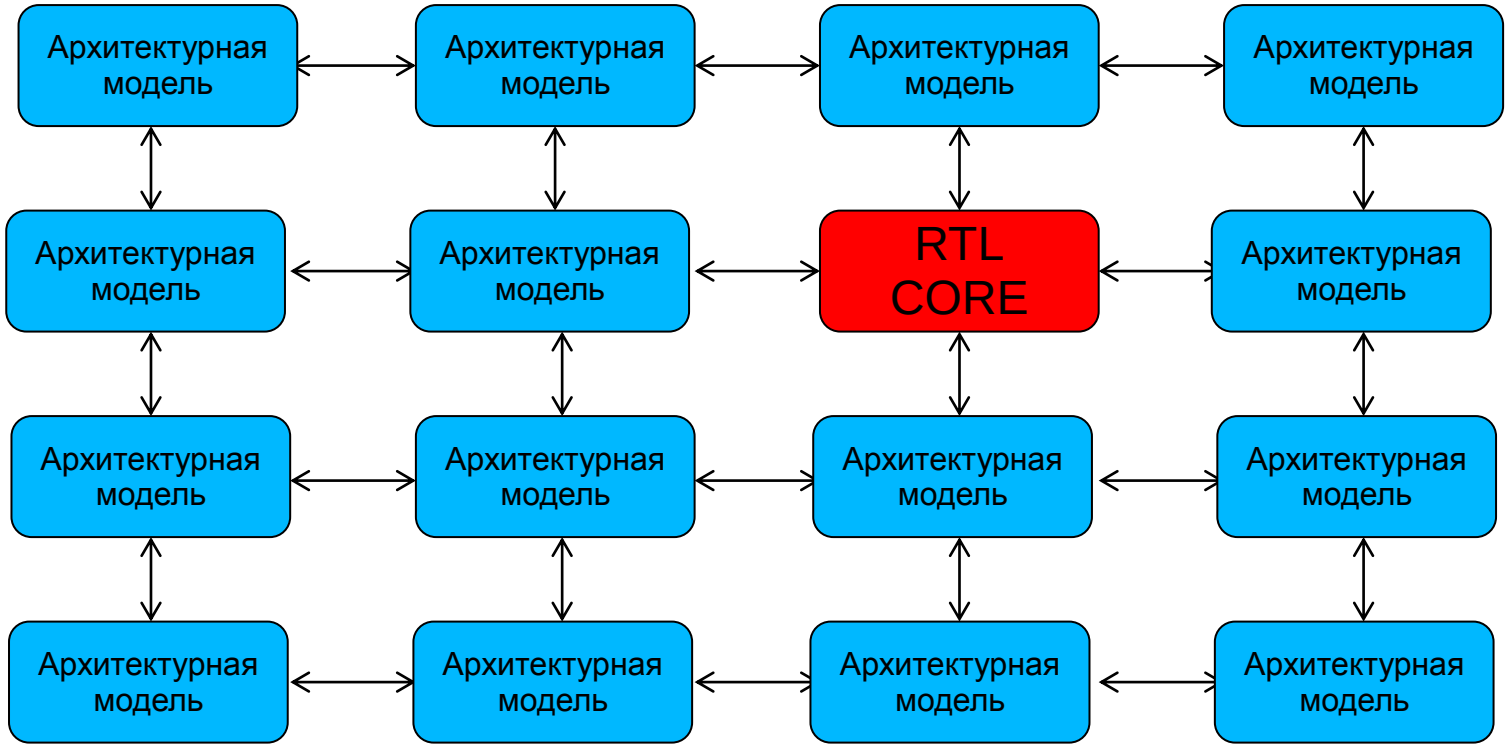


Верификация ядра:





В составе вычислительной системы:



Автоматический запуск тестов по изменениям RTL и оценка Coverage (Buildbot):

