Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра ВС

РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ

по дисциплине «Архитектура Вычислительных Систем»

Вариант 18

Выполнил: студент гр. ИП-712

Алексеев С.В.

Проверил: Ефимов А.В.

Новосибирск 2019 г.

Оглавление

[1. ЗАДАНИЕ 2](#_Toc28180963)

[2. АНАЛИЗ ВОЗМОЖНОСТЕЙ ПРОЦЕССОРОВ С МИКРОАРХИТЕКТУРОЙ ЭЛЬБРУС 2](#_Toc28180964)

[3. ПРИМЕР ФУНКЦИОНАЛЬНОЙ СТРУКТУРЫ СОВРЕМЕННОГО ПРОЦЕССОРА «Эльбрус-8C» 6](#_Toc28180965)

[4. РАСЧЕТ ФУНКЦИЙ НАДЕЖНОСТИ И ГОТОВНОСТИ 12](#_Toc28180966)

[5. СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 17](#_Toc28180967)

# 1. ЗАДАНИЕ

1. Произвести анализ возможностей процессоров с микроархитектурой Эльбрус. Привести пример функциональной структуры современного процессора.

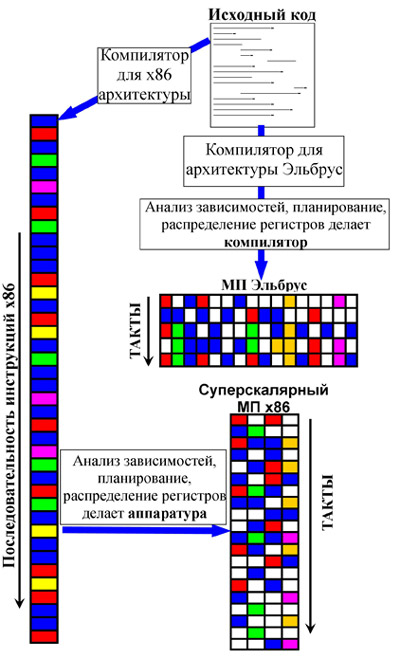
2. Выполнить численный расчет и построить графики для функции *r*(*t*) надежности и функции S(i, t) готовности ЭВМ для следующих количественных характеристик:

– интенсивности отказов 10-2 1/*ч* ,

– интенсивности восстановления 1 1/*ч* .

# 2. АНАЛИЗ ВОЗМОЖНОСТЕЙ ПРОЦЕССОРОВ С МИКРОАРХИТЕКТУРОЙ ЭЛЬБРУС

Процессоры данной архитектуры сравниваются по производительности с современными процессорами фирмы Интел в подсчете ГигаФлопс, при том, что имеют более скромные характеристики по частоте. Например при тактовой частоте всего в 1,3 ГГц Эльбрус 8С имеет производительность 250 гигафлопс (то есть 250 млрд операций в секунду) на операциях с одинарной точностью, что сравнимо с последними поколениями современного семейства процессоров Intel с архитектурой X86-64: 6-ядерный Intel Core i7-4930K (поколение Ivy Bridge) при тактовой частоте 3,7-4,2 ГГц имеет теоретический пик производительности в районе 177 гигафлопс.. Это достигается благодаря более продуманному распределению инструкций для процессора Эльбрус.



В традиционных архитектурах типа RISC или CISC (х86, PowerPC, SPARC, MIPS, ARM), на вход процессора поступает поток инструкций, которые рассчитаны на последовательное исполнение. Процессор может детектировать независимые операции и запускать их параллельно (суперскалярность) и даже менять их порядок (внеочередное исполнение). Однако динамический анализ зависимостей и поддержка внеочередного исполнения имеет свои ограничения: лучшие современные процессоры способны анализировать и запускать до 4-х команд за такт. Кроме того, соответствующие блоки внутри процессора потребляют заметное количество энергии.

В архитектуре «Эльбрус» основную работу по анализу зависимостей и оптимизации порядка операций берет на себя компилятор. Процессору на вход поступают т.н. «широкие команды», в каждой из которых закодированы инструкции для всех исполнительных устройств процессора, которые должны быть запущены на данном такте. От процессора не требуется анализировать зависимости между операндами или переставлять операции между широкими командами: все это делает компилятор, исходя из анализа исходного кода и планирования ресурсов процессора. В результате аппаратура процессора может быть проще и экономичнее.

