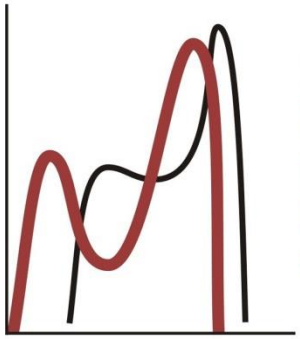


ИНСТИТУТ
МАТЕМАТИКИ
МЕХАНИКИ
КОМПЬЮТЕРНЫХ
НАУК

имени И.И. Воровича —

Архитектура компьютера и операционные системы

Лекция 8. Уровень микроархитектуры

Андреева Евгения Михайловна
доцент кафедры информатики и вычислительного эксперимента

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

План лекции

- Расписание контрольных работ
- Микроархитектура
  - Тракт данных Mic-1
  - Микропрограммное управление Mic-1
  - Микроассемблер для Mic-1 и реализация IJVM
- Домашнее задание

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

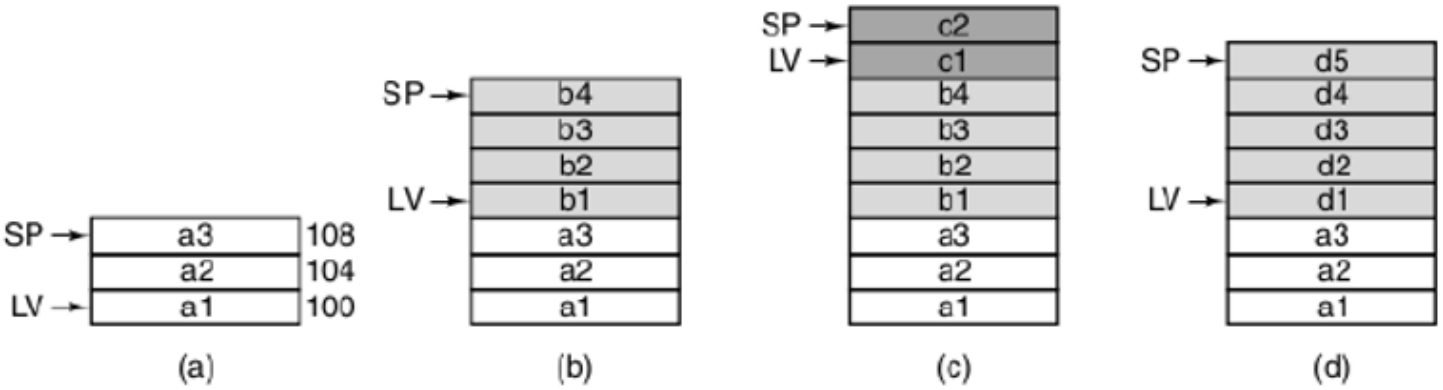
Многоуровневая архитектура

5. ЯВУ	• Компиляторы, Библиотеки
4. Язык ассемблера	• Ассемблер, Линкер (компоновщик), Отладчик
3. Уровень ОС	• Этот уровень и ниже – системное программирование
2. Машинный код (Instruction Set Arch, ISA)	•ОЗУ, Системная шина, ЦП
1. Микрокод процессора (микроархитектура)	• Внутренняя шина, Тракт данных, АЛУ
0. Схемы цифровой логики	• Логические вентили и схемы
-1. Уровень физических устройств	• Сфера электронной техники и радиофизики

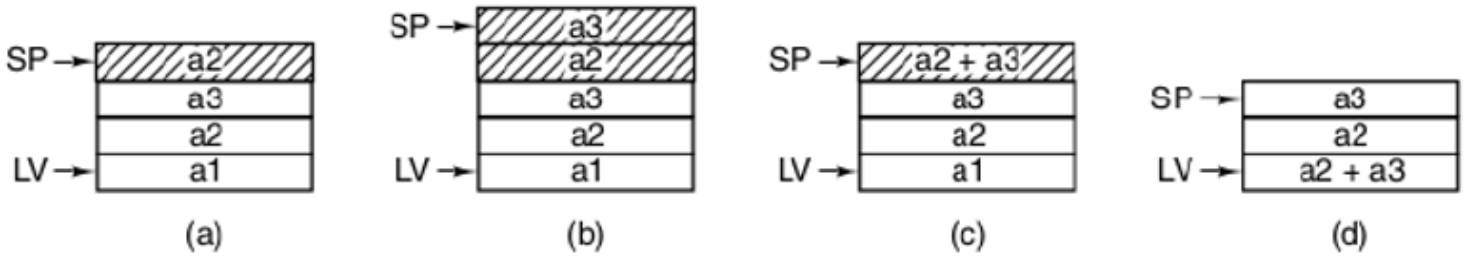
ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Стековые машины

Стек кадров подпрограмм:



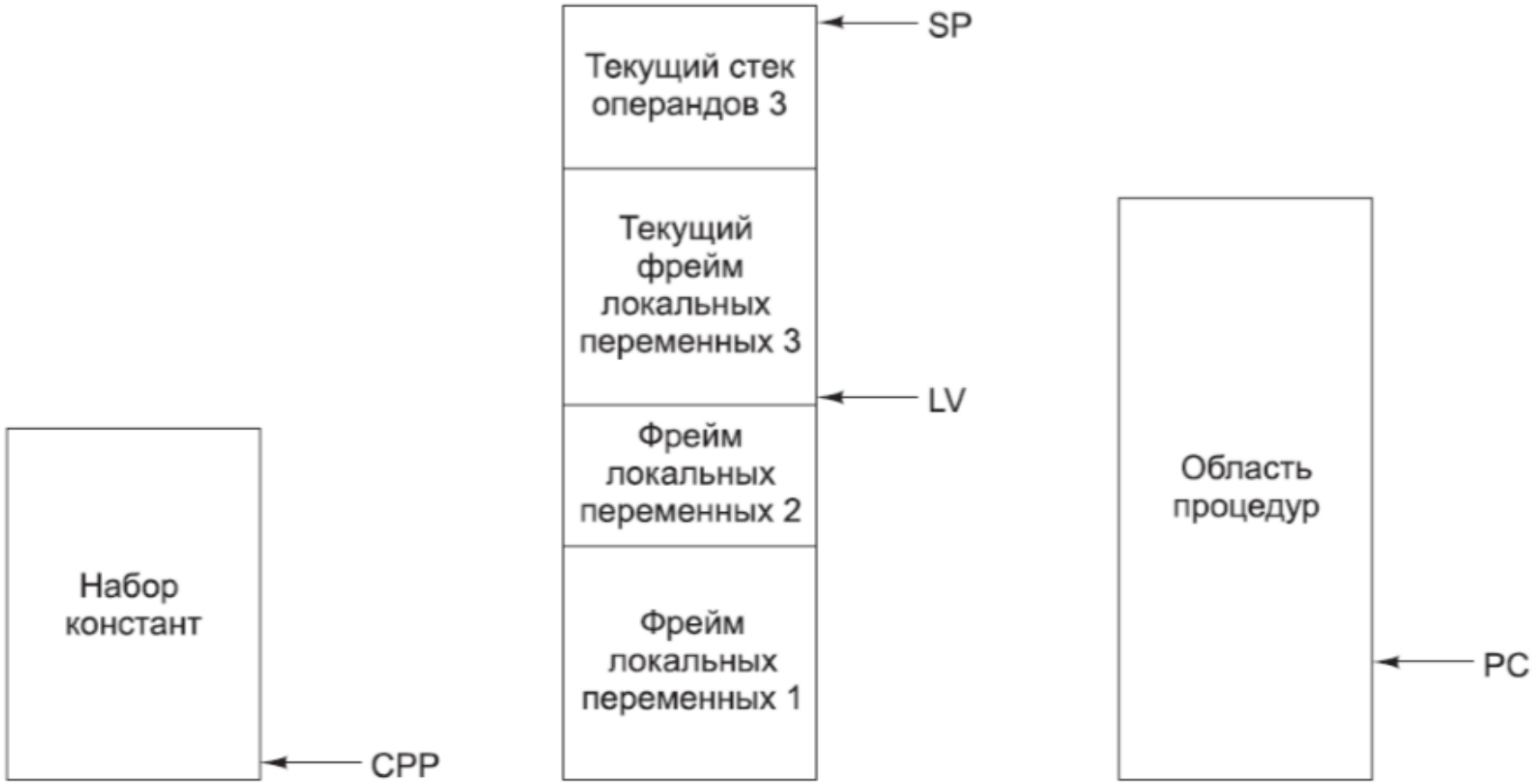
Стек операндов, пример: a1 = a2 + a3:

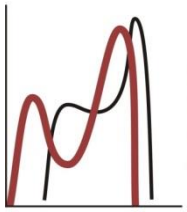


Стековые машины vs. регистровые машины.

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Модель памяти IJVM





ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Набор инструкций IJVM-1

Таблица 4.2. Набор IJVM-команд. Размер операндов byte, const и varnum — 1 байт. Размер операндов disp, index и offset — 2 байта

Hex	Мнемоника	Примечание
0x10	BIPUSH *byte*	Помещает байт в стек
0x59	DUP	Копирует верхнее слово стека и помещает его в стек
0xA7	GOTO *offset*	Безусловный переход
0x60	IADD	Выталкивает два слова из стека; помещает в стек их сумму
0x7E	IAND	Выталкивает два слова из стека; помещает в стек результат логического умножения (операция И)
0x99	IFEQ *offset*	Выталкивает слово из стека и совершает переход, если оно равно нулю
0x9B	IFLT *offset*	Выталкивает слово из стека и совершает переход, если оно меньше нуля
0x9F	IF_ICMPEQ *offset*	Выталкивает два слова из стека; совершает переход, если они равны
0x84	IINC *varnum const*	Прибавляет константу к локальной переменной
0x15	ILOAD *varnum*	Помещает локальную переменную в стек
0xB6	INVOKEVIRTUAL *disp*	Вызывает процедуру
0x80	IOR	Выталкивает два слова из стека; помещает в стек результат логического сложения (операция ИЛИ)