Coverage Summary Report, Module-Based

Top Level Summary

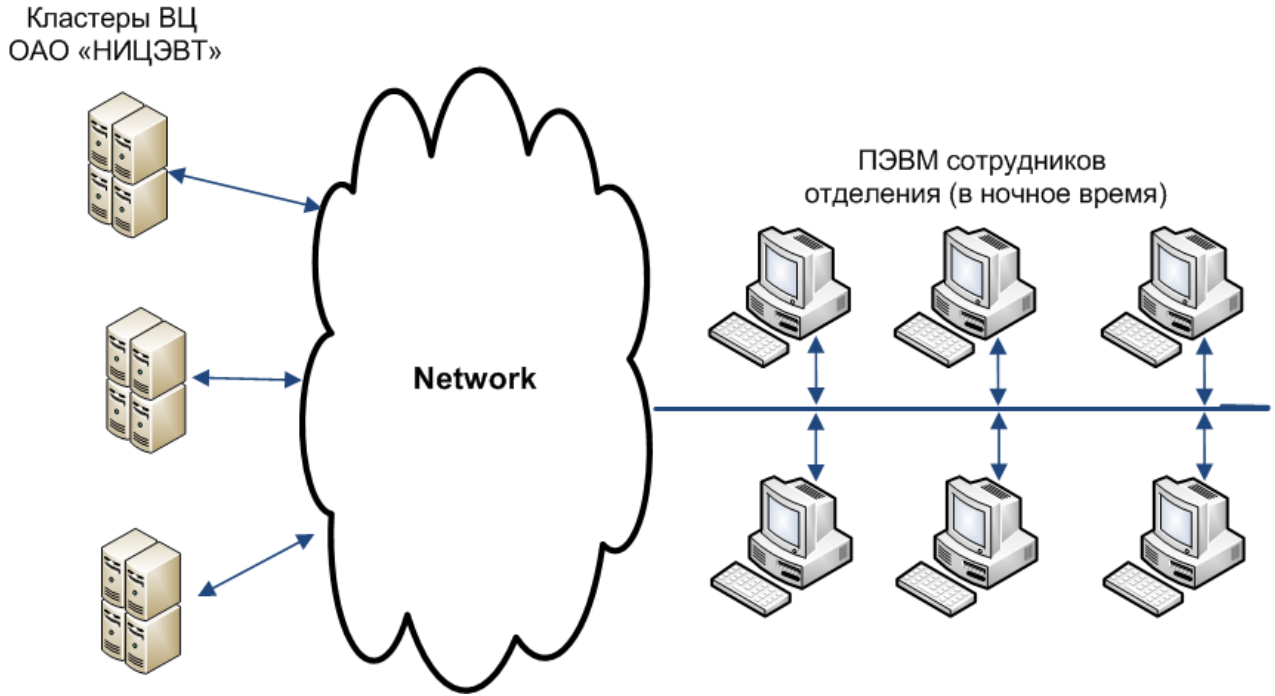
Overall Module-Based Coverage

Total	Block	Expression	Toggle	FSM
87%	96% (5945/6164)	89% (926/1045)	72% (102261/141413)	93% (43/46)

Coverage Summary Report, Module-Based

Total	Block	Expression	Toggle	FSM	Name
No Items	Not Scored	Not Scored	Not Scored	Not Scored	testbench
No Items	Not Scored	Not Scored	Not Scored	Not Scored	router_top_sc
No Items	Not Scored	Not Scored	Not Scored	Not Scored	router_pll
No Items	Not Scored	Not Scored	Not Scored	Not Scored	router_out
No Items	Not Scored	Not Scored	Not Scored	Not Scored	router_in
89%	No Items	No Items	89% (2592/2900)	No Items	router_top
57%	91% (10/11)	No Items	24% (18/74)	No Items	router_reset_ctrl
78%	No Items	No Items	78% (1250/1605)	No Items	bui
94%	96% (110/114)	No Items	92% (929/1009)	No Items	bui_pcie_rx
82%	87% (139/160)	No Items	78% (2211/2815)	82% (14/17)	bui_pcie_tx
99%	100% (7/7)	No Items	99% (405/406)	No Items	hs_reg_out
59%	40% (2/5)	No Items	78% (1310/1675)	No Items	bui_rti
62%	94% (49/52)	No Items	30% (109/361)	No Items	bui_rti_reg
95%	96% (25/26)	No Items	94% (706/748)	No Items	bui_rti_adec
96%	100% (27/27)	No Items	92% (174/188)	No Items	hs_chan2c_gen

Вычислительные ресурсы:



Выводы:

1. Система верификации – важнейший элемент разработки, во многом определяющий её сроки и стоимость
2. Цена ошибки может достигать десятков и сотен миллионов рублей
3. Как бы ни был высок соблазн, как бы не «давило» руководство, не стоит запускать в производство СБИС, если нет уверенности в отсутствии грубых (неустранимых) ошибок

Макетирование

Задача:

Отработка RTL Router Core и
стека программного обеспечения

Выводы:

1. Макетирование (FPGA -прототипирование) – важный этап разработки, позволяющий оценить реализуемость системы и работоспособность совместно с ПО
2. Не стоит экономить на ПЛИС, опыт показывает, что ёмкости всегда не хватает

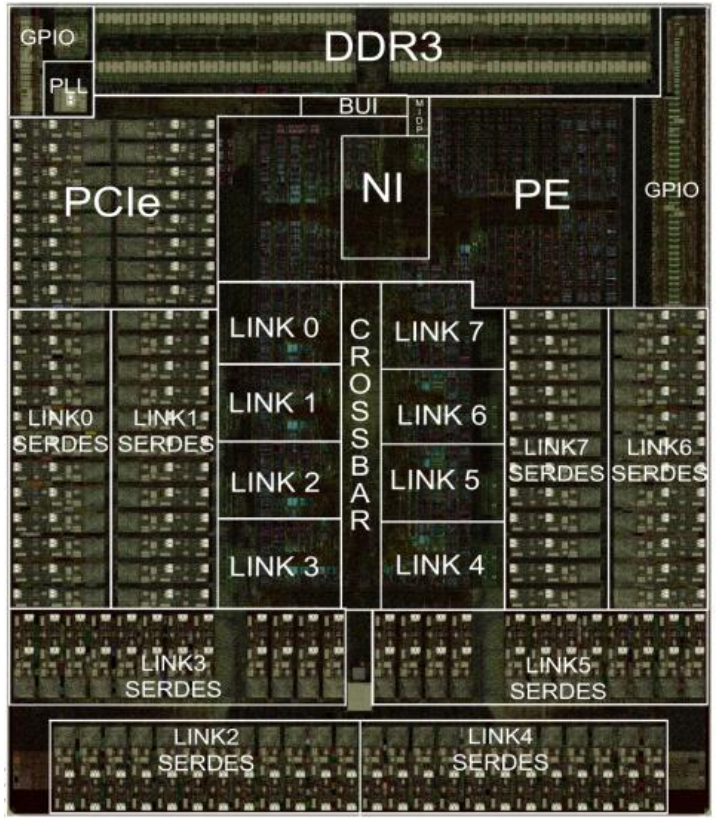
Проектирование СБИС

1. Принятые решения по порядку выполнения Backend и о снижении технических рисков

