



Министерство науки и высшего образования Российской Федерации
Мытищинский филиал
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Космический

КАФЕДРА «Прикладная математика, информатика и вычислительная техника» КЗ-МФ

Лабораторная работа №2

ПО ДИСЦИПЛИНЕ:

Организация ЭВМ и систем

Студент КЗ-63Б
(Группа)

(Подпись, дата)

Чернов Владислав Дмитриевич
(И.О.Фамилия)

Студент КЗ-63Б
(Группа)

(Подпись, дата)

Братов Аким Романович
(И.О.Фамилия)

Преподаватель

(Подпись, дата)

Ефремов Николай Владимирович
(И.О.Фамилия)

2025 г.

Цель работы

Изучение архитектуры системы команд (АСК) процессора NIOS II, приобретение навыков создания новых программных компонентов и их отладки.

Приложенные материалы

Форматы команд

Машинные команды кодируются 32 разрядными словами. В программе на ассемблере могут использоваться псевдокоманды, которые после трансляции программы будут заменены одной или двумя машинными командами.

В процессоре Nios II используется 3 формата команд. [12,3]

I-тип. Поля A и B, размером 5 бит, используются для указания регистров. Для непосредственных операндов используется поле IMMED16, которое в АЛУ будет расширено до 32 бит.

R-тип. Поля A, B и C, размером 5 бит, используются для указания регистров. Поле OPX используется для расширения кода операций.

J-тип. Данный формат используется для инструкций вызова подпрограмм Call. Поле IMMED26, дополненное справа двумя нулевыми битами и слева четырьмя старшими разрядами счетчика команд PC, используется для задания прямого адреса вызываемой процедуры.

I – типа;

31	27	26	22	21	6	5	0
A					IMMED 16		КОП

R – типа;

31	27	26	22	21	17	16	6	5	0
A			B		C		OPX		КОП

J – типа.

31	6	5	0
IMMED 26			КОП

Адресация в процессоре Nios II

Процессор Nios II использует для адресации 32-битный адрес, при этом память является адресуемой по байтам. С помощью команд можно записывать/считывать слова (32), полуслова (16) и байты данных(8). Чтение или запись данных по адресам, которые не связаны с памятью или периферийными устройствами, приводит к неопределенным результатам. В процессоре NIOS II используются следующие способы адресации.

- **Непосредственная адресация.** 16 битный операнд присутствует в самой команде. Он будет дополнен знаковыми разрядами до 32 разрядов при выполнении арифметической операции над 32 разрядными словами.

- **Регистровая адресация.** Операнды находятся в регистрах процессора.

- **Относительная регистровая адресация.** Эффективный адрес операнда получается путем суммирования содержимого регистра и знакового 16 разрядного смещения, находящегося в самой команде.

- **Косвенная регистровая адресация.** Содержимое регистра является эффективным адресом операнда. Этот способ эквивалентен предыдущему способу, когда смещение в команде равно нулю.

- **Абсолютная адресация.** 16-битный абсолютный адрес операнда может быть определен путем использования смещения относительно регистра **r0**, который всегда равен нулю.

Листинг 2.