Компиляция в IJVM

Листинг 4.2. Программа для IJVM на языке ассемблера Java

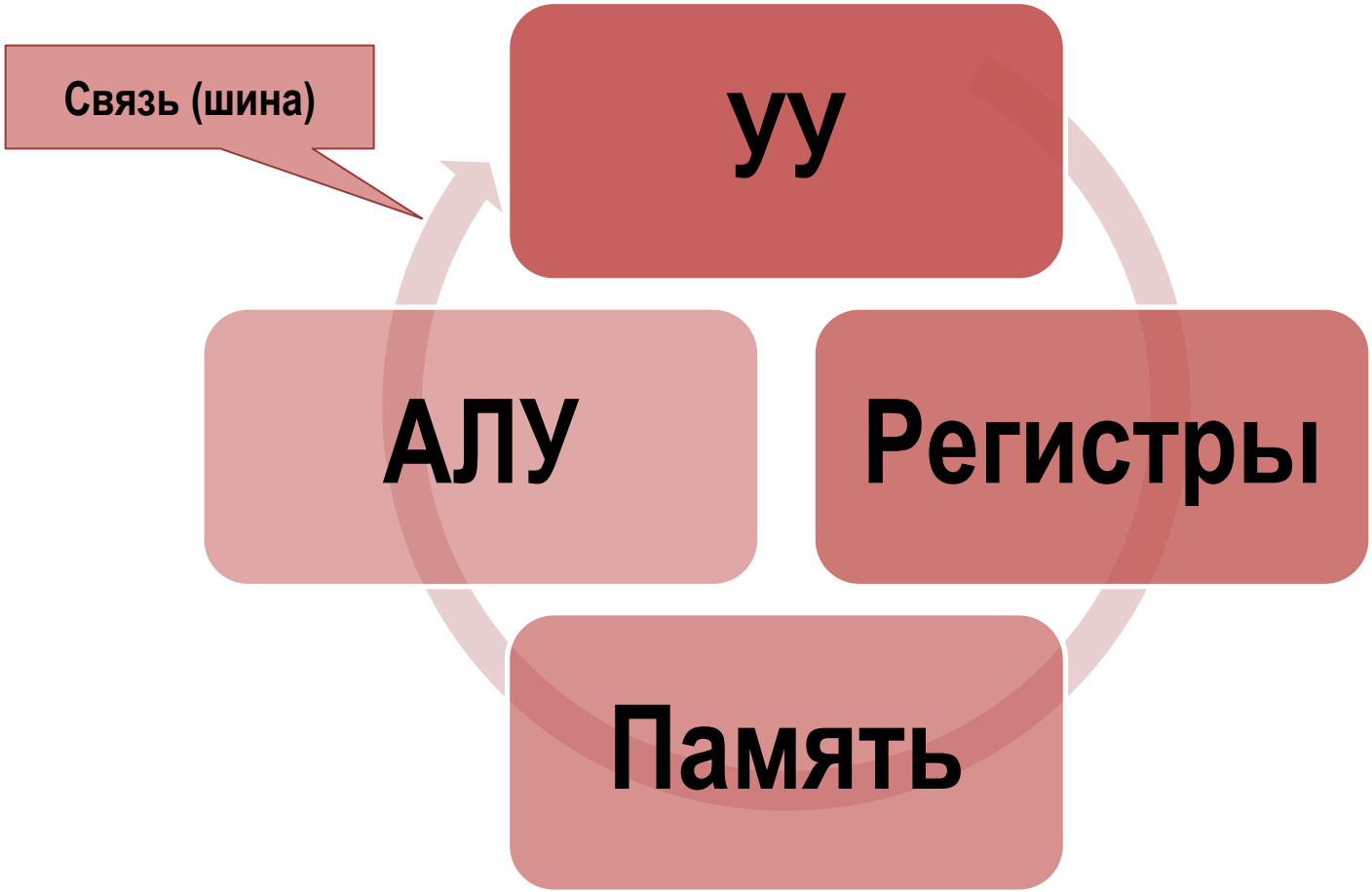
```
 1        ILOAD j          // i=j+k
 2        ILOAD k
 3        IADD
 4        ISTORE i
 5        ILOAD i          // if(i==3)
 6        BIPUSH 3
 7        IF_ICMPEQ L1
 8        ILOAD j          // j=j-1
 9        BIPUSH 1
10        ISUB
11        ISTORE j
12        GOTO L2
13   L1:  BIPUSH 0         // k=0
14        ISTORE k
15   L2:
```

```
i=j+k;
if(I==3)
     k=0;
else
     j=j-1;
```

```
0x15 0x02
0x15 0x03
0x60
0x36 0x01
0x15 0x01
0x10 0x03
0x9F 0x00 0x0D
0x15 0x02
0x10 0x01
0x64
0x36 0x02
0xA7 0x00 0x07
0x10 0x00
0x36 0x03
```