2. Об эффектах, требующих учета при проектировании топологии (SI, PI, TDP)

3. Немного о DFT (Design For Test)

4. Юридические аспекты



Характеристики EC8430:

Техпроцесс…….…. TSMC 65nm GP

Размер…….…….…13.0mm x 10.5mm

Кол-во транзисторов……..……180M

TDP……..……20W

Электропитание:

SerDes ……..…….1.0V±5%

Core…….…….…….1.0V±5%

I/O……..…2.5V±10%

Корпус…..……..….FCBGA-1521 40mm x 40mm

Выводы:

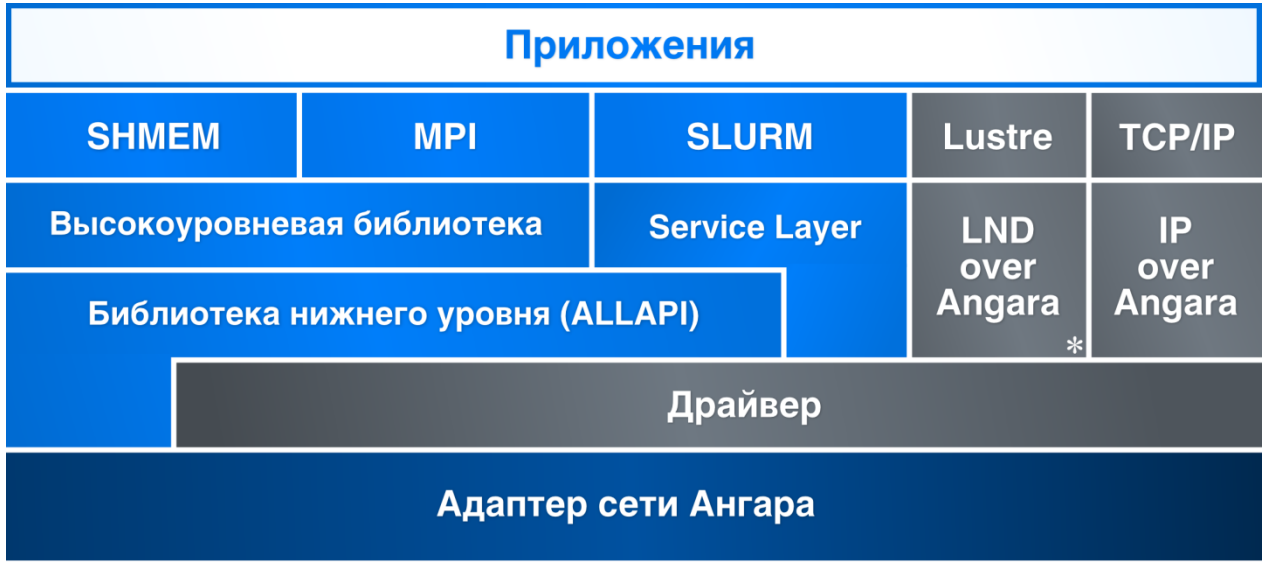
1. Создание заказной СБИС позволяет получить существенный выигрыш по характеристикам и цене
2. Разработка СБИС – очень дорогой процесс
3. Снижение технических рисков приводит к снижению числа перезапусков изготовления СБИС, как следствие позволяет снизить стоимости продукции и сократить сроки разработки
4. Наше достижение - за счет системы верификации получили работоспособную СБИС с первого запуска!

Разработка сетевого оборудования и системного программного обеспечения

1. Программное обеспечение – «лицо» системы!
2. СБИС получена, и….. долгий путь к достижению работоспособности
3. О вариативности сетевого оборудования
4. Что есть сейчас (адаптер, платформа)