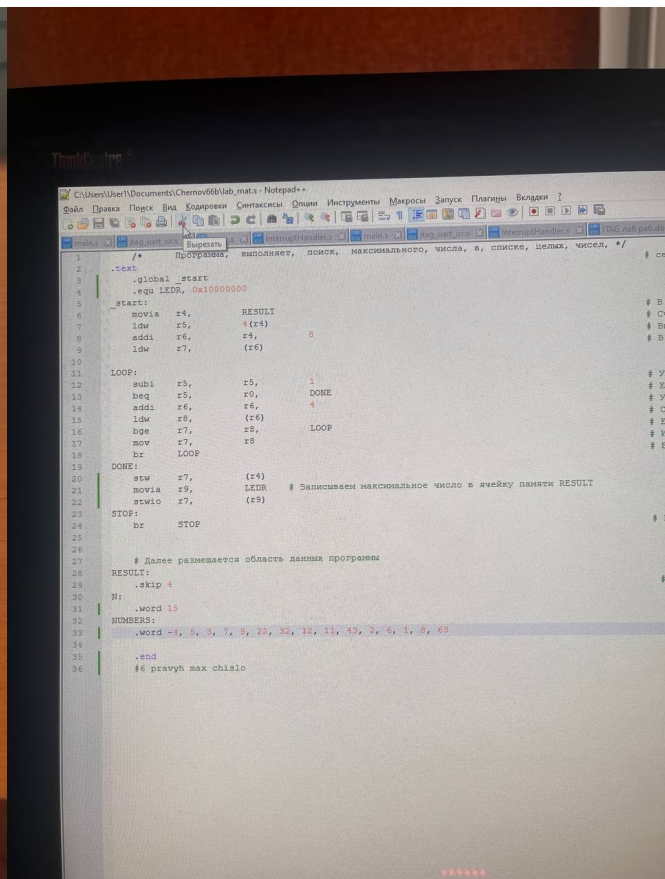
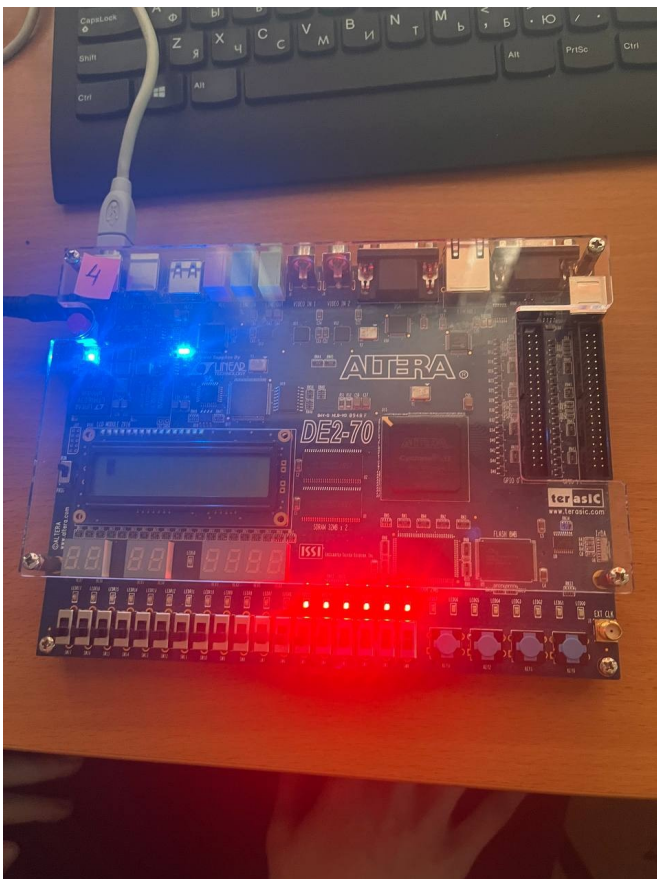
Исходный код программы lab_max.s, нахождения наибольшего числа из списка