Компилятор способен анализировать исходный код гораздо тщательнее, чем аппаратура RISC/CISC процессора, и находить больше независимых операций. Поэтому в архитектуре Эльбрус больше параллельно работающих исполнительных устройств, чем в традиционных архитектурах, и на многих алгоритмах она демонстрирует непревзойденную архитектурную скорость.

**Возможности архитектуры Эльбрус:**

* 6 каналов арифметико-логических устройств (АЛУ), работающих параллельно.
* Регистровый файл из 256 84-разрядных регистров.
* Аппаратная поддержка циклов, в том числе с конвейеризацией. Повышает эффективность использования ресурсов процессора.
* Программируемое асинхронное устройство предварительной подкачки данных с отдельными каналами считывания. Позволяет скрыть задержки от доступа к памяти и полнее использовать АЛУ.
* Поддержка спекулятивных вычислений и однобитовых предикатов. Позволяет уменьшить число переходов и параллельно исполнять несколько ветвей программы.
* Широкая команда, способная при максимальном заполнении задать в одном такте до 23 операций (более 33 операций при упаковке операндов в векторные команды).

# 3. ПРИМЕР ФУНКЦИОНАЛЬНОЙ СТРУКТУРЫ СОВРЕМЕННОГО ПРОЦЕССОРА «Эльбрус-8C»

Микропроцессор «Эльбрус-8C» спроектирован и изготовлен по технологическим нормам 28 нм. Его кристалл имеет площадь 321 мм2 и содержит 2,73 млрд транзисторов. За основу взят процессор Эльбрус-4С. Частота 1.3Ггц. 250GFLOPS на опреациях с одинарной точностью при частоте 1.3Ггц. Эльбрус-8С - это пятое поколение процессоров Эльбрус.

Топологический план приведен на рисунке 1.