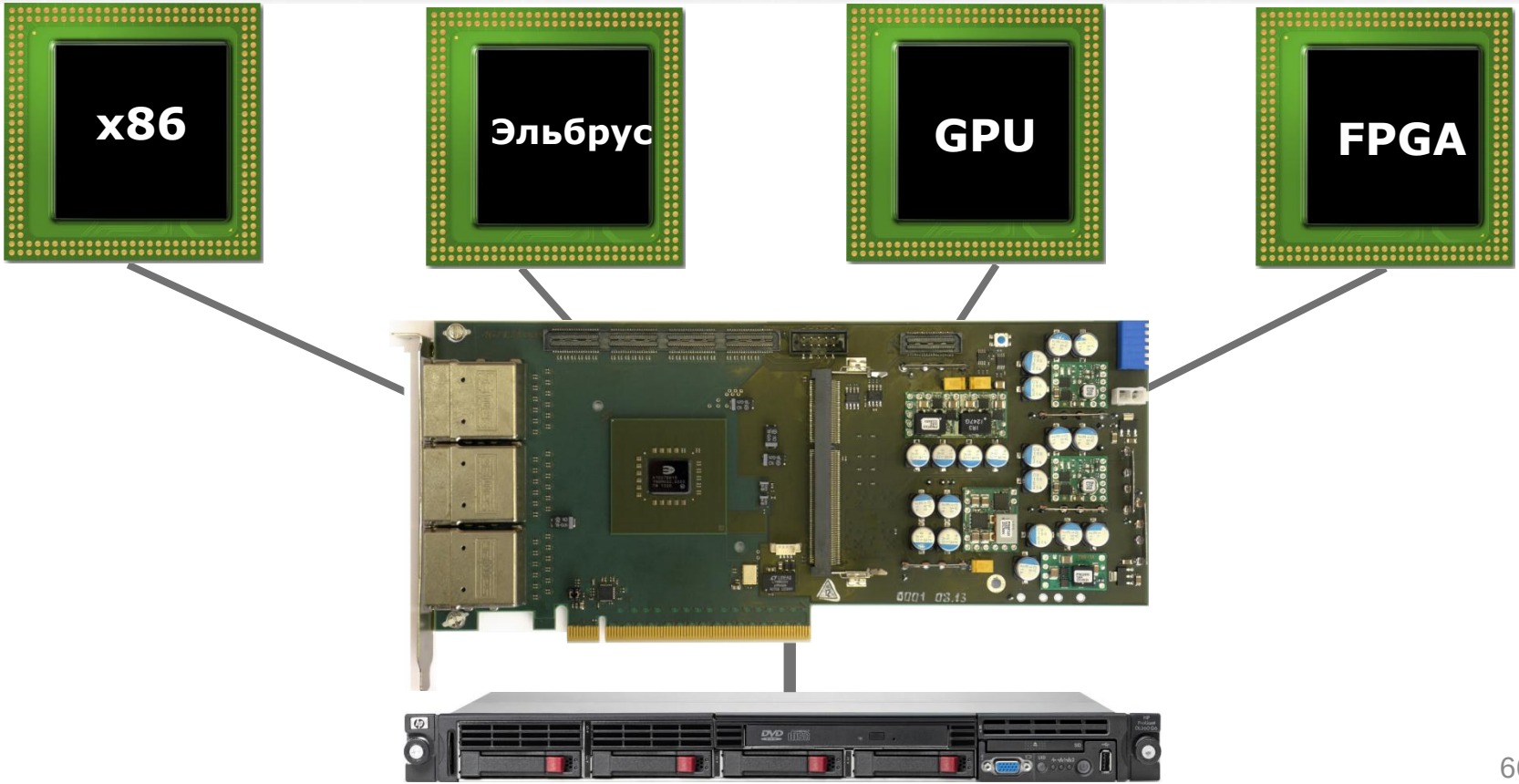
Стек программного обеспечения

Готовы открыть API

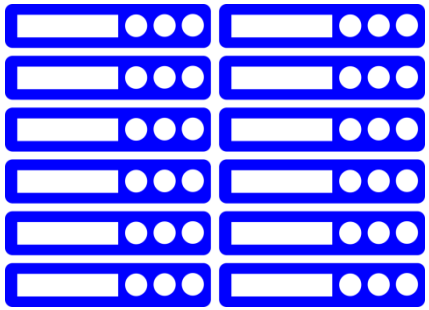


- Поддержка ОС : Astra Linux SE 1.3, ОС «Эльбрус» ,OpenSUSE/SLES 11SP3, CentOS 6.0-6.3, Версия ядра Linux от 2.6.21 до 3.16.0
- Поддержка компиляторов языков Fortran 77/90/95 (GNU, Intel), C/C++ (GNU, Intel)

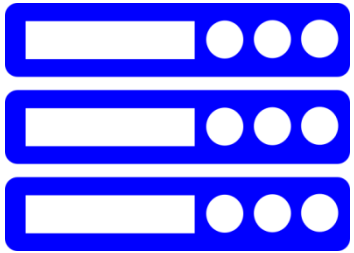
Сетевое оборудование



Приложения



HPC



ЦОДы

Big Data

АНГАРА

ВЫСОКОСКОРОСТНАЯ СЕТЬ

Не пренебрегайте пожеланиями потребителей, в идеале, именно потребители должны «играть первую скрипку» в разработке спецификации

Оценочное тестирование

Тестовый вычислительный кластер «Ангара-К1» 36 узлов

24 вычислительных узла

- Supermicro SuperServer 5017GR-TF
- 2 процессора Intel Xeon E5-2630 (LGA2011, 6 ядер, 2.3 ГГц)
- 64 ГБ

12 вычислительных узлов

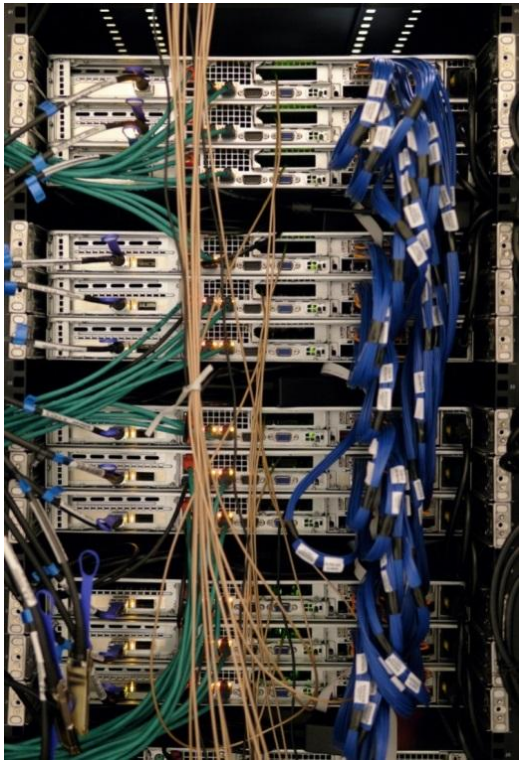
- Supermicro SuperServer 5017GR-TF
- процессор Intel Xeon E5-2660 (LGA2011, 8 ядер, 2.2 ГГц)
- 64 ГБ