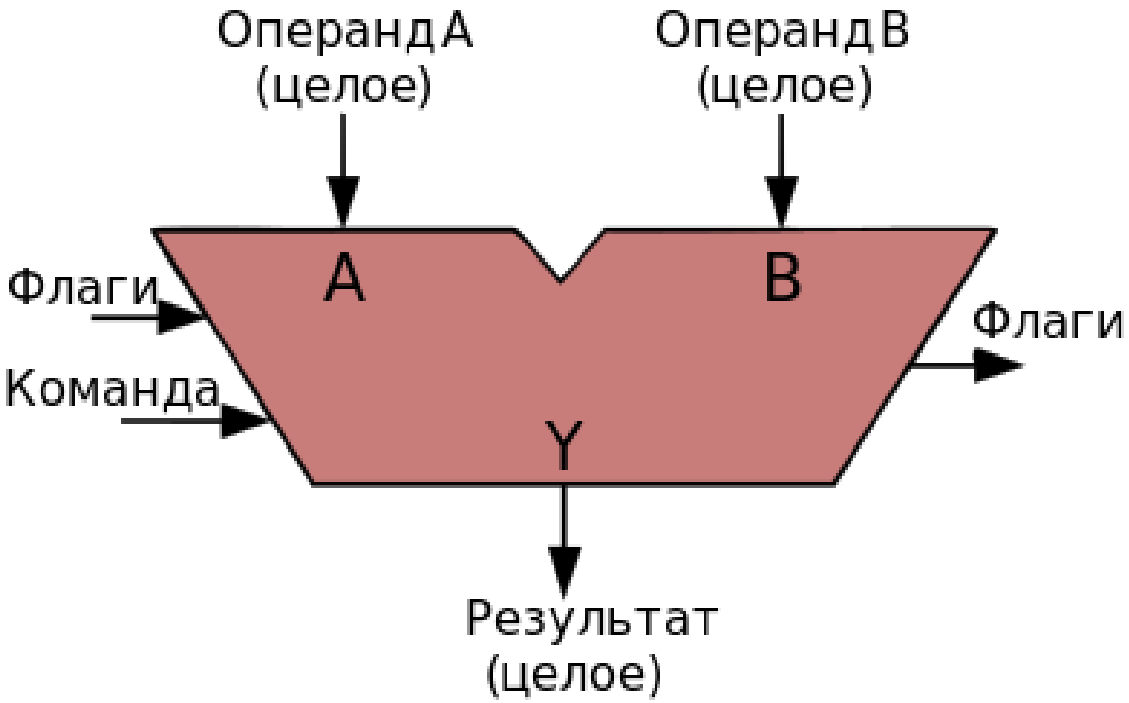
ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Схема процессора



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Арифметико-логическое устройство (АЛУ)

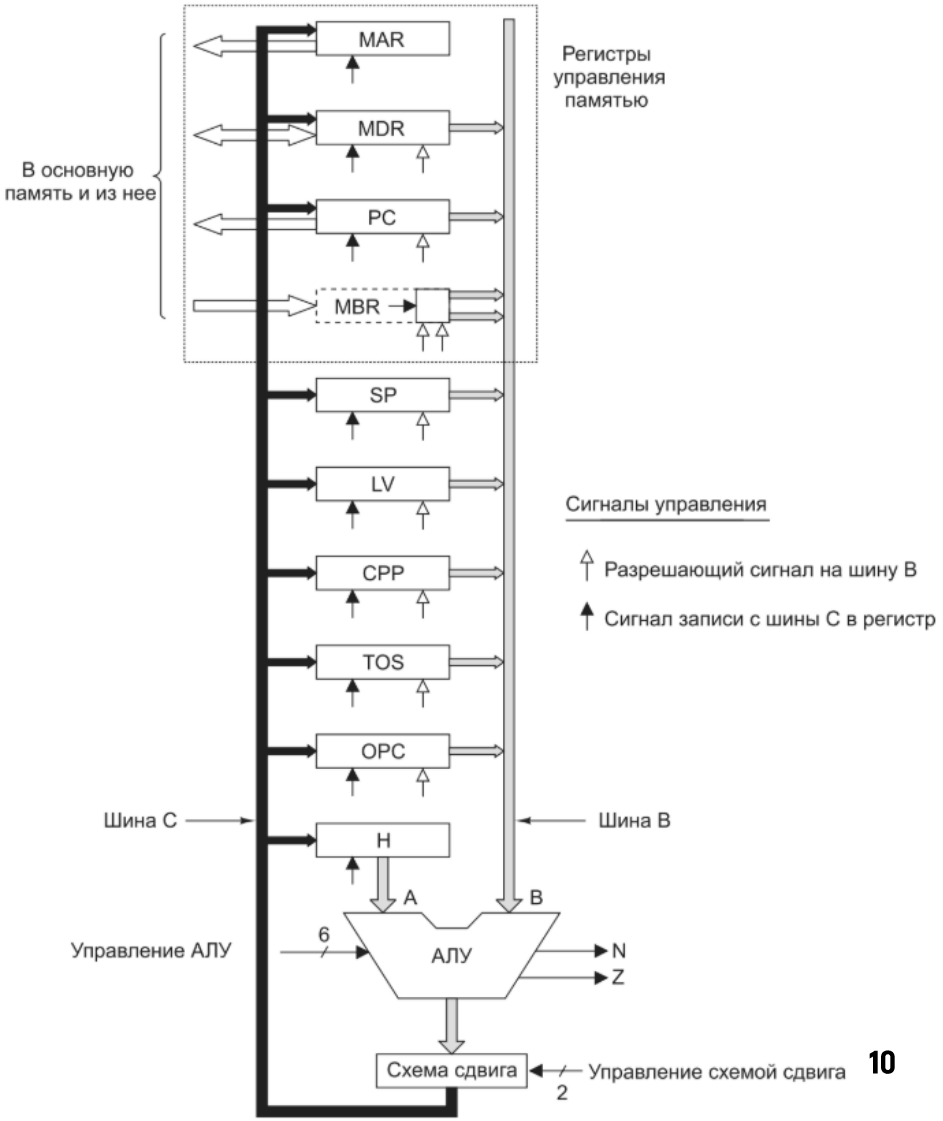


- Унарные операции: NOT, взятие противоположного, INC, DEC, тождество…
- Бинарные операции: сложение, вычитание, AND, OR…

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Тракт данных Mic-1

1. Устанавливаются сигналы управления.
2. Значения регистров загружаются на шину B.
3. Действуют АЛУ и схемы сдвига.
4. Результаты проходят по шине C обратно к регистрам.