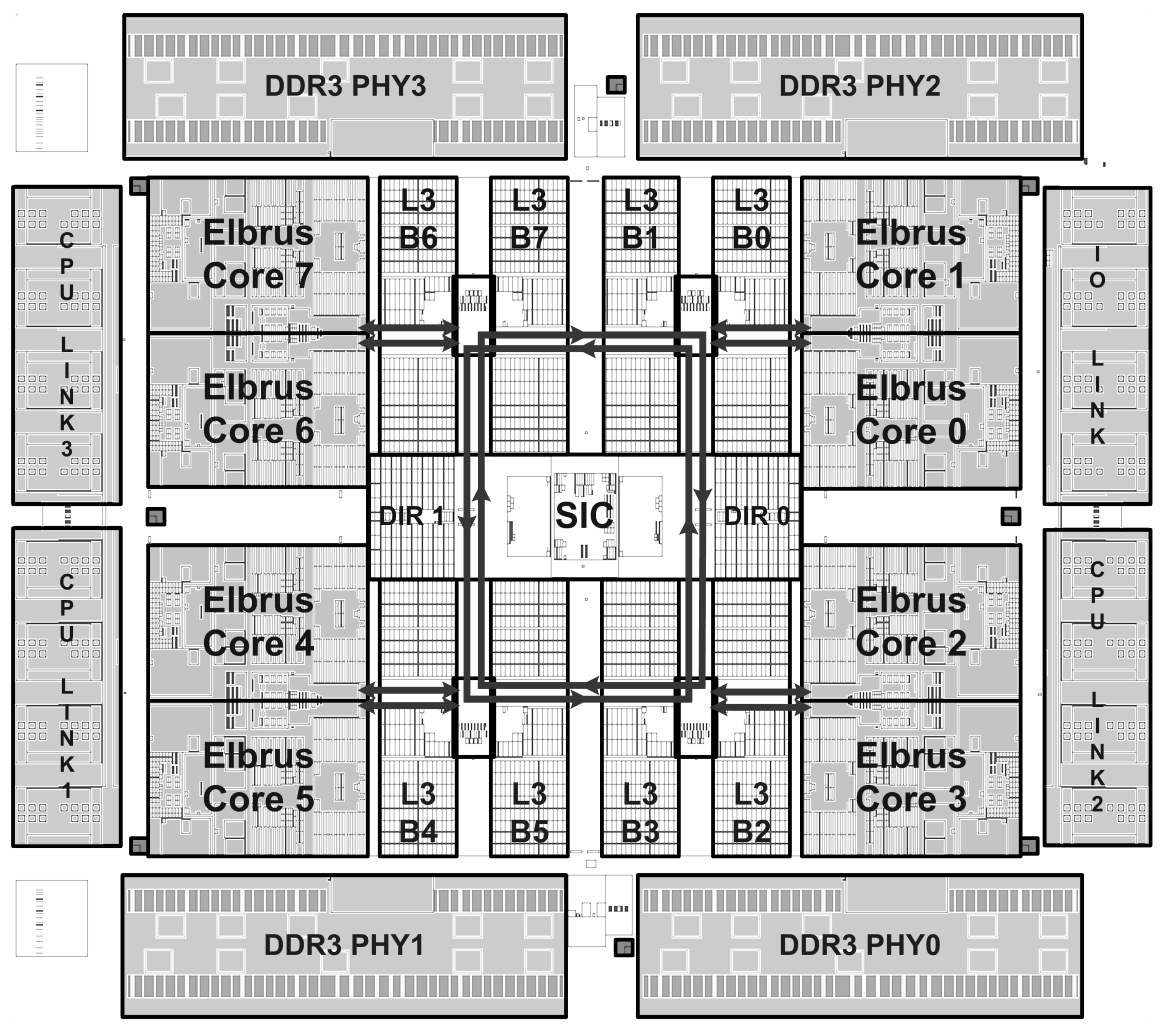


Рисунок 1. Топологический план микропроцессора «Эльбрус-8C»:

Elbrus Core 0-7 – процессорные ядра; L3 B0-7 – банки кэш-памяти третьего уровня; SIC – контроллер системных обменов; DIR 0,1 – глобальный справочник, DDR3 PHY0-3 – физуровни памяти; CPU LINK1,2,3 – физуровни каналов межпроцессорного обмена; IO LINK – физуровень канала ввода-вывода

На одном кристалле размещены восемь процессорных ядер архитектуры «Эльбрус» (Elbrus Core 0-7), каждое из которых способно исполнять до 25 операций за такт в скалярном режиме (32- и 64-разрядные данные) и до 41 операции за такт в векторном режиме (упакованные 32-разрядные данные). Ядра подключены к общей кэш-памяти третьего уровня (L3 кэш, 16 МБ. Ядра и банки общей кэш-памяти объединены в узлы Quart (кварты) по два ядра и два банка в каждом. Кварты соединены двунаправленным буферизующим кольцом. На рисунке 2 показана структурная схема микропроцессора.

