Лабораторная работа №2.

Исследование простой ассемблерной программы.

Цель работы: Изучение команд арифметических операций ARM-ассемблера. Работа с областью памяти данных, стеком и памяти подключения периферии.

Порядок выполнения работы.

Часть 1.

Прогр.1

```
.text
.global _start
_start: LDR R0, ADR1
        MOVW R1, # 655
        ADD R2, R0, R1
        SUB R3, R0, R1
        PUSH {R2, R3}
        LDR R4, =ADR2
        LDR R0, [R4]
        LDR R1, [R4, #4]
        MOV R3, #5
        MLA R2, R0, R1, R3
        MOV R5, R2
        POP {R3, R2}
STOP: B STOP
ADR1: .word 20
ADR2: .word 12, 10
.end
```

1. Ознакомиться с текстом программы – Прогр. 1.
2. На рабочем столе открыть редактор **GVim**: Меню приложений – Инструменты – Gvim.
3. Создать новый файл (Файл – Новый), войти в режим вставки (**Insert**).
4. Набрать текст Прогр.1.
5. Сохранить файл, создав любую папку проекта в своем каталоге, присвоив ему расширение s. Например: part2_1.s
6. Подключить макет. Открыть Altera Monitor Program (Файловая система – opt – altera15.0 – University Program – Monitor Program – bin – altera monitor program.)
7. Выбрать File > New Project.
8. Далее выбрать имя папки, определить имя проекта в соответствии с тем, как Вы его назвали. Выбрать архитектуру ARM Cortex-A9. Клик Next.
9. Выбрать DE1-SoC Computer, Next.
10. Выбираем Assembly Program. Окно вставок не инициализируем! Next.
11. В следующем окне добавляем созданный файл. Next.
12. Specify system parameters оставляем без изменений. Next.
13. Диапазоны используемой памяти также оставляем без изменения. Finish.
14. Осуществить загрузку кодов в макет, дав положительный ответ на запрос “Download System”.
15. Произвести компиляцию файла и загрузку (Actions> Compile&Load)*.
16. Перед началом выполнения программы просмотреть содержимое ячеек памяти, содержащих данные **Memory** (под окном программы), и записать адрес ячейки и данные в ней. Вернуться на **Disassembly**, записать содержимое sp.

    ADR1 - ……. **.word** 0x……

    ADR2 - ……. **.word** 0x……

    ……. **.word** 0x……

    SP = 0x……..
17. Выполнить программу пошагово, заполняя таблицу 1 после каждого шага.

Таблица 1

Номер выполненной команды	Состояние программного счетчика	Измененные состояния регистров
1.	pc = 0x00000004	R0 = 0x…….
2.		
3.		

Если при компиляции в отладчике появились сообщения об ошибках, внимательно прочтите их и исправьте исходный файл в редакторе GVim. Затем повторите путь до компиляции в отладчике.

Часть 2.

Изменить программу, добавив возможность вывода результата умножения на светодиоды, и состояния регистров, извлеченных из стека, в ячейки памяти. Для этого в GVim открыть файл с прогр.1, Insert. Полученную программу записать под новым именем в том же проекте.

В отладчике выполнить программу в пошаговом режиме, проверить состояние заданных в программе ячеек памяти для сохранения содержимого R2 и R3, заполнить таблицу 2. Продемонстрировать результат преподавателю. Получить задание на защиту лабораторной работы.

Таблица 2

Содержимое ячеек памяти (Данные 1 –до исполнения программы, Данные 2 – после выполнения программы).

Имя метки	Адрес метки	Адрес команды	Данные 1	Данные 2

Отчет должен содержать:

1. ***Текст программы Части 1 работы, таблица 1.***
2. ***Текст программы Части 2 работы таблица 2.***
3. ***Текст программы – защиты лабораторной работы.***