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» К3-МФ

Отчет к Лабораторной работе №1

ПО ДИСЦИПЛИНЕ:

Организация ЭВМ и систем

Студент __К3-66Б__ _____ Чернов Владислав Дмитриевич
(Группа) (Подпись, дата) (И.О.Фамилия)

Студент __К3-66Б__ _____ Братов Аким Романович
(Группа) (Подпись, дата) (И.О.Фамилия)

Преподаватель _____ Ефремов Николай Владимирович
(Подпись, дата) (И.О.Фамилия)

2025 г.

Цель работы

Приобретение навыков, необходимых для выполнения лабораторных работ по дисциплине «Организация ЭВМ и систем» с использованием стендов Intel DE 2-115 с помощью приложения Intel Monitor Program (IMP). Умение использовать в своих проектах кнопочные и ползунковые переключатели, светодиоды и hex индикаторы, входящие в состав стенда.

Исходные файлы лабораторной работы

Файлы, которые используются в этой лабораторной работе, входят в состав приложения IMP. Для процессорной системы DE 2-115 Media Computer это следующие файлы: nios_system.sopcinfo и DE2-115_Media_Computer.sof. Первый из них содержит перечисление компонентов процессорной системы с указанием их особенностей и адресов. Второй файл является конфигурационным файлом, то есть используется для прошивки кристалла ПЛИС стенда. Третий файл test_Media_Computer.s содержит программу тестирования процессорной системы. Листинг основной части программы с подробными комментариями:

```
.include "address_map.s"

.equ RIBBON_CABLE_INSTALLED, 0

/* Программа демонстрирует различные возможности процессорной системы.

Она выполняет следующие действия:

 1. Тестирует статическую память.

 2. Прокручивает текст на 7-сегментном дисплее. Если ошибок при тестировании статической

памяти не обнаружено, то текст содержит слова "dE2" и "PASSEd". Если были обнаружены

ошибки, то выводится слово "Error".

 3. Мигает зелеными светодиодами. Скорость мигания светодиодов и прокрутки текста на 7

сегментных индикаторах регулируется прерываниями таймера.

 4. Подключает переключатели к красным светодиодам.

 5. Обрабатывает прерывания от кнопок. Нажатие кнопки KEY1 увеличивает скорость

прокрутки текста. Нажатие кнопки KEY2 снижает скорость, кнопки KEY3 - останавливает

прокрутку.

 6. Тестирует порты расширения JP1, JP2.

 7.Отсылает обратно данные, полученные по интерфейсу JTAG UART (символы, введенные в

терминальном окне программы Intel Monitor Program) и наоборот.

*/

    .text
```

```
    .global _start

_start:

    /* инициализируем регистры sp и fp */

    movia   sp, 0x07FFFFC         # Стек начинается с адреса последнего слова в SDRAM памяти

    mov     fp, sp

    /* инициализируем буфер 7-сегментных индикаторов */

    movia   r16, DISPLAY_BUFFER

    movi    r17, 0xde2

    stw     r17, 0(r16)

    stw     zero, 4(r16)

    stw     zero, 8(r16)

    /* инициализируем зеленые светодиоды */

    movia   r2, 0x55555555

    movia   r16, GREEN_LED_PATTERN

    stw     r2, 0(r16)

    /* инициализируем счетчик задержки, используемый для определения изменений

отображаемого текста */

    movia   r16, EIGHT_SEC

    stw     zero, 0(r16)

    /* инициализируем переключатели */

    movia   r16, DISPLAY_TOGGLE

    stw     zero, 0(r16)

    /* направление передачи может быть сохранено в SHIFT_DIRECTION, где 0-влево, 1-вправо

*/

    movi    r2, 1

    movia   r16, SHIFT_DIRECTION

    stw     r2, 0(r16)

    /* запускаем таймер и разрешаем его прерывания */

    movia   r16, INTERVAL_TIMER_BASE

    movi    r15, 0b0111     # START = 1, CONT = 1, ITO = 1

    sthio   r15, 4(r16)

    /* разрешаем прерывания от кнопок KEY3, KEY2, KEY1*/
```

```
    movia   r16, PUSHBUTTON_BASE
    movi    r15, 0b01110   # устанавливаем биты маски прерывания в 1
    stwio   r15, 8(r16)  # заносим в регистр маски
    /* разрешаем прерывания процессора от кнопок и таймера */
    movi        r15, 0b011
.if RIBBON_CABLE_INSTALLED
    ori     r15, r15, 0b1000000000000   # также разрешаем прерывания для порта расширения JP2
.endif
    wrctl   ienable, r15
    movi    r15, 1 # разрешаем прерывания текущей программы
    wrctl   status, r15
    /* цикл, в котором тестируется статическая память и обновляются 7-сегментные индикаторы
*/
    movia   r15, 0x55555555
    movia   r17, SRAM_END
 DO_DISPLAY:
    movia   r16, SRAM_BASE
    movia   r17, SRAM_END
 MEM_LOOP:
    call    UPDATE_HEX_DISPLAY
    call    UPDATE_RED_LED
    call    UPDATE_UARTS
    /* Тестирование только для порта расширения JP2 */
 .if RIBBON_CABLE_INSTALLED
    call        TEST_EXPANSION_PORTS        # возвращает 0 если тест не пройден
    beq     r2, zero, SHOW_ERROR
.endif
    stw     r15, 0(r16)
    ldw     r14, 0(r16)
    bne     r14, r15, SHOW_ERROR
    addi    r16, r16, 4
    ble     r16, r17, MEM_LOOP
```

```
    xori    r15, r15, 0xFFFF
    xorhi   r15, r15, 0xFFFF
    /* меняет буфер 7-сегментных индикаторов приблизительно каждые 8 сек */
    movia   r16, EIGHT_SEC
    ldw     r17, 0(r16)
    movi    r14, 80
    ble     r17, r14, DO_DISPLAY
    stw     zero, 0(r16)
    /* toggle display of dE2 and PASSEd */
    movia   r16, DISPLAY_TOGGLE
    ldw     r17, 0(r16)
    beq     r17, zero, SHOW_PASSED
    stw     zero, 0(r16)
    /* показать "dE2" */
    movia   r16, DISPLAY_BUFFER
    movi    r17, 0xdE2
    stw     r17, 0(r16)
    stw     zero, 4(r16)
    stw     zero, 8(r16)
    br      DO_DISPLAY
    /* показать "Passed" */
SHOW_PASSED:
    movi        r17, 1
    stw     r17, 0(r16)
    movia   r16, DISPLAY_BUFFER
    movia   r17, 0xbA55Ed
    stw     r17, 0(r16)
    stw     zero, 4(r16)
    stw     zero, 8(r16)
    br      DO_DISPLAY
    /* показать "Error" */
```

```
SHOW_ERROR:
  movia   r16, DISPLAY_BUFFER
  movia   r17, 0xe7787
  stw     r17, 0(r16)
  stw     zero, 4(r16)
  stw     zero, 8(r16)
DO_ERROR:
  call    UPDATE_HEX_DISPLAY
  br      DO_ERROR
```