Сеть «Ангара»

- Адаптер EC8430, топология 3D-тор 3x3x4
- Собственная реализация OpenSHMEM
- MPI: MPICH 3.0.4

Операционная система

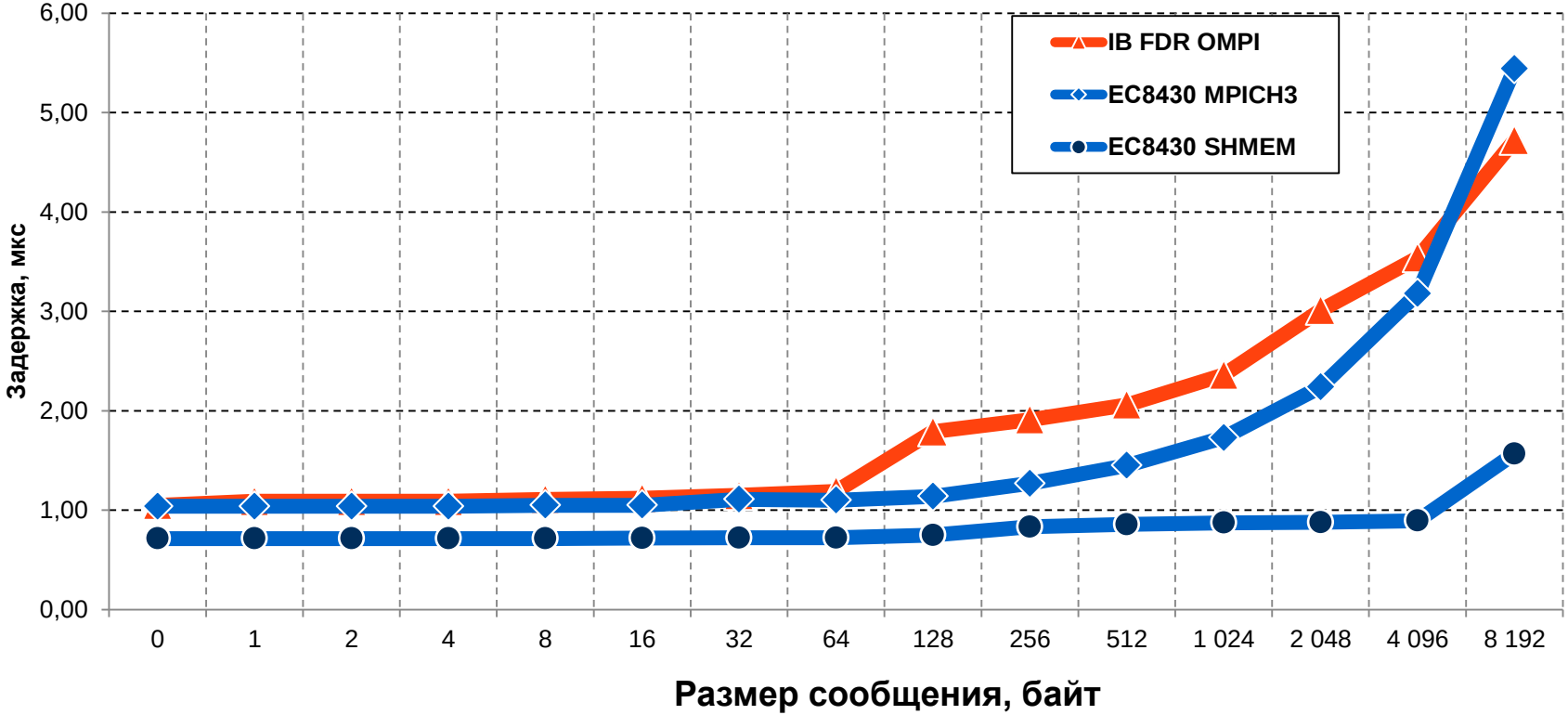
- SLES 11 SP2, Linux 3.0.13-0.27-default
- GCC 4.3.4 (revision 152973)



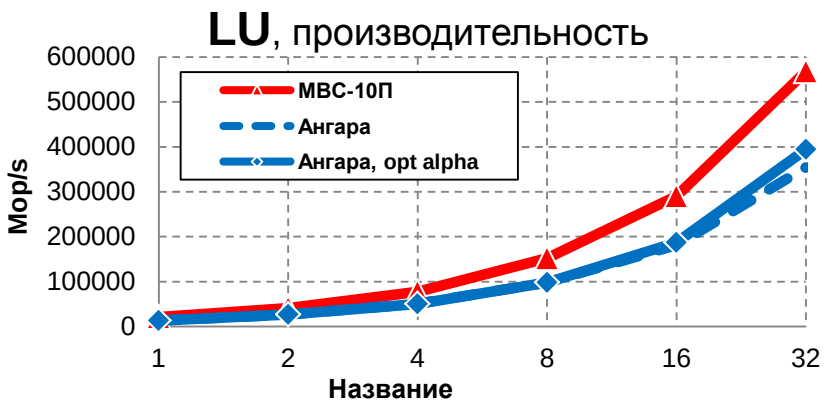
«Ангара-К1» и МВС-10П (МСЦ РАН)

