



Санкт-Петербургский государственный университет  
аэрокосмического приборостроения

Лабораторные работы  
по дисциплине  
**«Проектирование на системном уровне»**  
**(System Level Design)**

Разработаны:

Е.А. Суворова

к.т.н., ассистент кафедры Информационных систем

Санкт-Петербург  
2005 г.

## ЛАБОРАТОРНАЯ РАБОТА 1.

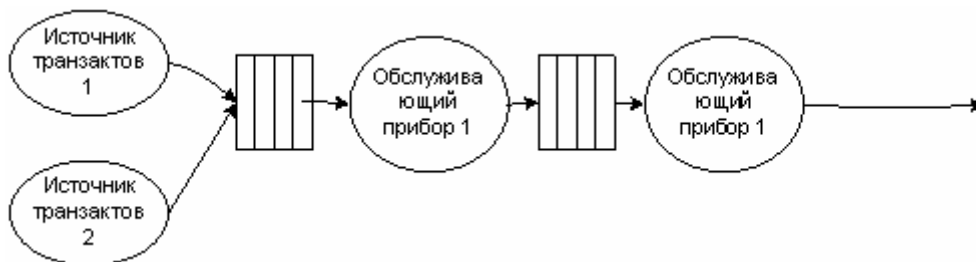
Тема: **Изучение основ SystemC и основ работы с Incisive Unified Simulator.**

Задание:

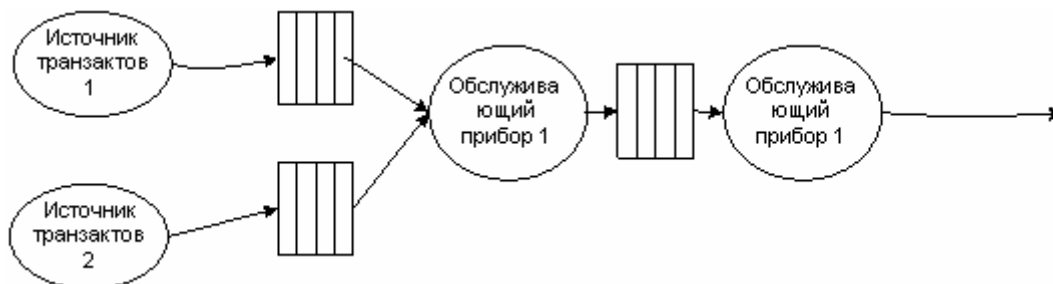
Выполнить описание стохастической сети на SystemC. Промоделировать работу сети в Incisive Unified Simulator, убедиться в правильности ее функционирования.

Варианты задания:

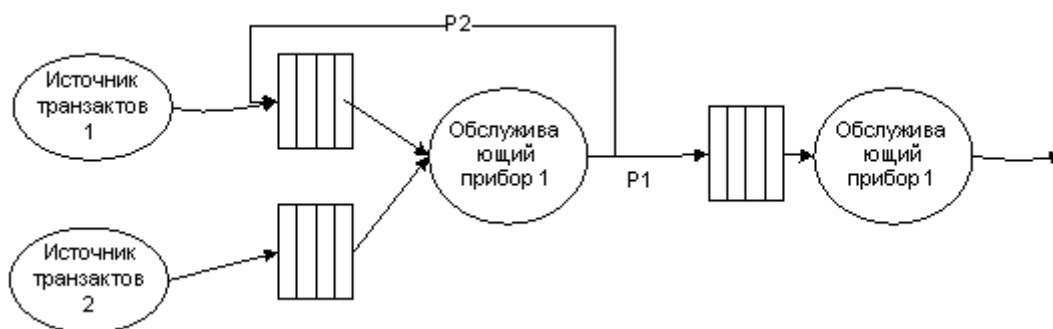
1.