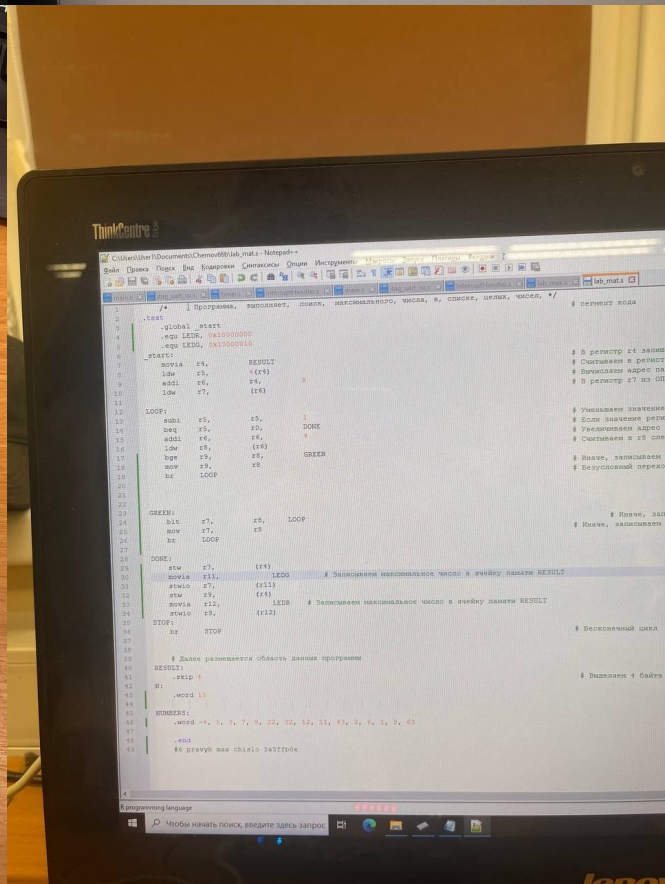
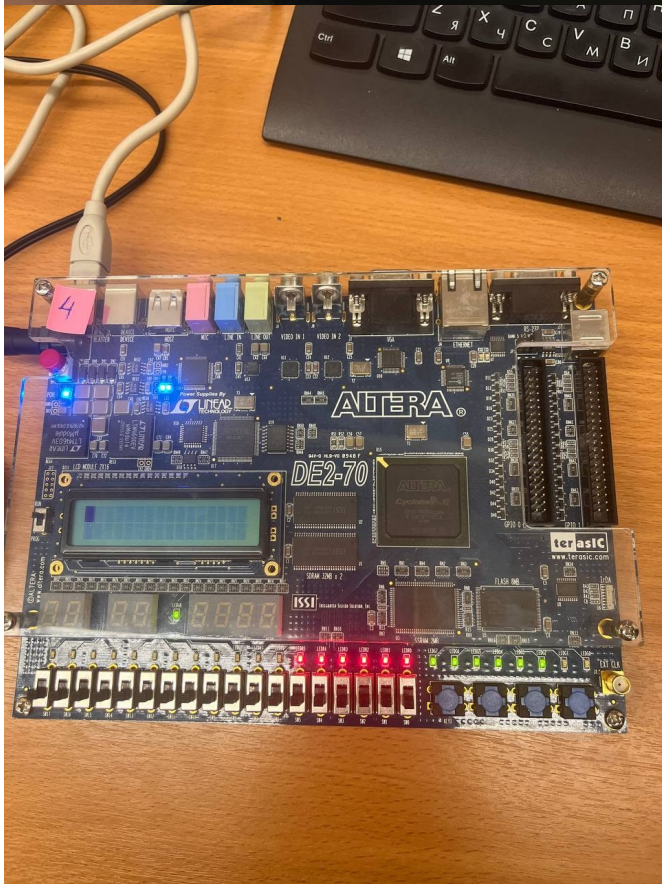
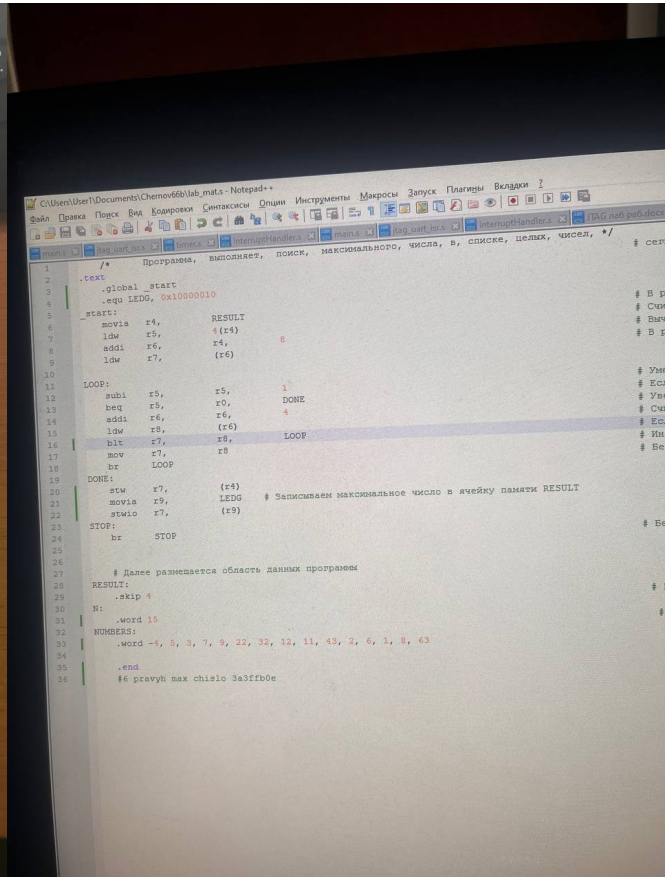
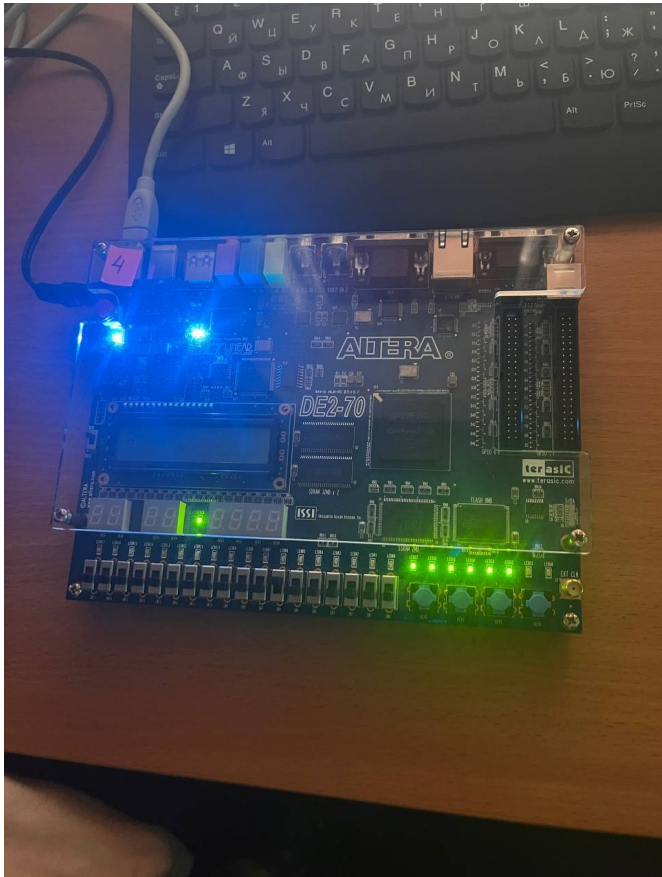
```
/* Программа выполняет поиск максимального числа в списке целых чисел */
.text
.global _start
_start:
    movia r4, RESULT # В регистр r4 запишем адрес ОП, куда поместим результат
    ldw r5, 4(r4) # Считываем в регистр r5 значение N - количество чисел в списке
    addi r6, r4, 8 # Вычисляем адрес памяти, с которого начинают располагаться числа для поиска и записываем его в r6
    ldw r7, (r6) # В регистр r7 из ОП считываем первое число из списка

LOOP:
    subi r5, r5, 1 # Уменьшаем значение количества чисел в списке
    beq r5, r0, DONE # Если значение регистра r5 равно 0, то выходим из цикла
    addi r6, r6, 4 # Увеличиваем адрес памяти на 4 для перехода к следующему числу в списке
    ldw r8, (r6) # Считываем в r8 следующее число из списка
    bge r7, r8, LOOP # Если найденное максимальное число больше или равно считанному, то возвращаемся в начало цикла
    mov r7, r8 # Иначе, записываем в r7 новое максимальное число
    br LOOP # Безусловный переход в начало цикла
DONE:
    stw r7, (r4) # Записываем максимальное число в ячейку памяти RESULT

STOP:
    br STOP # Бесконечный цикл
/* Далее размещается область данных программы */
RESULT:
    .skip 4 # Выделяем 4 байта для записи результата
N:
    .word 7 # Количество чисел в списке
NUMBERS:
    .word 4, 5, 3, 6, 1, 8, 2 # Числа из списка
.end
```

Выполнение заданий лабораторной работы





Заключение

В ходе выполнения лабораторной работы №2, студенты обрели следующие навыки :

1. Создавать новые программные компоненты для процессорной системы и работать с ними;
2. Находить в ОП исполняемый код программы и отдельные её команды, исходные данные и их отдельные элементы;
3. Отлаживать программные компоненты, используя контрольные точки, выполнение по шагам и в автоматическом режиме