

Программа и методики экспериментальных исследований ПО

УТВЕРЖДАЮ

ПРОГРАММА И МЕТОДИКИ ИСПЫТАНИЯ

Программный модуль StreamManager
А.В.00001-01 ПМ

Условные обозначения и сокращения, принятые в тексте

АЛУ	Арифметико-логическое устройство
ПД	Программная документация
ПМ	Программа и методики

1. Общие положения

1.1 Наименование и обозначение программного модуля (далее – модуль).

Программный модуль реализации потокового менеджера

1.2. Цель экспериментальных исследований

- исследование технических соответствия программного модуля (далее – программы) функциям, установленной требованиями задания на выпускную квалификационную работу.

1.3 Условия предъявления программы на экспериментальные исследования

1.3.1 Программа предъявляется для экспериментальных исследований на следующем носителе:

флэш-диск

в следующей комплектности:

носитель;

комплект эксплуатационной документации (руководство программиста).

1.3.5 Комплекс предъявляется на экспериментальных исследованиях в сопровождении следующих документов:

задание выпускную квалификационную работу;

комплект ПД;

настоящая ПМ.

2 Общие требования к условиям, обеспечению и проведению экспериментальных исследований

2.1 Место проведения экспериментальных исследований

Экспериментальные исследования проводятся на базе «Московского института электроники и математики Национального исследовательского университета «Высшая школа экономики»» в департаменте компьютерной инженерии.

2.2 Требования к средствам проведения экспериментальных исследований

2.2.1 Перечень средств проведения экспериментальных исследований приведен в приложении Б.

2.3 Требования к подготовке программы к экспериментальным исследованиям

2.4 Требования к персоналу, осуществляющему подготовку к экспериментальным исследованиям.

К проведению экспериментальных исследований допускается персонал, изучивший техническую документацию на программу.

При проведении работ при проверке и программы персонал обязан соблюдать правила техники безопасности согласно «Правилам технической эксплуатации электроустановок потребителей» и «Правилам техники безопасности при эксплуатации электроустановок потребителей».

3 Требования безопасности

Не предъявляются.

4 Программа экспериментальных исследований

Определяемые показатели и точность их измерений:

Пункт программы эксп. исслед.	Наименование Показателя	Пункт требований ТЗ	Ед. изм.	Номинальное значение	Предельные отклонения	Пункт методик и
4.1.	Проверка корректности работы программы при синтаксическом анализе текста	п. 6.1.1	-	-	0	6.1

5 Режимы экспериментальных исследований

5.1. Порядок экспериментальных исследований

Проведение экспериментальных исследований осуществляется руководителем выпускной квалификационной работы.

Последовательность проведения экспериментальных исследований может быть изменена по руководителю выпускной квалификационной работы.

5.2. Ограничения и другие указания, которые необходимо выполнять на всех или на отдельных режимах экспериментальных исследований.

Экспериментальные исследования прекращаются в случаях:

- несоответствия получаемых результатов требованиям ТЗ;
- трехкратного неудачного запуска текста;
- выход из строя оборудования.

5.3 Условия перерыва, аннулирования и возобновления экспериментальных исследований на всех или на отдельных режимах.

Необходимость, условия и порядок перерыва, аннулирования или прекращения экспериментальных исследований определяется комиссией.

6 Методы экспериментальных исследований

6.1 Экспериментальное исследование по п. 4.1. Программы выполняется следующим образом.