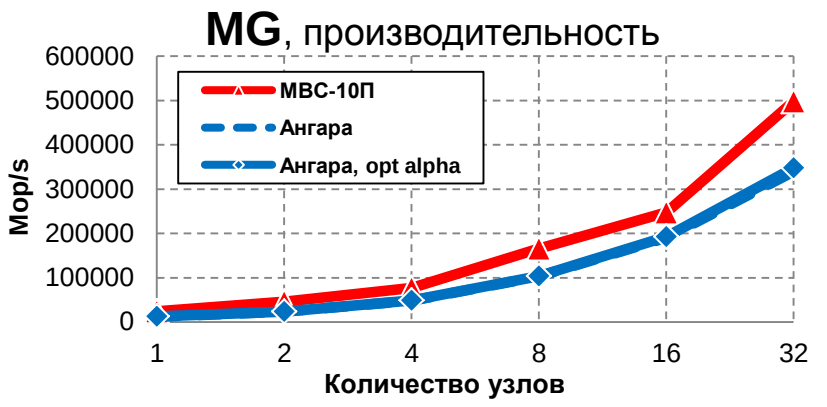
	Ангара-К1		МВС-10П
Узлы	A	2x Xeon E5-2630 по 6 ядер, 2.3 ГГц	2x Xeon E5-2690 по 8 ядер, 2.9 ГГц
	B	Xeon E5-2660 по 8 ядер, 2.2 ГГц	
Количество узлов		24*A+12*B = 36	207 (36)
Память узла		64 ГБ	64 ГБ
Сеть		Ангара 3D-тор 3x3x4	Infiniband 4xFDR Fat Tree

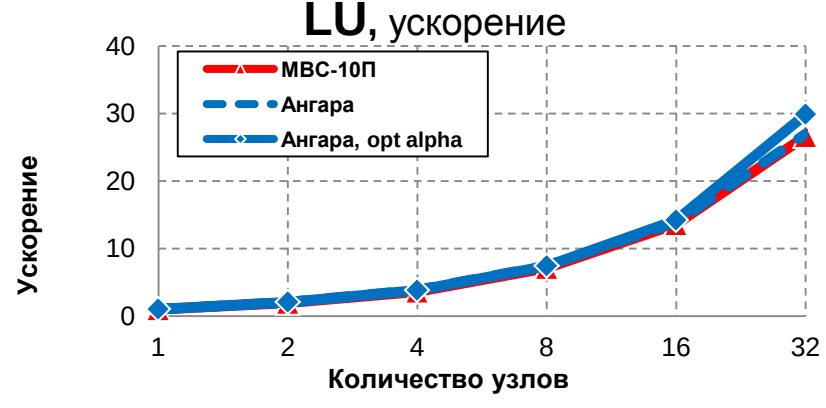
«Ангара» vs IB FDR
2 узла – задержка на MPI (osu_latency)

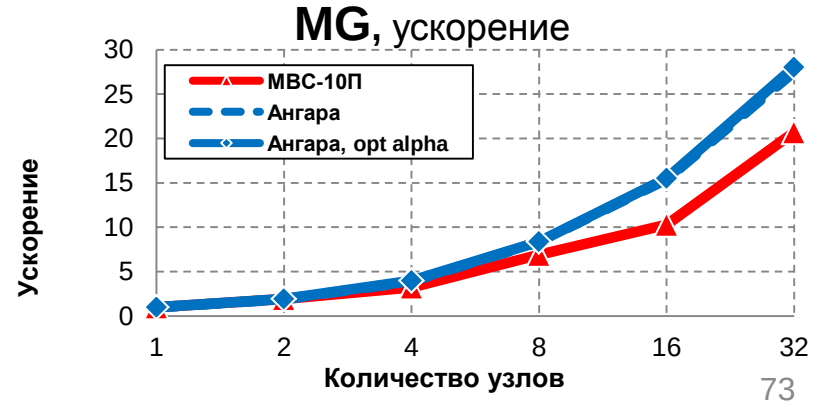


Результаты тестов NPB LU, MG (класс C)