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Регистры и сигналы

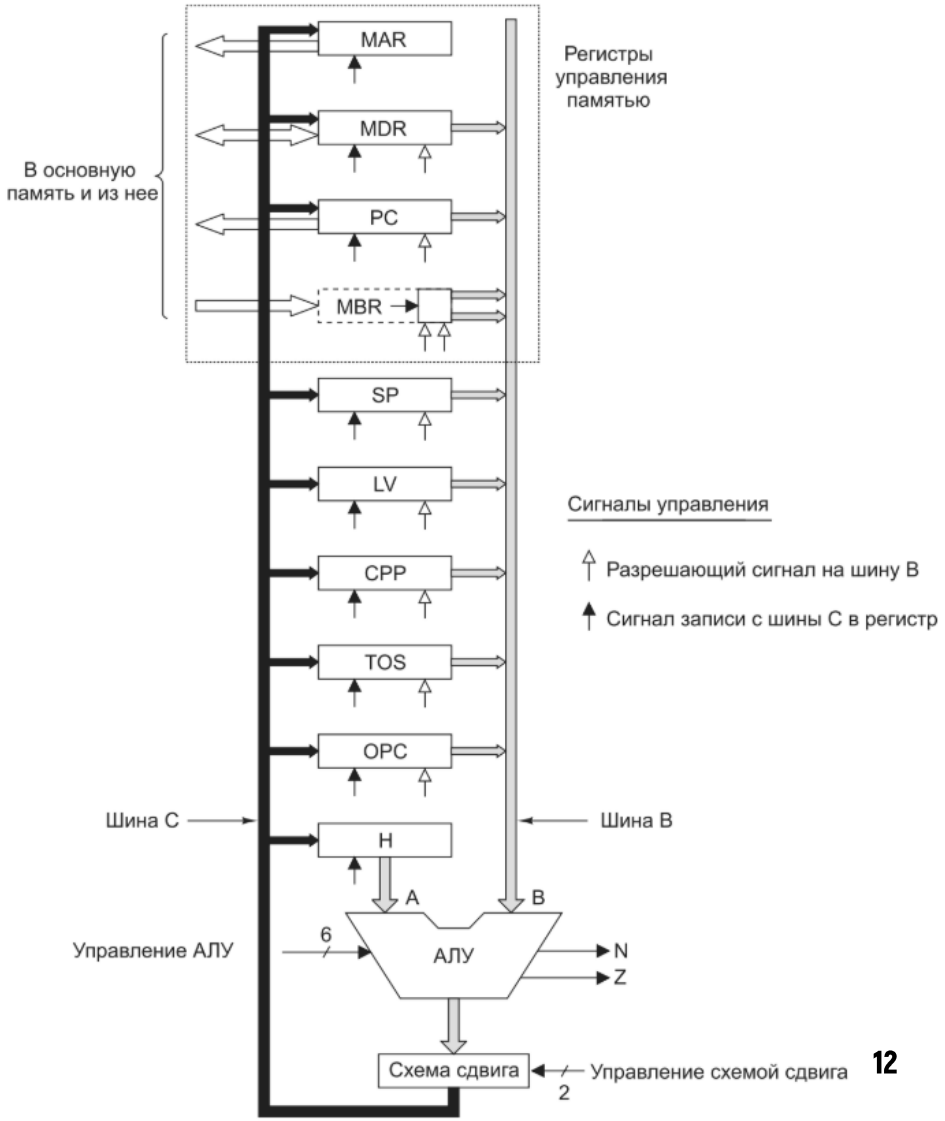
- MAR (Memory Address Register) — word-oriented, 32-bit, используется для rd или wr.
- MDR (Memory Data Register) — 32-bit.
- PC (Program Counter) — адрес следующей инструкции или её операнда, 32-bit.
- MBR (Memory Buffer Register) — значение следующей инструкции или первого байта операнда текущей инструкции, 8-bit.
- SP — адрес вершины стека в памяти
- TOS — значение вершины стека
- H —левый операнд АЛУ для бинарной операции
- Управляющие сигналы: rd, wr, fetch.