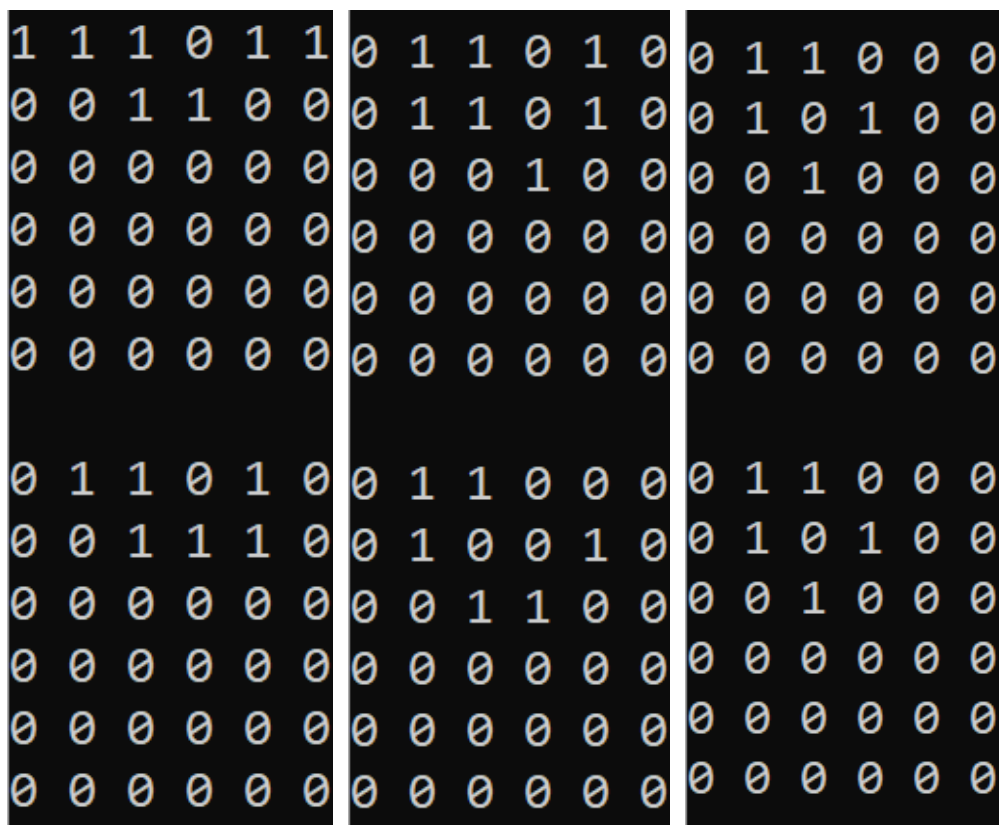
Последовательность действий:

1. Запуск «Среды программирования и имитационного моделирования объектно-атрибутной суперкомпьютерной системы с управлением потоком данных» (файл millicom.exe).

3. Загрузка файла ALUCellular.oar в среде программирования.

4. Запуск программы.

5. Проверка полученного результата. Корректный результат должен быть:



7 Отчетность

7.1 Заданные и фактические данные, полученные при экспериментальных исследованиях по каждому пункту программы, оформляются протоколами. В согласованных случаях допускается оформлять одним протоколом данные, полученные при экспериментальных исследованиях по нескольким пунктам программы.

7.2 По результатам экспериментальных исследований руководитель выпускной квалификационной работы составляется акт экспериментальных исследований.

Акт экспериментальных исследований должен содержать:

подтверждение выполнения программы экспериментальных исследований;

оценку результатов экспериментальных исследований с конкретными точными формулировками, отражающими соответствие испытуемого программного обеспечения требованиям задания на выпускную квалификационную работу;

выводы по результатам экспериментальных исследований;

К акту прилагаются протоколы экспериментальных исследований по пунктам программы.

Приложения

Приложение А	Перечень ссылочных документов
Приложение Б	Перечень средств измерений и испытательного оборудования, необходимых для проведения экспериментальных исследований
Приложение В	Программа на ОА-языка для тестирования программы

Перечень ссылочных документов

ГОСТ 15150	Машины, приборы и другие технические средства. Исполнение для различных климатических районов. Категория, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды.
ОСТ 88 211	Приборы и средства автоматизации для научных исследований. Приемка, маркировка, упаковка, транспортирование, хранение и гарантии изготовителя.
ГОСТ 21657	Электрическая изоляция изделий ГСП. Технические требования и методики экспериментальных исследований. «Правила техники безопасности при эксплуатации электроустановок потребителей». М., Госэнергонадзор, 1994 «Правила технической эксплуатации электроустановок потребителей», М., Энергосервис, 2003 г.

Приложение Б
к программе и методикам экспериментальных исследований ПО

Перечень средств проведения экспериментальных исследований

Наименование, тип и марка	Кол- во	ГОСТ, ТУ или обозначение	Основные характеристики
Компьютер IBM-совместимый	1		Процессор Pentium4 от 1 ГГц; Оперативная память от 1 Мб; Дисковое пространство – от 1 Гб; От 32 Мб видеопамати
Монитор	1		
Программное обеспечение		Операционная система Windows 7 и выше	