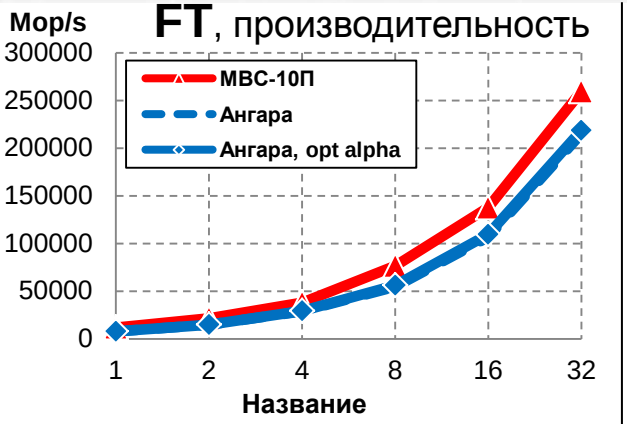


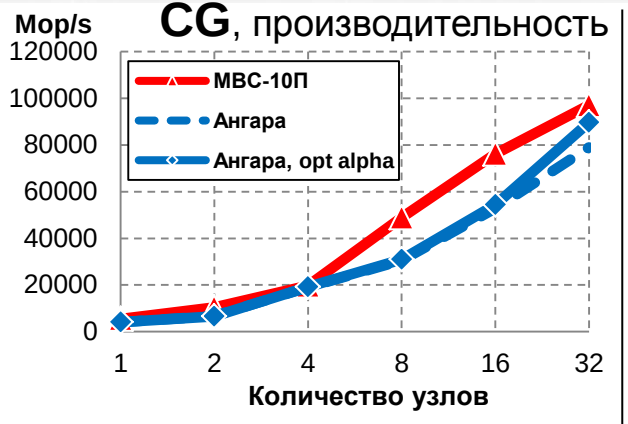


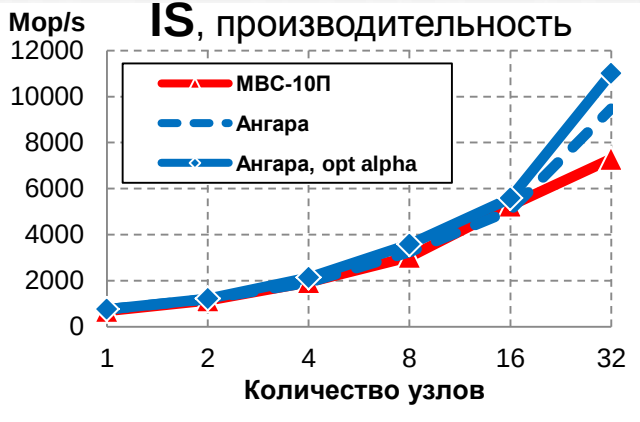


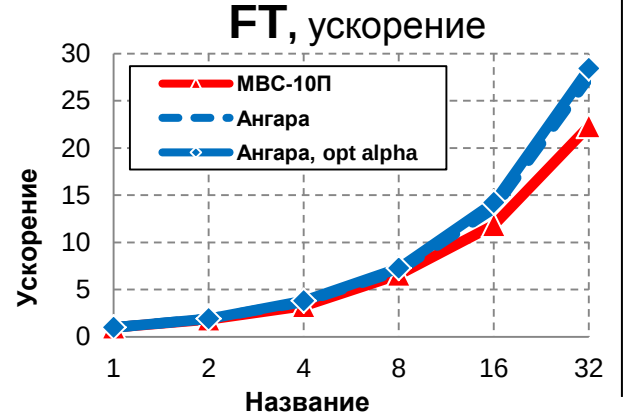


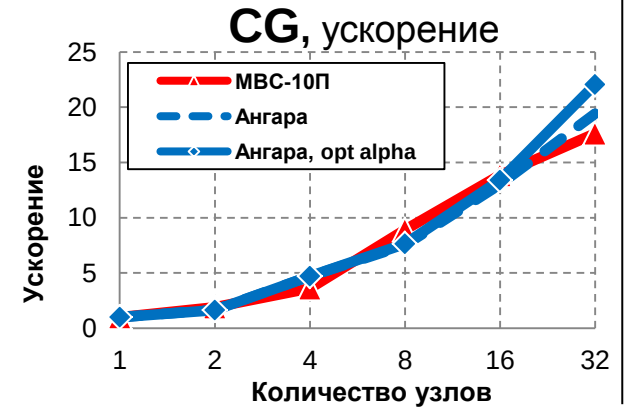
Результаты тестов NPB FT, CG, IS (класс C)