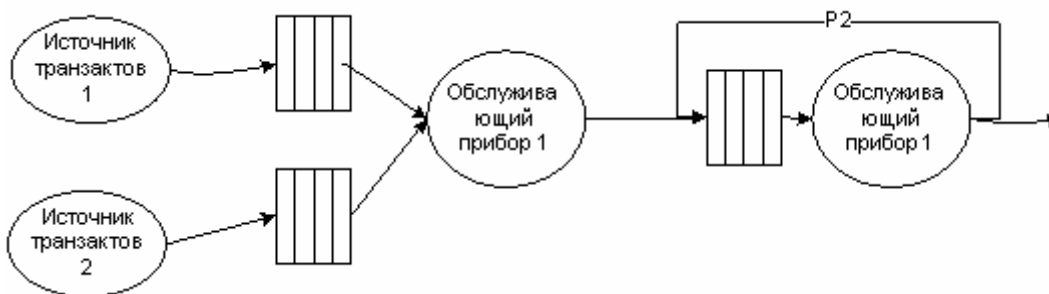
2.



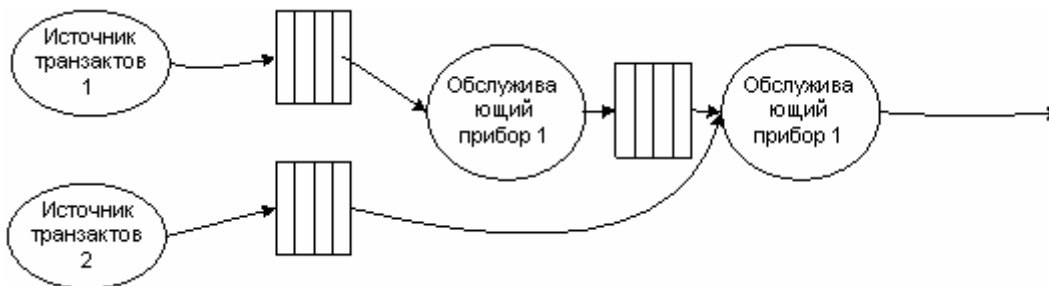
3.



4.



5.



### Используемое программное обеспечение

Поток проектирования Cadence IUS 5.3: утилиты nscs, simvisio

### Порядок выполнения работы:

1. Выполнить описание объекта моделирования на SystemC в любом текстовом редакторе (например, в gedit, kwrite)
2. С помощью утилиты nscs осуществить компиляцию проекта, исправить выявленные ошибки. Загрузить проект в среду моделирования.
3. Выполнить тестирование правильности функционирования объекта моделирования.

### Содержание отчета:

1. Постановка задачи, вариант задания
2. Текст программы на языке SystemC
3. Результаты моделирования

## ЛАБОРАТОРНАЯ РАБОТА 2.

Тема: **Изучение использования транзакторов для моделирования системы коммуникаций между функциональными блоками.**

Задание:

Промоделировать взаимодействие между ведущими и ведомыми устройствами через коммуникационную систему.

Модели ведущих и ведомых устройств, коммуникационной системы формируются на базе транзакторов. В работе рассматривается два варианта коммуникационной системы – на базе шины и на базе коммутатора. В качестве блока, реализующего функции коммуникационной системы используется готовый компонент. Ведомые устройства представляют собой памяти. Ведущие устройства обращаются к ведомым устройствам в соответствии с различными схемами. Количество ведущих и ведомых устройств в системе, схемы обращения ведущих устройств к ведомым, времена обработки обращений ведомыми устройствами определяются в соответствии с вариантом.

Варианты задания :

№ варианта	Количество ведущих устройств	Количество ведомых устройств	Номер схемы обращений для ведущих устройств	Номер схемы обработки обращений для ведомых устройств
1	4	4	1	1
2	4	2	2	2
3	4	6	3	1
4	2	4	4	2
5	2	2	1	1
6	2	6	2	2
7	4	4	3	1
8	4	2	4	2
9	2	6	1	1
10	2	4	2	2

Схемы обращений для ведущих устройств.