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Управляющие сигналы

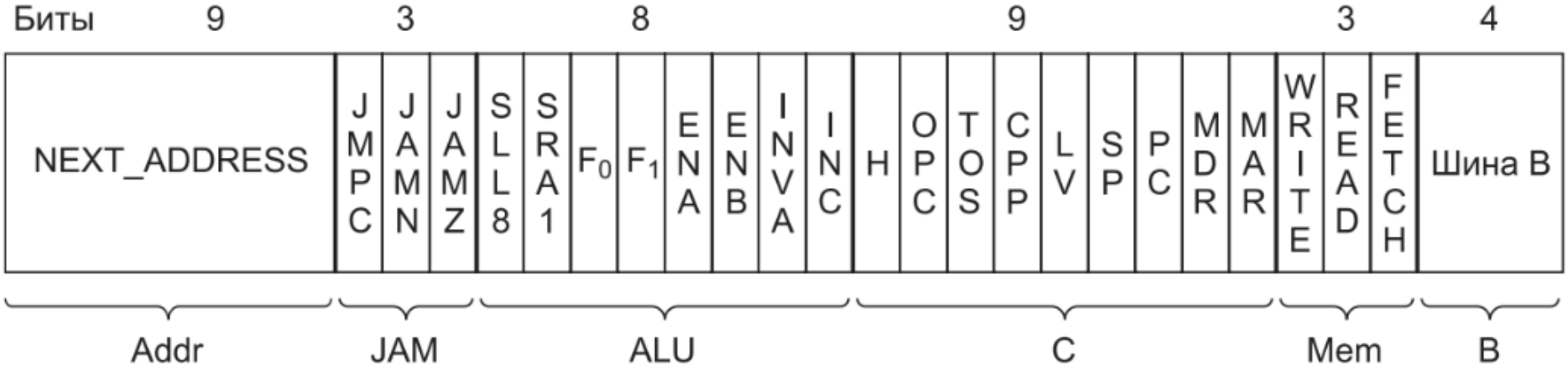
Можно уменьшить до 4-х битов

- 9 сигналов – регистр на шину B
- 9 сигналов – с шины C на регистры
- 8 сигналов - для АЛУ и сдвига
- 2 сигнала – запись или чтение из памяти
- 1 сигнал – необходимость fetch



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Формат микрокоманды



- **Addr** — адрес следующей потенциальной микрокоманды;
- **JAM** — определение того, как выбирается следующая микрокоманда;
- **ALU** — функции АЛУ и схемы сдвига;
- **C** — выбор регистров, которые записываются с шины C;
- **Mem** — функции памяти;
- **B** — выбор источника для шины B (закодированный номер)

Регистры шины B

0 — MDR	5 — LV
1 — PC	6 — CPP
2 — MBR	7 — TOS
3 — MBRU	8 — OPC
4 — SP	9–15 — нет

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Определение адреса следующей микрокоманды

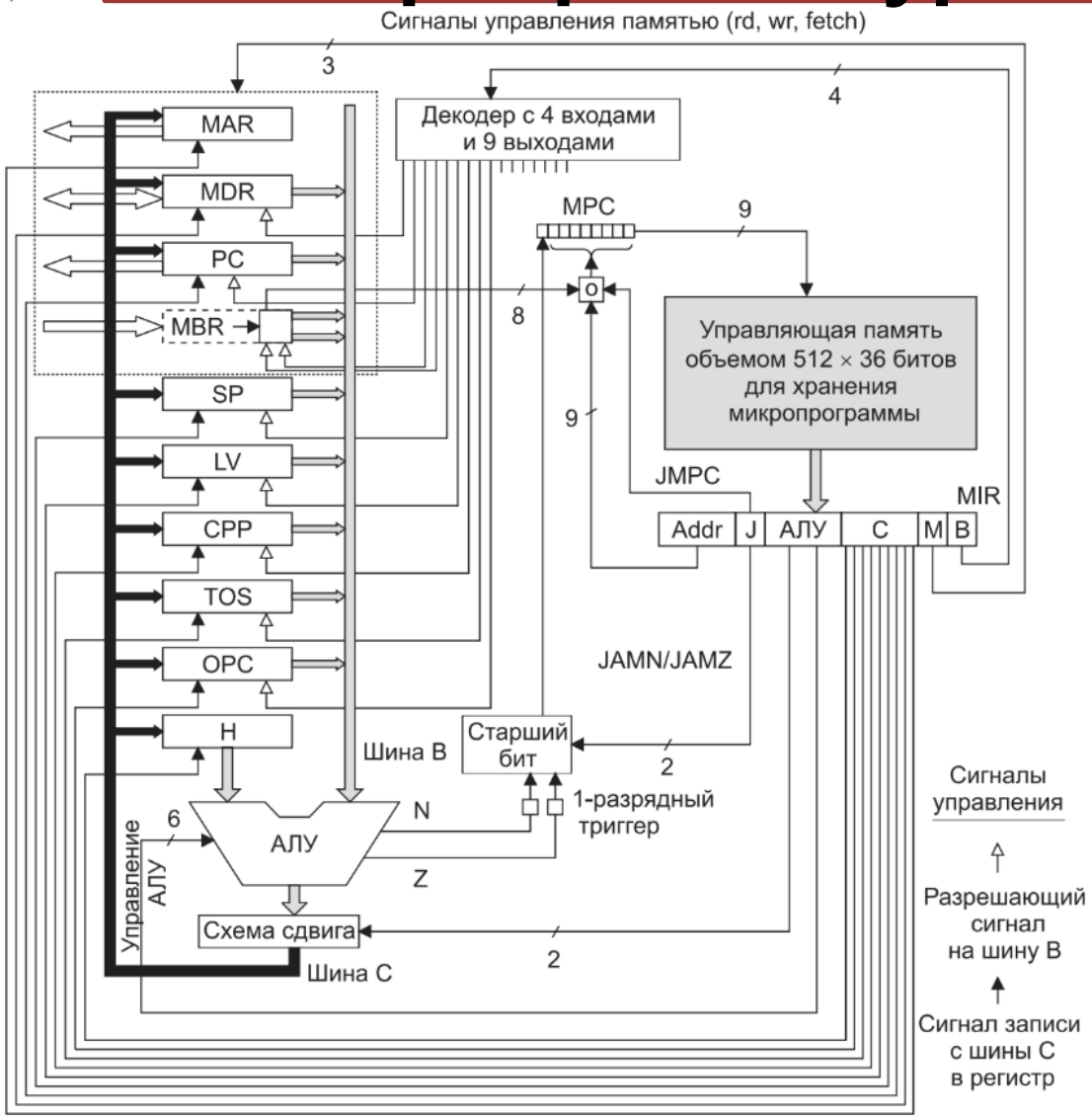
1. Значение NEXT_ADDRESS.
2. Значение NEXT_ADDRESS со старшим битом, соединенным операцией ИЛИ с логической единицей