Файл address_map.s содержит определения базовых адресов периферийных устройств и памяти.

```
.equ SRAM_BASE, 0x8000000
.equ SRAM_END, 0x81FFFFF
.equ RED_LED_BASE, 0x10000000
.equ GREEN_LED_BASE, 0x10000010
.equ HEX3_HEX0_BASE, 0x10000020
.equ HEX7_HEX4_BASE, 0x10000030
.equ SLIDER_SWITCH_BASE, 0x10000040
.equ PUSHBUTTON_BASE, 0x10000050
.equ JP1_EXPANSION_BASE, 0x10000060
.equ JP2_EXPANSION_BASE, 0x10000070
.equ JTAG_UART_BASE, 0x10001000
.equ UART_BASE, 0x10001010
.equ INTERVAL_TIMER_BASE, 0x10002000
```

ВЫПОЛНЕНИЕ ЗАДАНИЙ ЛАБОРАТОРНОЙ РАБОТЫ

1.Подключить стенд к инструментальному компьютеру и подать 1питание на стенд, нажав красную кнопку.

2. Создать новый проект в IMP.

3. Запустить программу Test Computer, которая:

Тестирует статическую память. Тестирование заключается в заполнении оперативной памяти значениями 0x55555555. Каждый цикл записи сопровождается считыванием записанной информации и сравнением с эталоном. Затем число-заполнитель меняется на инверсное значение, и цикл тестирования продолжается.

Отображает бегущую строку на 7-сегментном дисплее. Если ошибок при тестировании статической памяти не обнаружено, то строка содержит слова "dE2" и "PASSEd". Если обнаружены ошибки, то выводится слово "Error".