1. Каждое ведущее устройство равновероятно обращается ко всем ведомым устройствам. Следующее обращение не начинается до тех пор, пока не будет выполнено предыдущее. Время между завершением предыдущего

- обращения и началом следующего равномерно распределено на интервале от 0 до 10 тактов. Вероятность того, что обращение будет являться запросом на чтение –  $P_1$  ( $P_1=0,3; 0,5; 0,7$ ). Начальные адреса обращения для всех ведомых устройств одинаковы. Адреса следующих обращений вычисляются по формуле  $A_i=A_{i-1}+N$ , где  $N$  – порядковый номер ведущего устройства
2. Каждое ведущее устройство с четным номером обращается только к ведомым устройствам с четными номерами, каждое ведущее устройство с нечетным номером обращается только к ведомым устройствам с нечетными номерами. Следующее обращение не начинается до тех пор, пока не будет выполнено предыдущее. Время между завершением предыдущего обращения и началом следующего равномерно распределено на интервале от 0 до 8 тактов. Вероятность того, что обращение будет являться запросом на чтение –  $P_1$  ( $P_1=0,2; 0,5; 0,8$ ). Адреса следующих обращений вычисляются по формуле  $A_i=A_{i-1}+N$ , где  $N$  – порядковый номер ведущего устройства
  3. Каждое ведущее устройство равновероятно обращается ко всем ведомым устройствам. Следующее обращение может быть начато до завершения выполнения предыдущего. Для одного ведущего устройства в один момент времени может быть  $N$  невыполненных обращений  $N=2, 4, 6$  (при достижении этого числа дальнейшая генерация обращений приостанавливается) Время между началом предыдущего обращения и началом следующего равномерно распределено на интервале от 0 до 10 тактов. Вероятность того, что обращение будет являться запросом на чтение –  $P_1$  ( $P_1=0,5$ ). Адреса следующих обращений вычисляются по формуле  $A_i=A_{i-1}*N$ , где  $N$  – порядковый номер ведущего устройства
  4. Каждое ведущее устройство с четным номером обращается только к ведомым устройствам с четными номерами, каждое ведущее устройство с нечетным номером обращается только к ведомым устройствам с нечетными номерами. . Следующее обращение может быть начато до завершения выполнения предыдущего. Для одного ведущего устройства в один момент времени может быть  $N$  невыполненных обращений  $N=3, 5, 7$  (при достижении этого числа дальнейшая генерация обращений приостанавливается) Время между завершением предыдущего обращения и началом следующего равномерно распределено на интервале от 0 до 12 тактов. Вероятность того, что обращение будет являться запросом на чтение –  $P_1$  ( $P_1=0,5$ ). Адреса следующих обращений вычисляются по формуле  $A_i=A_{i-1}*N$ , где  $N$  – порядковый номер ведущего устройства

Схемы обработки обращений для ведомых устройств.

1. Операция записи выполняется за 1 такт, операция чтения выполняется за  $K$  тактов,  $K=1, 2, 4$

2. Если адрес следующего обращения и адрес предыдущего обращения попадают в один блок, обращение выполняется за 1 такт, в противном случае – за 4 такта. Размер блока  $B = 64, 128, 256$  слов

### Используемое программное обеспечение

Поток проектирования Cadence IUS 5.3: утилиты `ncsc`, `simvisio`

### Порядок выполнения работы:

1. Выполнить описание ведущих и ведомых устройств на SystemC в любом текстовом редакторе (например, в `gedit`, `kwrite`)
2. Выполнить описание проекта в целом в любом текстовом редакторе. Описание должно обеспечивать в зависимости от значения параметра подключать блок коммуникационной системы на базе общей шины или на базе коммутатора.
3. С помощью утилиты `ncsc` осуществить компиляцию проекта, исправить выявленные ошибки. Загрузить проект в среду моделирования.
4. Выполнить тестирование правильности функционирования объекта моделирования.
5. Определить по результатам работы модели и теоретически загрузку коммуникационной системы, время простоя ведущих и ведомых устройств при различных параметрах обращений
6. Выбрать тип коммуникационной системы, лучшим образом соответствующий моделируемой задаче.

### Содержание отчета:

1. Постановка задачи, вариант задания
2. Алгоритмы функционирования ведущих и ведомых устройств
3. Текст программы на языке SystemC
4. Результаты моделирования

### ЛАБОРАТОРНАЯ РАБОТА 3.

Тема: **Моделирование работы конвейера процессора на уровне транзакций.**

#### Задание:

Разработать модель конвейера процессора.