   MPC[8]=((JAMZ И Z) ИЛИ (JAMN И N)) ИЛИ NEXT_ADDRESS [8]
3. MBR поразрядно связываются операцией ИЛИ с 8 младшими битами поля NEXT_ADDRESS текущей микрокоманды

   MPC[0..1]= JMPC ? (MBR ИЛИ NEXT_ADDRESS[0..7]) : NEXT_ADDRESS [0..7]

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Полная диаграмма микроархитектуры Mic-1



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Микроассемблер MAC

- MDR = SP + MDR - нельзя использовать!
- MAR = SP; rd - не успеет выполниться!
  MDR = H
- goto метка
- Z = TOS; if(Z) goto L1; else goto L2

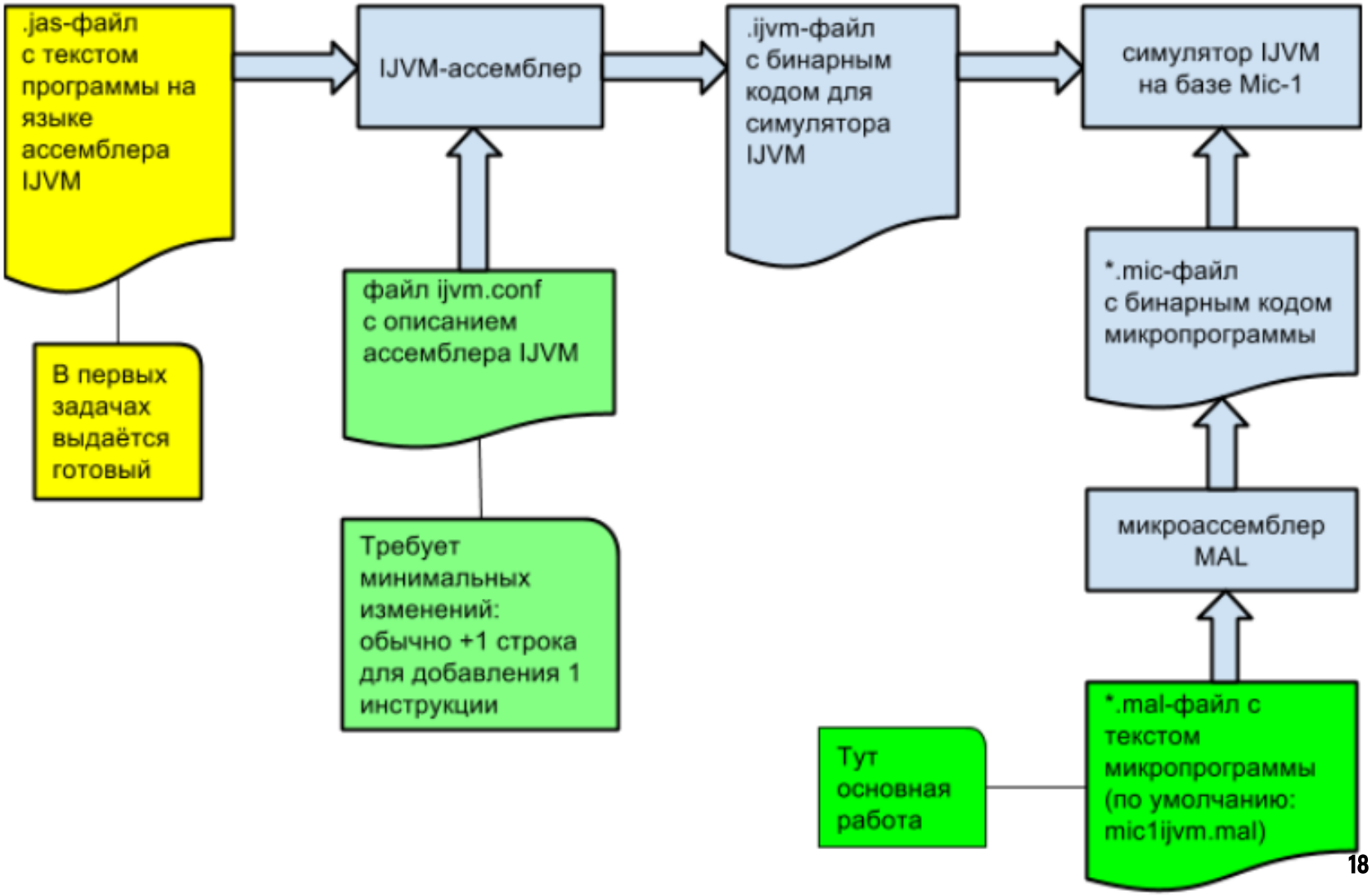
```
Main1   PC = PC + 1; fetch; goto (MBR)  // MBR holds opcode; get next byte; dispatch

nop1    goto Main1                      // Do nothing

iadd1   MAR = SP = SP - 1; rd           // Read in next-to-top word on stack
iadd2   H = TOS                         // H = top of stack
iadd3   MDR = TOS = MDR + H; wr; goto Main1     // Add top two words; write to top of stack

isub1   MAR = SP = SP - 1; rd           // Read in next-to-top word on stack
isub2   H = TOS                         // H = top of stack
isub3   MDR = TOS = MDR - H; wr; goto Main1     // Do subtraction; write to top of stack

iand1   MAR = SP = SP - 1; rd           // Read in next-to-top word on stack
iand2   H = TOS                         // H = top of stack
iand3   MDR = TOS = MDR AND H; wr; goto Main1   // Do AND; write to new top of stack
```