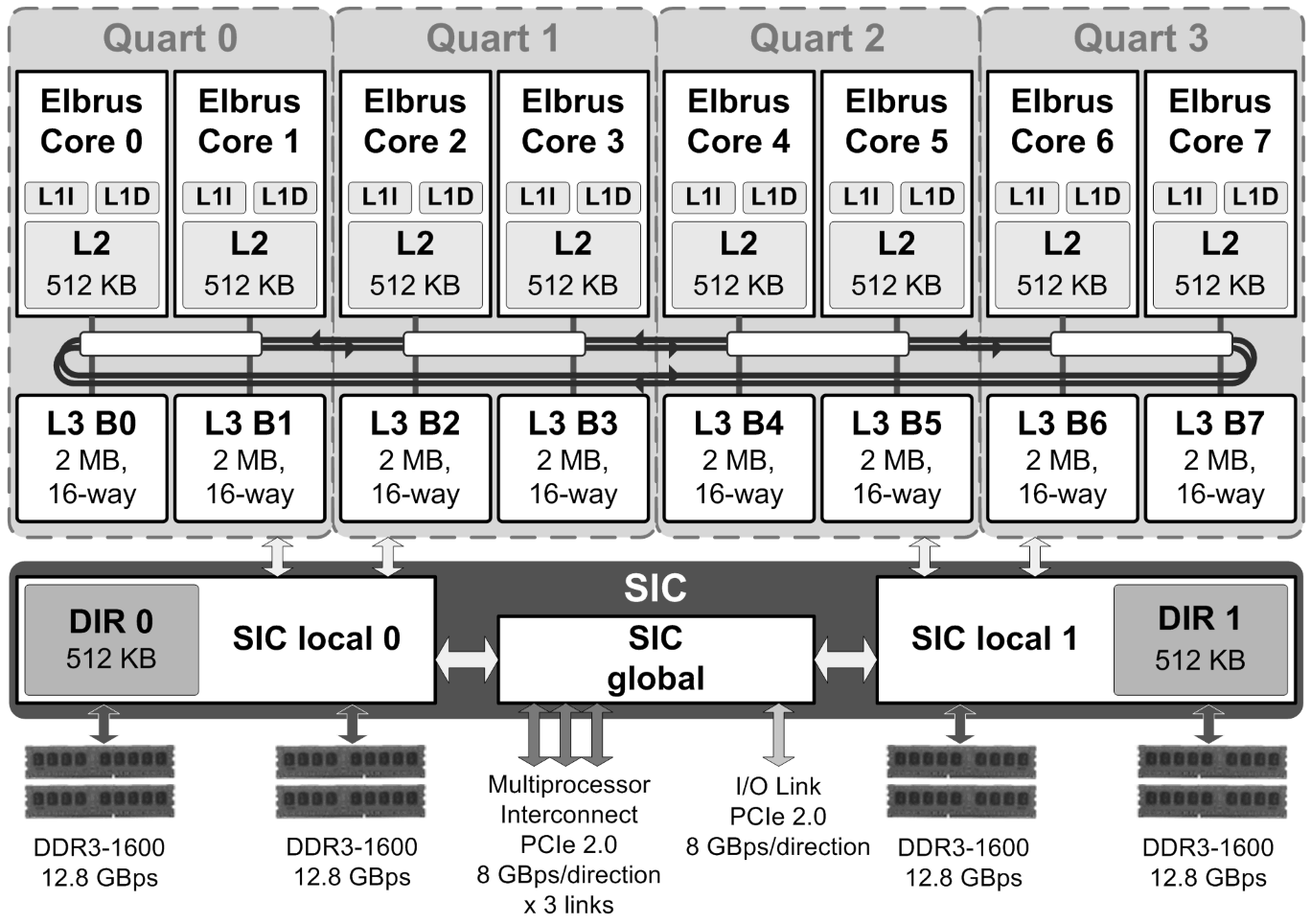


Рисунок 2. Структурная схема микропроцессора «Эльбрус-8C»

Есть возможность объединить 4 процессора в ccNUMA систему:

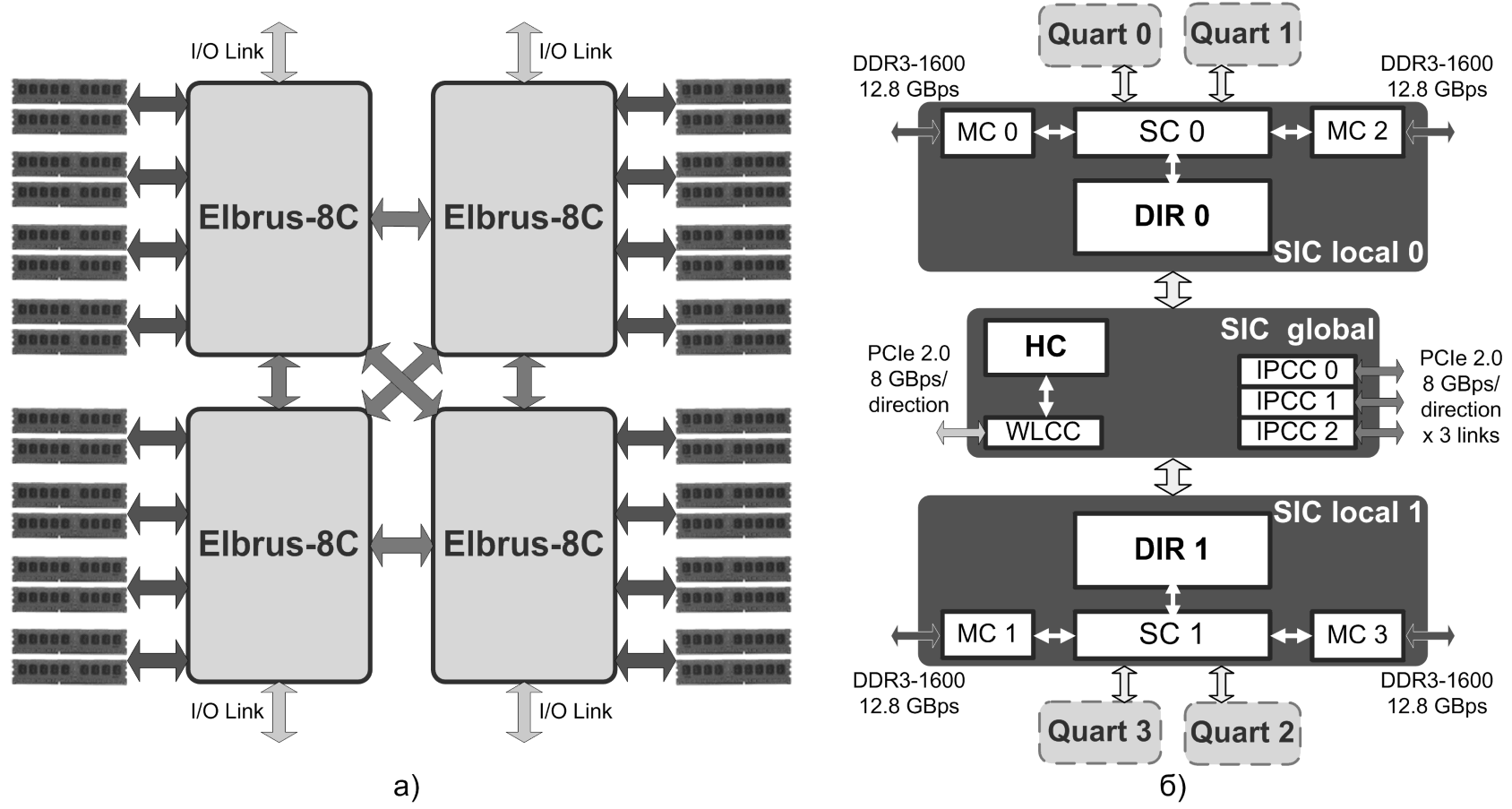
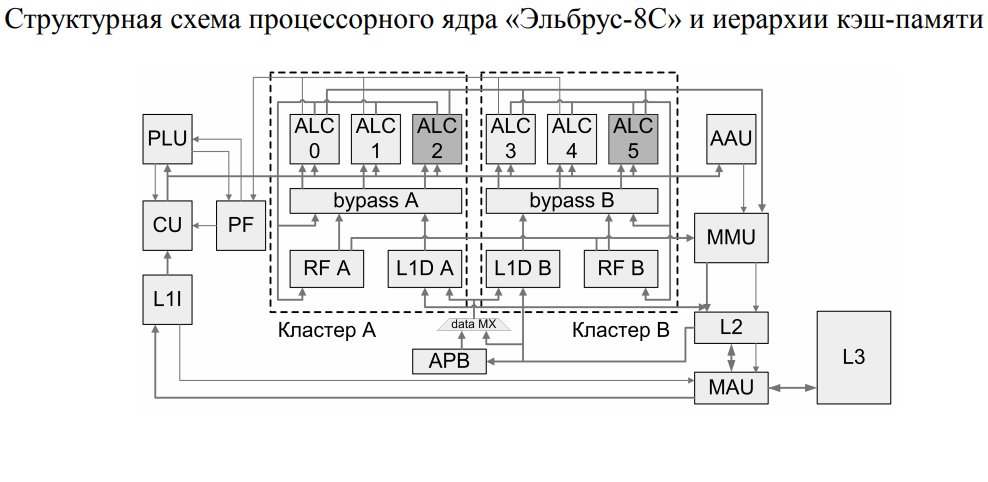