Программа на ОА-языке для тестирования модуля

```
// - обозначение переведенных в комментарии старых ФУ
\* date *\ //- последняя дата изменения или добавления строк и
ФУ

NewFU={Mnemo="Main_Bus" MkBegRange=1000 FUType=FUBusNew} //
Описание Bus

\\ Для создания индексного файла сначала надо запустить
программу !!!!
CapsManager.IndexVectCreate=20000
CapsManager.IpIcIdOutMk=MainBus.IpIcIdSet
CapsManager.IpIcIdOutMk=VariableManager.IpIcIdSet
CapsManager.IpIcIdOutMk=ListSyntez.IpIcIdSet
MainBus.ModeSet=1
\\ ----- \\
MainBus.FUTypeCorrectSet=-96 // Корректировка номера типа ФУ для
переноса на новую ОА-платформу

NewFU={Mnemo="Console" FUType=FUConsNew}
NewFU={Mnemo="Manager" FUType=FUStreamManager Hint="Менеджер
поля АЛУ"}
//NewFU={Mnemo="IntALU" FUType=FUIntAluNew}
NewFU={Mnemo="SrteamIntALU" FUType=FUStreamIntALU}

MainBus.PartialResetSet \\ Запомнить число созданных ФУ
MainBus.ModeSet=2

N=6 // Высота поля
M=6 // Ширина поля
K=6 // Количество итераций

Manager.CounterSet=N
Manager.CounterMul=M
Manager.FieldCreate=FUStreamIntALU // Создать поле целочисленных
АЛУ
Manager.ExecAll={FUStreamIntALU.NOoperandSet=0
FUStreamIntALU.ReadySet=1 FUStreamIntALU.OutRezBlockSet=True}
Manager.ExecAll={FUStreamIntALU.PreRezProgSet={FUStreamIntALU.Pu
sh}} // Программа перед получением результата

Manager.ExecAll={FUStreamIntALU.RezProgSet= // Программа
вычисления значения ячейки (запускается после вычисления суммы)
{
    FUStreamIntALU.Swap // Поменять местами сумму и старое
значение ячейки, которое было сохранено в стеке результата
    FUStreamIntALU.AccumModeSet=True // Переход в
аккумуляторный режим
    FUStreamIntALU.ZeroExec={ // Если клетка пустая (при 0 в
регистре результата
```



```

        FUStreamIntALU.PopMk // Выталкиваем из стека
результата в регистр результата
        FUStreamIntALU.Equal=3 // Если равно 3 (результат
записывается в регистр результата
        FUStreamIntALU.AccumModeSet=False // Выключение
аккумуляторного режима
        ProgStopAll // Выход из подпрограмм
    }
// Если клетка занята
    FUStreamIntALU.StackOutMk
    FUStreamIntALU.Equal=2 // Проверка двух соседей
    FUStreamIntALU.NZeroExec={ //Если 2 соседа
        FUStreamIntALU.StackClear
        FUStreamIntALU.AccumModeSet=False
        ProgStopAll}
    FUStreamIntALU.StackOutMk=FUStreamIntALU.Set
    FUStreamIntALU.Equal=3 // Проверка трех соседей
    FUStreamIntALU.StackClear
    FUStreamIntALU.AccumModeSet=False
    }
}

//Генерация сетки

// Горизонтальные связи
Manager.IndGroupSet=0
Manager.IndSet=0 // Установить индекс первого устройства
Manager.Ind2Set=1 // Установить индекс второго устройства
Manager.ExecCounterSet=N // Количество запусков по количеству
строк
Manager.Mk2Set=FUStreamIntALU.ContextOutMk // Вторая МК для
Выдачи контекста
Manager.MkSet=FUStreamIntALU.ReceiverAdd // Первая МК для
Добавления ссылки на получателя результата
Manager.ProgExec={
    Manager.ExecCounterSet=M
    Manager.ExecCounterSub=1 // M -1 циклов по строке
    FUStreamManager.ProgExec={
        FUStreamManager.Mk2Exec=FUStreamManager.MkExec //
Заслать ссылку из 2-го ФУ для первого (тот его запишет в
качестве приемника результата)

        FUStreamManager.ExecDev={FUStreamIntALU.ReceiverMkAdd=FUStr
eamIntALU.Add FUStreamIntALU.NOperandAdd} // Добавить МК для
приёмника результат и увеличить количество операндов у приёмника
        FUStreamManager.IndSwap // Поменять индексы ФУ
        FUStreamManager.Mk2Exec=FUStreamManager.MkExec

        FUStreamManager.ExecDev={FUStreamIntALU.ReceiverMkAdd=FUStr
eamIntALU.Add FUStreamIntALU.NOperandAdd}
        FUStreamManager.IndSwap
        FUStreamManager.Ind2Add=1 //увеличить индексы ФУ
(перейти к следующей паре