Все арифметические команды процессора являются двухоперандными, результат записывается по адресу второго операнда. Все команды перехода явно используют один операнд.

Ступени конвейера, которые выполняют обращения к памяти, реализуются на базе транзакторов.

Операнды могут находиться в регистрах процессора, кэш или ОЗУ. Вероятность того, что операнд будет расположен в регистре - P1, в кэш – P2. (значения этих параметров определяются вариантом). В данной работе модель системы памяти, включающей в себя кэш и ОЗУ, представляет собой одно устройство, эмулирующее обращения в кэш и в ОЗУ (для различных типов обращений в соответствии с вариантом определены различные времена обработки: время обращения в кэш – 4 такта, время обращения в ОЗУ – 10 тактов).

Набор команд процессора включает в себя арифметические операции и операции условного перехода. Вероятность того, что команда является арифметической – P3.

Промоделировать работу конвейера в двух вариантах:

1. Если в конвейер поступает команда перехода, дальнейшая выборка команд приостанавливается до тех пор, пока данная команда перехода не достигнет ступени, на которой вычисляется адрес перехода
2. Если в конвейер поступает команда перехода, то дальнейшая выборка команд продолжается. Вероятность того, что будет выбираться правильная последовательность команд – P4

Для выполнения работы используются утилиты ncsc, simvisio

#### Варианты задания:

N варианта	Тип архитектуры	P1	P2	P3	P4	Количество интерфейсов с ОЗУ данных
1	гарвардская	0,1; 0,4	0,4	0,9	0,5; 0,3	1
2	принстонская	0,3	0,3; 0,5	0,8	0,5; 0,9	1
3	гарвардская	0,2	0,6	0,6; 0,8	0,7; 0,3	2
4	принстонская	0,1; 0,2	0,7	0,7	0,5; 0,3	2

5	гарвардская	0,1; 0,3	0,5	0,8	0,5; 0,9	1
6	принстонская	0,2	0,4; 0,6	0,9	0,7; 0,3	1
7	гарвардская	0,3	0,3; 0,6	0,7	0,5; 0,3	2
8	принстонская	0,1	0,7	0,8; 0,9	0,5; 0,9	2
9	гарвардская	0,2	0,6	0,7; 0,9	0,7; 0,3	1
10	принстонская	0,3	0,4	0,8; 0,9	0,5; 0,3	1

### Используемое программное обеспечение

Поток проектирования Cadence IUS 5.3: утилиты ncsc, simvisio

### Порядок выполнения работы:

1. Выполнить описание объекта моделирования на SystemC в любом текстовом редакторе (например, в gedit, kwrite)
2. промоделировать работу системы, оценить загрузку ступеней конвейера, среднее время простоя ступеней конвейера, среднее время (количество тактов) между последовательными обращениями к подсистеме памяти.
3. Сравнить результаты, полученные при моделировании с теоретическими расчетами.

### Содержание отчета:

1. Постановка задачи, вариант задания
2. Алгоритмы работы ступеней конвейера
3. Текст программы на языке SystemC
4. Результаты моделирования, сравнение с теоретическими расчетами



## ЛАБОРАТОРНАЯ РАБОТА 4.

Тема: **Моделирование работы RISC процессора подсистемы памяти на уровне транзакций.**

**Задание:**

Промоделировать работу подсистемы памяти, включающей в себя кэш и ОЗУ в многозадачной системе.

Интерфейсы всех устройств реализуются на базе транзакторов. Для организации системы коммуникаций используется готовый компонент.

В системе выполняется 4 задачи.  $T_s$  – среднее время между переключением задач (определяется в количестве тактов).  $A_i$  – начальный адрес области памяти, в которой расположены данные для задачи  $i$ ,  $L_i$  – размер области памяти, в которой расположены данные для задачи  $i$ .