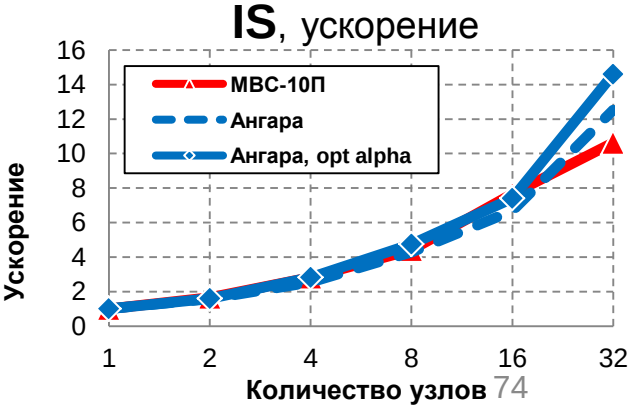








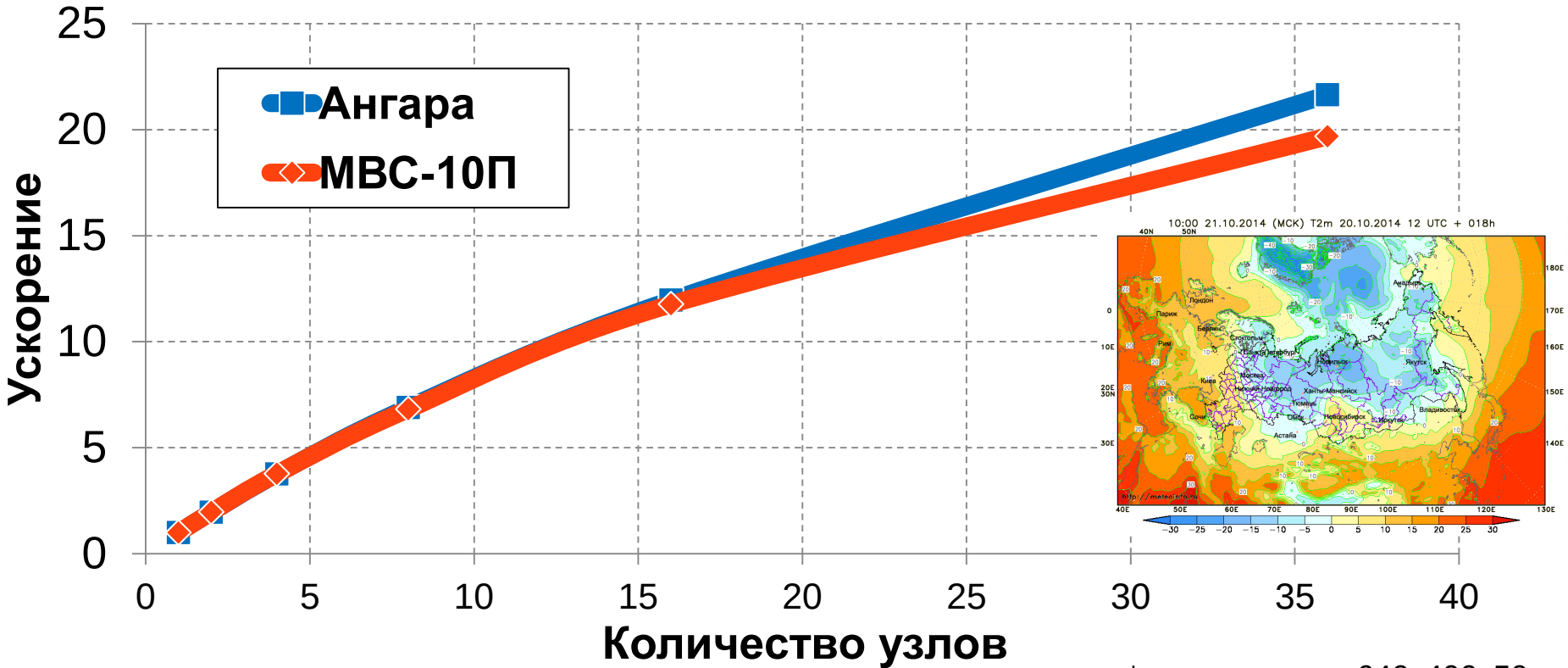




Ангара vs IB FDR Масштабирование модели прогноза погоды ПЛАВ*, ИВМ РАН, Гидрометцентр

НИЦ ВТ

Ростех Объединенная приборостроительная корпорация

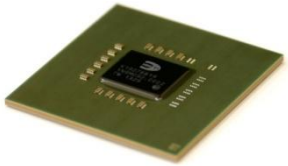


* - разрешение 640x400x50

Статистика использования кластера «Ангара-К1»

- В режиме внешнего доступа работает с 1.11.2015 г.
- Отключался на 12 дней – на новогодние праздники для установки дополнительного оборудования
- За 5 месяцев запуски тестовых приложений осуществили 23 пользователя

Сетевое оборудование «Ангара» и сервисы



Чип ЕС8430

FCBGA 1521
40 40 мм
35 Вт

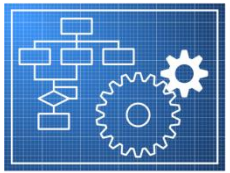
Development kit

PCIe
Full-height
full length

Адаптер

PCIe
Full-height
full length

Заказная разработка платы адаптера



Доработка ПО адаптера и адаптация библиотек

Адаптация и профилирование прикладного ПО

«Бог дал мне чрезвычайно короткую память, что позволяет мне иметь дело не с прошлым, а с будущим»

Джордж Сорос

Перспективы

Коммуникационная сеть «Ангара» Gen 2

Развитие архитектуры и функциональных возможностей:

- Поддержка различных топологий (butterfly, dragonfly)
- Улучшение аппаратной поддержки MPI и парадигмы PGAS
- Улучшение аппаратной поддержки коллективных и синхронизационных операций

Интеграция и партнёрство:

- Интеграция с процессором
- Интеграция с ускорителем

Усовершенствование технологий:

- Оптические трансиверы
- Новые IP

GRAPH HPC 2017

Научно-практическая конференция

Технологии параллельной обработки больших графов

2 марта 2017, Москва

- Конференция GraphHPC: 2014, 2015, 2016 годы
- Конкурс с автоматической системой проведения
- Доступ на кластер с сетью Ангара
- Сайт конференции graphhpc.dislab.org
- Открытое ПО

NVIDIA

intel Software

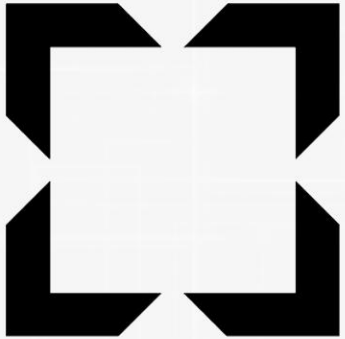
IBM

SAPPHIRE Professional Graphics Solutions

ПЛАТФОРМЫ

Благодарю за внимание!
Вопросы?

simonov@nicevt.ru



Ростех

Объединенная
приборостроительная
корпорация