```

```

        FUStreamManager.IndAdd=1
    }
    FUStreamManager.IndAdd=1 // Пропустить край поля
    FUStreamManager.Ind2Add=1
}

// Вертикальные связи
Manager.IndSet=0
Manager.Ind2Set=M
Manager.ExecCounterSet=M
Manager.ProgExec={
    Manager.ExecCounterSet=N
    Manager.ExecCounterSub=1
    FUStreamManager.ProgExec={
        FUStreamManager.Mk2Exec=FUStreamManager.MkExec

        FUStreamManager.ExecDev={FUStreamIntALU.ReceiverMkAdd=FUStreamIntALU.Add FUStreamIntALU.NOoperandAdd}
        FUStreamManager.IndSwap
        FUStreamManager.Mk2Exec=FUStreamManager.MkExec

        FUStreamManager.ExecDev={FUStreamIntALU.ReceiverMkAdd=FUStreamIntALU.Add FUStreamIntALU.NOoperandAdd}
        FUStreamManager.IndSwap
        FUStreamManager.Ind2Add=M
        FUStreamManager.IndAdd=M
    }
    FUStreamManager.IndMod=M
    FUStreamManager.IndAdd=1
    FUStreamManager.IndOutMk=FUStreamManager.Ind2Set
    FUStreamManager.Ind2Add=M
}

// Связи слева вверху направо внизу
Manager.Ind2Set=M
Manager.Ind2Add=1
Manager.IndSet=0
Manager.ExecCounterSet=M
Manager.ExecCounterSub=1
Manager.ProgExec={
    Manager.ExecCounterSet=N
    Manager.ExecCounterSub=1
    FUStreamManager.ProgExec={
        FUStreamManager.Mk2Exec=FUStreamManager.MkExec

        FUStreamManager.ExecDev={FUStreamIntALU.ReceiverMkAdd=FUStreamIntALU.Add FUStreamIntALU.NOoperandAdd}
        FUStreamManager.IndSwap
        FUStreamManager.Mk2Exec=FUStreamManager.MkExec

        FUStreamManager.ExecDev={FUStreamIntALU.ReceiverMkAdd=FUStreamIntALU.Add FUStreamIntALU.NOoperandAdd}
        FUStreamManager.IndSwap
    }
}

```

```

        FUStreamManager.Ind2Add=1
        FUStreamManager.IndAdd=1
    }
    FUStreamManager.IndAdd=1
    FUStreamManager.Ind2Add=1
}

// Связи слева внизу направо вверх
Manager.IndSet=1
Manager.Ind2Set=M
Manager.ExecCounterSet=M
Manager.ExecCounterSub=1
Manager.ProgExec={
    Manager.ExecCounterSet=N
    Manager.ExecCounterSub=1
    FUStreamManager.ProgExec={
        FUStreamManager.Mk2Exec=FUStreamManager.MkExec

        FUStreamManager.ExecDev={FUStreamIntALU.ReceiverMkAdd=FUStreamIntALU.Add
FUStreamIntALU.NOperandAdd}
        FUStreamManager.IndSwap
        FUStreamManager.Mk2Exec=FUStreamManager.MkExec

        FUStreamManager.ExecDev={FUStreamIntALU.ReceiverMkAdd=FUStreamIntALU.Add
FUStreamIntALU.NOperandAdd}
        FUStreamManager.IndSwap
        FUStreamManager.Ind2Add=1
        FUStreamManager.IndAdd=1
    }
    FUStreamManager.IndAdd=1
    FUStreamManager.Ind2Add=1
}

//Manager.ExecAll={FUStreamIntALU.ReceiverCountOutMk=Console.OutLn}
// Расстановка точек на поле
Manager.IndSet=0
Manager.ExecDev={FUStreamIntALU.Set=1} // Запись точки в регистр
результата и в буфер рассылки
Manager.IndSet=1
Manager.ExecDev={FUStreamIntALU.Set=1}
Manager.IndSet=2
Manager.ExecDev={FUStreamIntALU.Set=1}
Manager.IndSet=4
Manager.ExecDev={FUStreamIntALU.Set=1}
Manager.IndSet=5
Manager.ExecDev={FUStreamIntALU.Set=1}
Manager.IndSet=8
Manager.ExecDev={FUStreamIntALU.Set=1}
Manager.IndSet=9
Manager.ExecDev={FUStreamIntALU.Set=1}

Manager.ExecAll={FUStreamIntALU.BufSet} //

```

```
Console.ArrayBracketStartSet="" // Настройка консоли
Console.ArrayBracketFinSet=""
Console.VectColSet=M

//Manager.RezBufVectOutMk=Console.OutLn // Выдать начальное
расположение точек
Manager.RezVectOutMk=Console.OutLn // Выдать начальное
расположение точек

Manager.ExecCounterSet=K
Manager.ProgExec={
    FUStreamManager.ExecAll={FUStreamIntALU.BufSend} //
    Рассылка значений из буфера рассылки
    FUStreamManager.ExecAll={FUStreamIntALU.BufSet} //
    Переписать значение из регистра результат в буфер рассылки
    FUStreamManager.RezVectOutMk=Console.OutLn // Выпод поля
}
```