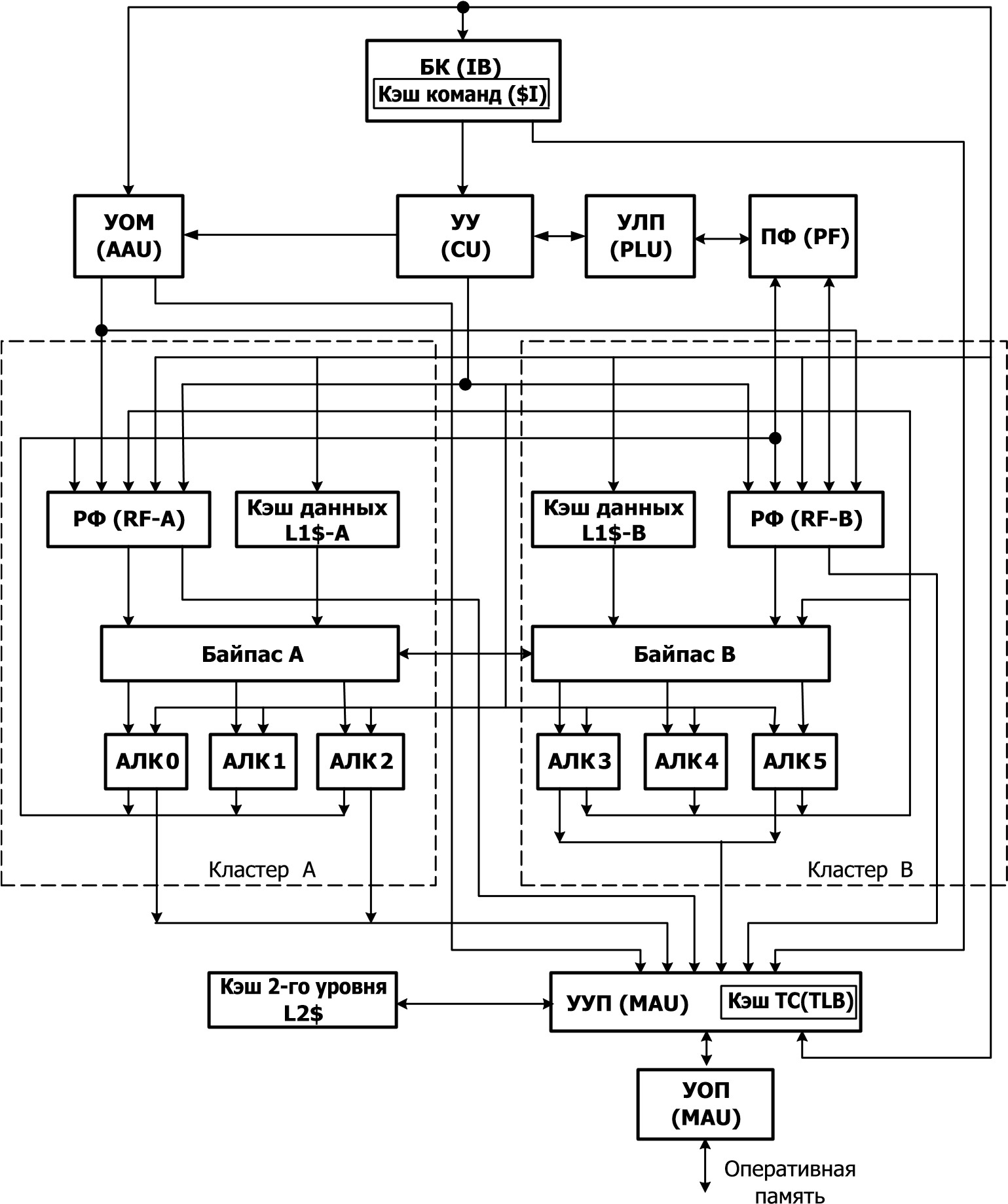
Рисунок 4. Подсистема памяти: а) ccNUMA система с общей когерентной памятью из четырех процессоров «Эльбрус-8C»; б) Структурная схема контроллера системных обменов SIC

**Процессорное ядро:**



При разработке ядра «Эльбрус-8C» за основу было взято ядро микропроцессора предыдущего поколения «Эльбрус-4C». Для повышения производительности ядра было решено увеличить тактовую частоту и число одновременно исполняемых арифметических операций с плавающей точкой.

Рисунок 3. Структурная схема процессорного ядра «Эльбрус-8C» и иерархии кэш-памяти Увеличение числа одновременно исполняемых арифметических операций с плавающей точкой было достигнуто за счет добавления в ALC 2 и 5 возможности исполнения операций сложения и умножения с плавающей точкой. При этом потребовалось модифицировать структуру широкой команды, а именно, организовать упаковку в нее новых слогов для кодирования новых команд ALC 2 и ALC 5 при сохранении обратной совместимости. В результате применения обоих методов удалось повысить производительность ядра почти в 2,5 раза, при этом увеличение его площади составило всего 3,2%.



**БК — буфер команд** (IB — Instruction Buffer) предназначен для вызова программного кода из памяти.

**УУ —** **устройство управления** (CU — Control Unit) выполняет считывание программного кода из буферной памяти БК, распаковку широких команд, их дешифрацию и переключение ветвей программы при выполнении команд переходов.

**ПФ —** **предикатный файл** (PF — Predicate File) хранит первичные предикаты — битовые значения, выработанные операциями сравнения, и вторичные предикаты — результаты логических операций над первичными предикатами.