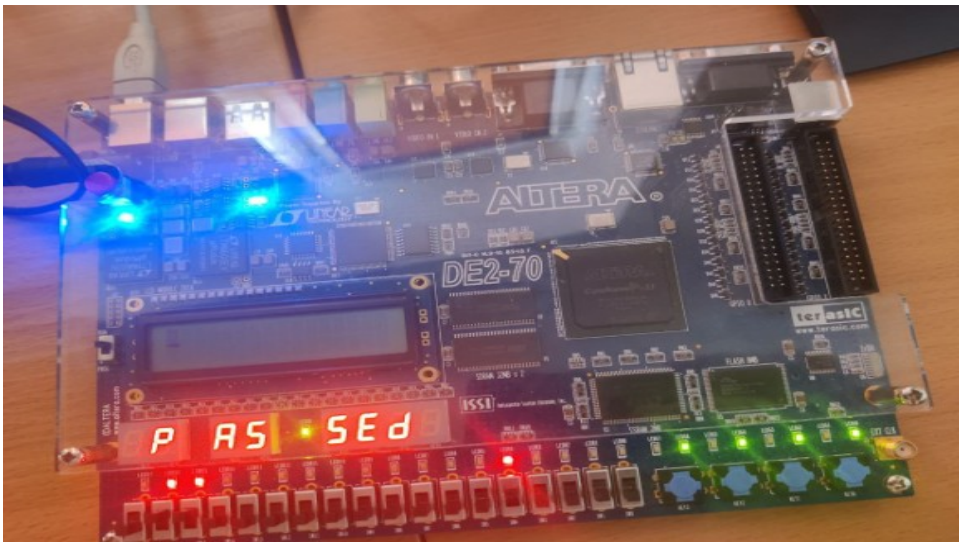
Включает мерцание зеленых светодиодов. Скорость мерцания светодиодов и прокрутки текста на 7-сегментных индикаторах регулируется прерываниями от таймера.

Подключает переключатели к красным светодиодам.

Обрабатывает прерывания от кнопок. Нажатие кнопки KEY1 увеличивает скорость прокрутки строки. Нажатие кнопки KEY2 снижает скорость, нажатие кнопки KEY3 - останавливает прокрутку.

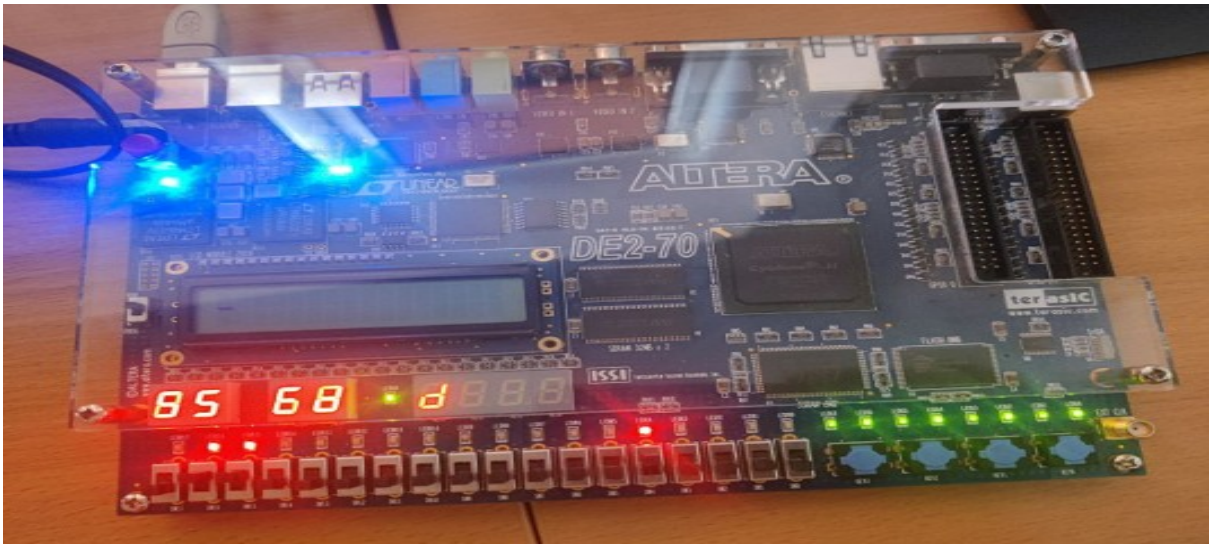
Тестирует порт расширения JP1, если установлены соответствующие перемычки.

Принимает данные, вводимые в терминальное окно IMP, и отсылает их обратно, используя интерфейс JTAG UART, и дополнительно пересылает их в com порт.



4. Экспериментально определить назначение кнопок KEY1, KEY2, KEY3.

KEY1 - увеличивает скорость прокрутки строки. KEY2 снижает скорость, KEY3 - останавливает прокрутку.