Микропрограмма для Mic-1

Микро-команда	Операции	Комментарий
Main1	PC = PC + 1; fetch; goto(MBR)	MBR содержит код операции; получение следующего байта; отсылка
nop1	goto Main1	Ничего не происходит
iadd1	MAR = SP = SP — 1; rd	Чтение слова, идущего после верхнего слова стека
iadd2	H = TOS	H = вершина стека
iadd3	MDR = TOS = MDR + H; wr; goto Main1	Суммирование двух верхних слов; запись суммы в верхнюю позицию стека

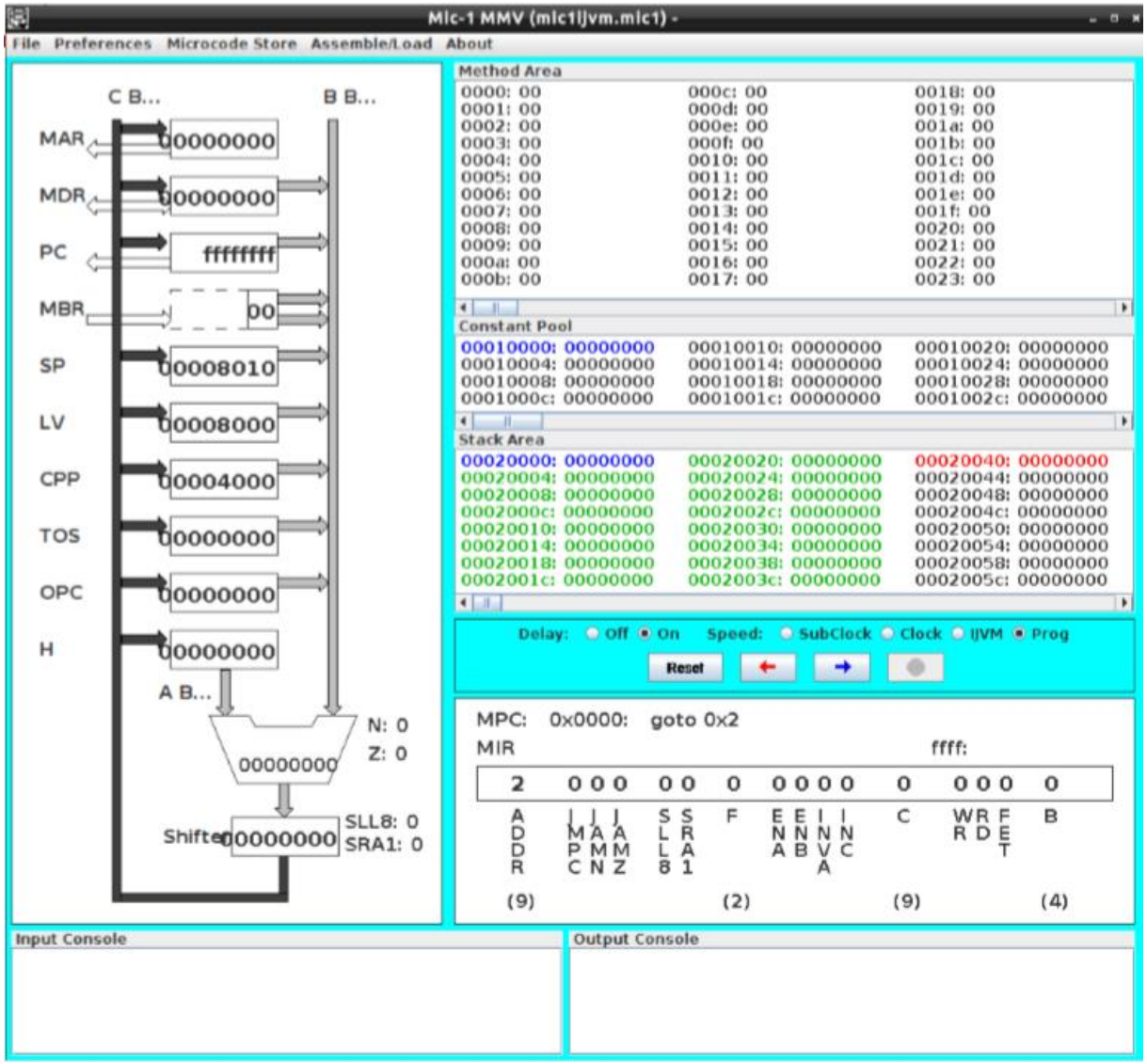
ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Лабораторная работа №7



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Лабораторная работа №7



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Домашнее задание

- Подготовка к тесту по лекции
  - раздел книги Таненбаума и Остина, пример архитектуры набора команд — IJVM (глава 4)
- Подготовка к лабораторному занятию 7