Предикатный файл — набор 32 двухразрядных регистров (по одному разряду для предиката и тега).

**УЛП —** **устройство логических предикатов** (PLU — Predicate Logic Unit) предназначено для выполнения операций считывания предикатов из файла ПФ и логических операций формирования вторичных предикатов.

**Арифметико-логические каналы (АЛК)** предназначены для исполнения обычных арифметических и логических операций, операций обращения к памяти и обработки адресных данных (дескрипторов, указателей и др.). В состав микропроцессора входят шесть арифметико-логических каналов (АЛК0—АЛК5), разделенных на два кластера. Арифметико-логические каналы работают параллельно и исполняют в основном одинаковый набор операций. В качестве операндов служат данные из РгФ или результаты других исполнительных устройств.

**Регистровый файл (РгФ)** предназначен для хранения локальных данных процедуры и результатов выполненных операций. Он представляет собой сверхоперативное запоминающее устройство с произвольным доступом, обращение которому осуществляется через порты. Многочисленность абонентов регистрового файла требует большого количества портов для обслуживания всех запросов одновременно. С целью сокращения их количества регистровый файл реализован в виде двух одинаковых блоков — РгФ-A и РгФ-B, по одному в каждом кластере. Причем, в отличие от используемых во многих микропроцессорах, блок является общим для целочисленных и вещественных устройств арифметики, что позволило повысить эффективность его использования. Блок РгФ содержит 256 регистров.

**Кэш данных первого уровня L1$** выполнен в виде двух одинаковых блоков (L1$-A и L1$-B) емкостью 64 Кбайт, по одному в каждом кластере. Блоки L1$ хранят одинаковые данные, поскольку запись данных выполняется одновременно в оба блока. В блоке хранятся данные, которые используются в качестве операндов для исполнительных устройств АЛК. Но поскольку в общем случае операнды считываются из регистрового файла, они должны быть предварительно загружены в него из кэша L1$.

**Кэш второго уровня L2$** является общим для данных и программного кода, его объем составляет 256 Кбайт, степень ассоциативности — 4. Обращение к L2$ выполняется при отсутствии требуемых данных в L1$ или нужного программного кода в буферной памяти команд устройства БК. Если нужная информация отсутствует и в L2$, то формируется запрос к оперативной памяти. Считанная из ОП информация поступает потребителю и одновременно записывается в кэш-память вместо устаревших данных.

**УОМ — Устройство обращения к массивам** (AAU — Array Access Unit) предназначено для упреждающей подкачки элементов массива при выполнении векторных операций. Подкачка элементов массивов в буфер, использующий дисциплину очереди (FIFO), осуществляется на фоне выполнения основной (синхронной) программы параллельной ей (асинхронной) программой.

**УУП — устройство управления памятью** (MMU — Memory Management Unit) преобразует виртуальные адреса в физические.

**УОП — устройство обращения в память** (MAU — Memory Access Unit) предназначено для связи микропроцессора с ОП. Обмен с памятью осуществляется через 16-байтовый канал с раздельными шинами для передачи и приема данных. Обмен выполняется блоками по 32 или 64 байт.

# 4. РАСЧЕТ ФУНКЦИЙ НАДЕЖНОСТИ И ГОТОВНОСТИ

Функция (или вероятность безотказной работы) относится к основным показателям надежности ЭВМ. Характеризует производительность ЭВМ на промежутке времени, то есть эта функция обеспечивает потенциально возможную производительность. Функцией надежности ЭВМ называется



где запись  означает вероятность того, что для всякого , принадлежащего промежутку времени  производительность  ЭВМ равна единице, т.е. равна потенциально возможной.

Функция  обладает следующими свойствами:

1.  Т.е. машина в момент начала функционирования находится в работоспособном состоянии.
2.  Событие, заключающееся в том, что ЭВМ работоспособна на конечном промежутке времени, является достоверным.
3.  для ;

Функцией ненадежности (или вероятностью отказа) ЭВМ называется



Функция  позволяет определить среднее время безотказной работы (средняя наработка до отказа). По определению, среднее время  безотказной работы ЭВМ и оценка  соответственно равны:



где – время безотказной работы -й машины, 

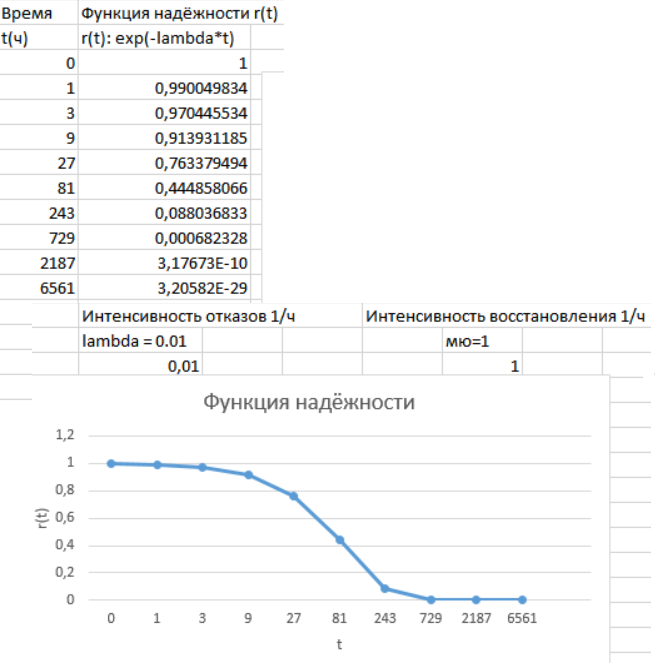
Интенсивностью отказов (лямбда-характеристикой) ЭВМ называется функция



Практически установлено, что зависимость интенсивности отказов от времени имеет место на периоде приработки ЭВМ. После приработки ЭВМ интенсивность отказов остается постоянной (до вхождения в предельное состояние или, по крайней мере, в течение промежутка времени, перекрывающего время морального старения). Следовательно, в нормальных условиях эксплуатации ЭВМ  а функция надежности и математическое ожидание времени *безотказной работы* соответственно равны:



рассчитаем значения функции и построим график:



Теперь рассчитаем значения функции готовности. Функция готовности ЭВМ

есть вероятность того, что (в условиях потока отказов и восстановлений) машина будет иметь в момент времени  производительность, равную единице, т.е. равную потенциально возможной.

Функция готовности ЭВМ обладает следующими свойствами:

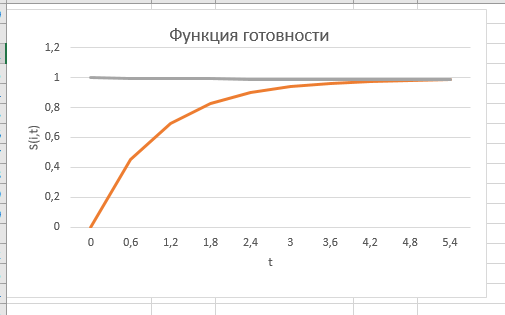
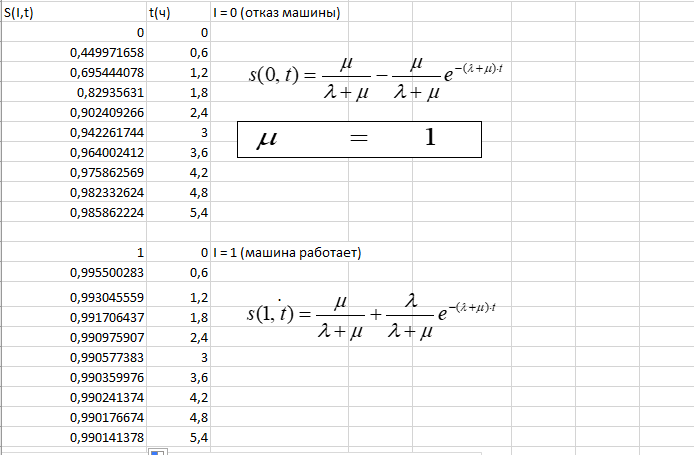
1. 
2. 
3.   для 

Расчет будем производить по следующим формулам:

;

.

для начальных состояний ЭВМ , причем  соответствует состоянию отказа, а  – работоспособному состоянию машины.

Рассчитаем значения функции и построим график: 

+

# 5. СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Хорошевский В.Г. Архитектура вычислительных систем. – М.: МГТУ им. Н.Э. Баумана, 2008. – 520 с.
2. Конспект лекций по курсу “Архитектура вычислительных систем”
3. Ким А. К., Перекатов В. И., Ермаков С. Г. Микропроцессоры и вычислительные комплексы семейства «Эльбрус». — СПб.: Питер, 2013. — 272 с