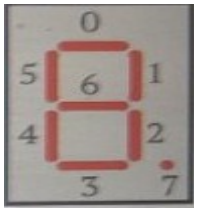


5. В терминальном окне IMP ввести своё имя и фамилию.

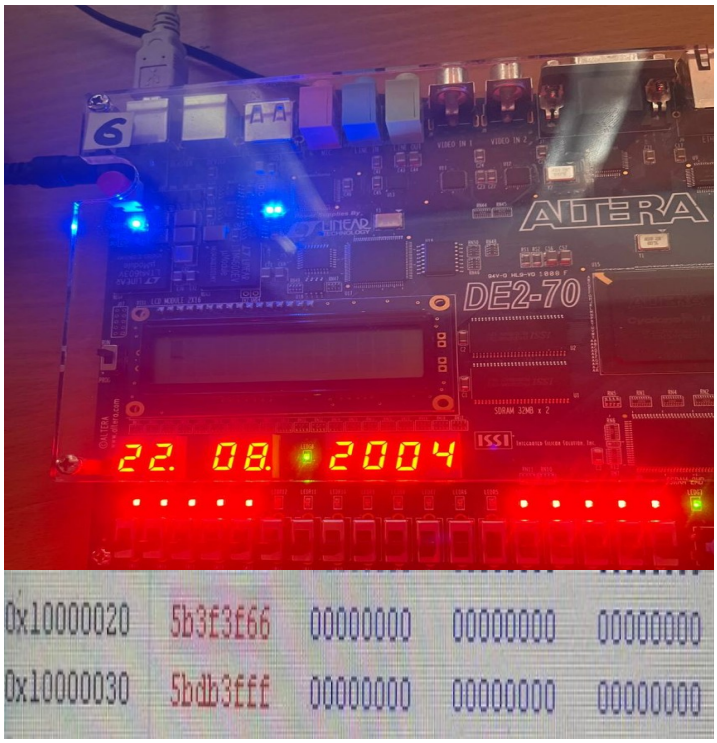
Из терминального окна текст отправляется на стенд, потом возвращается обратно и уже после этого печатается на экране.

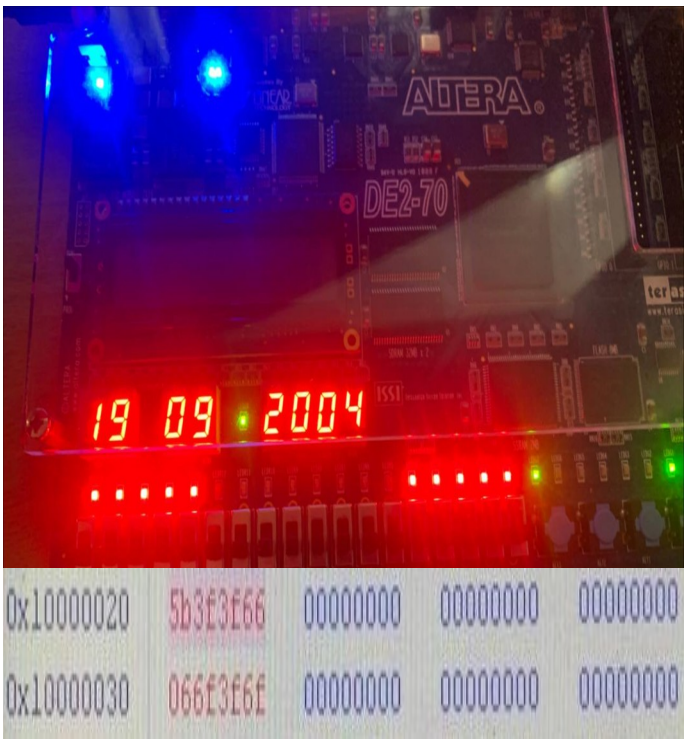
6. Управлять сегментами индикаторов шестнадцатеричной цифры, подавая различные наборы данных в соответствующие порты вывода данных.

Принцип работы индикатора следующий: если представить, что каждый сегмент соответствует своему разряду в двоичном числе, то ставя эти разряды в единицу, можно включить сегмент, в нуль — выключить.



7. Сформировать наборы данных таким образом, чтобы на HEX индикаторах высветилась дата нашего рождения в формате дд.мм.гггг.





Заключение

В ходе выполнения лабораторной работы 1, студенты обрели навык работы со стендом и приложением IMP для работы с ним, загрузки процессорной системы в кристалл ПЛИС учебного стенда, использовать Disassembly для управление программой и вкладкой Memory для управления периферийных состояний ввода(ползунковые и кнопочные переключатели) и управление устройствами вывода (светодиоды и индикаторы hex)