$T_p$  – среднее время между обращениями к подсистеме памяти (определяется в количестве тактов, в этой работе используется значение, полученное в работе 3).  $T_c$  – количество тактов, необходимое для выборки слова данных из кэш.  $T_m$  – количество тактов, необходимо е для выборки слова данных из памяти.  $N_s$  – количество слов в строке кэш.

Схемы определения адреса следующего обращения (С):

1. Адреса случайно распределены по всей области адресного пространства задачи.
2. Каждый последующий адрес на 1 больше предыдущего. Когда достигается максимальный адрес выделенной области памяти, происходит переход к началу области памяти
3. Каждый последующий адрес на 10 больше предыдущего. Когда достигается максимальный адрес выделенной области памяти, происходит переход к началу области памяти
4. Каждый последующий адрес на  $N_s+2$  больше предыдущего. Когда достигается максимальный адрес выделенной области памяти, происходит переход к началу области памяти
5. Каждый последующий адрес на 5 меньше предыдущего. Когда достигается максимальный адрес выделенной области памяти, происходит переход к началу области памяти

Промоделировать работу системы без кэш и при использовании прямо отображаемого, 2-way и 4-way кэш. Определить тип подсистемы памяти, наилучшим образом соответствующий схеме обращения к памяти для решаемых задач.

Варианты задания:

N варианта	Ts	Ai	Li	Tc	Tm	Ns	C
1	200	1 – 0 2 – 0 3 – 0 4 – 0	1 – 256 2 – 1024 3 – 256 4 – 512	2	10	32	1 – 1 2 – 2 3 – 3 4 – 4
2	500	1 – 0 2 – 256 3 – 512 4 – 768	1 – 256 2 – 256 3 – 256 4 – 256	2, 3, 4	10	32	1 – 5 2 – 1 3 – 2 4 – 3
3	1000	1 – 0 2 – 0 3 – 0 4 – 0	1 – 256 2 – 256 3 – 256 4 – 256	2	5, 8, 12	32	1 – 4 2 – 5 3 – 1 4 – 2
4	200	1 – 0 2 – 256 3 – 512 4 – 768	1 – 256 2 – 1024 3 – 256 4 – 512	2	5	64	1 – 3 2 – 4 3 – 5 4 – 1
5	500	1 – 0 2 – 1024 3 – 0 4 – 512	1 – 256 2 – 256 3 – 256 4 – 256	2, 3, 4	8	64	1 – 2 2 – 3 3 – 4 4 – 5
6	1000	1 – 0 2 – 0 3 – 512 4 – 512	1 – 256 2 – 256 3 – 256 4 – 256	2	5, 7, 9	64	1 – 1 2 – 2 3 – 3 4 – 4
7	200	1 – 0 2 – 0 3 – 512 4 – 512	1 – 512 2 – 256 3 – 512 4 – 256	3	10	16	1 – 5 2 – 1 3 – 2 4 – 3
8	500	1 – 0 2 – 0 3 – 512 4 – 512	1 – 1024 2 – 1024 3 –	2, 3, 4	10	16	1 – 4 2 – 5 3 – 1 4 – 2

			1024 4 – 1024				
9	1000	1 – 0 2 – 0 3 – 256 4 – 256	1 – 512 2 – 256 3 – 512 4 – 256	2	5, 10, 15	16	1 – 3 2 – 4 3 – 5 4 – 1
10	500	1 – 0 2 – 0 3 – 256 4 – 256	1 – 256 2 – 256 3 – 256 4 – 256	3	5, 10, 15	32	1 – 2 2 – 3 3 – 4 4 – 5

### Используемое программное обеспечение

Поток проектирования Cadence IUS 5.3: утилиты nesc, simvisio

### Порядок выполнения работы:

1. Выполнить описание объекта моделирования на SystemC в любом текстовом редакторе (например, в gedit, kwrite)
2. промоделировать работу системы, оценить время простоя процессора по причине ожидания данных
3. Сравнить результаты, полученные при моделировании с теоретическими расчетами.

### Содержание отчета:

1. Постановка задачи, вариант задания
2. Текст программы на языке SystemC
3. Результаты моделирования, сравнение с теоретическими расчетами