|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий

Кафедра Вычислительной Техники (ВТ)

**Отчёт**

по дисциплине

«Разработка программно-аппаратного обеспечения информационных и автоматизированных систем»

Выполнил студент группы ИКМО-05-23: Миронов Д. С.

Принял преподаватель: Данилович Е.С.

Работы выполнена «\_\_»\_\_\_\_\_20\_\_г

«Зачтено»

«\_\_»\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Москва 2023

**Задание**: Освоение маршрута проектирования процессорных ядер на уровне системного моделирования путем разработки модели процессорного ядра на языке программирования высокого уровня. Проект заключается в описании модели регистров и памяти процессора, а также проверки модели путем составления тестовой программы с предсказуемым результатом исполнения. После записи команд тестовой программы в массив, имитирующий память процессора, запуск модели должен привести к пошаговому моделированию состояний процессора при выполнении каждой из команд.

**Эмулируемые компоненты**

Создается память размером 200 значений.

Первые сто значений определяются под ячейки данных.

Вторые сто значений определяются под ячейки команд

**Система команд**

Система команд, которые поддерживает процессор состоят из 18 бит, первые 4 бита отведены под команды, 7 под первое число и еще 7 под второе число.

Перевод команд происходит через битовое определение. Далее декодируются в 16 формат чисел и вызывают команду.

MEM\_DATA – команда записи числа в память.

WRITE\_MEMMORY – команда записи команд в память команд.

WRITE\_ONE\_READ - запись числа в первое указанное число из ячейки, которую указали вторым числом.

ADDW – увеличивает число на указанное значение.

WRITE\_DOUBLE\_READ - запись числа в первое указанное число из значения, полученного в ячейки число которое указали вторым.

COMPARISON - если число в первой указанной ячейке больше чем во второй указанной ячейке перезаписываем вторую ячейку значением из первой.

CONTROL\_NUMBER -если число из ячейки первого числа меньше чем число в ячейки второго числа, то 'pc' становится значением из ячейки минус один от второго числа.

**Методы внутри класса эмулятора**

Выполняет переход на следующую команду.

self.step()

Получение информации из адреса команд

self.fetch()

Декодирование команд

self.decode()

Завершение работы эмулятора и вывод информации

self.trace()

Запись числа в x из значения, полученного в ячейки y

self.write\_double\_read\_d(x,y)

Запись числа в x из ячейки y

self.write\_one\_read\_d(x,y)

Если число из ячейки x меньше чем число в ячейки y то 'pc' становится значением из ячейки y-1

self.control\_number\_d(z,c)

Сравнивает два числа из ячеек памяти и возвращает True или False

self.sravnenie\_d(x,y)

Записывает число x в y

self.write\_memory\_d(x,y)

Записывает число x из y в полученное значение

self.write\_number\_d(x,self.read\_memory(y))

Увеличивает число x на y

self.addw\_d(c,step)

Проверяет что число x не больше y и меняет их местами

self.comparison\_d(z,y)

Запись данных в память

cpu.mem\_data

**Вызываемые команды и их эквиваленты в процессоре**

write\_double\_read =2 = self.write\_double\_read\_d(x,y)

write\_one\_read =3 = self.write\_one\_read\_d(x,y)

control\_number =5 = self.control\_number\_d(z,c)

write\_mem = 7 = self.write\_memory\_d(x,y)

comparison =12 = self.comparison\_d(z,y)

addw =11 = self.addw\_d(c,step)

mem\_data =100 = cpu.mem\_data

CONTROL\_NUMBER= 'CONTROL\_NUMBER'# 0x5 # 5

COMPARISON= 'COMPARISON' # 0x12 # 12

ADDW= 'ADDW'# 0x11 # 11

WRITE\_ONE\_READ= 'WRITE\_ONE\_READ'# 0x3 # 3

WRITE\_DOUBLE\_READ= 'WRITE\_DOUBLE\_READ'# 0x2 # 2

WRITE\_MEMMORY= 'WRITE\_MEMMORY'# 0x7 # 7

MEM\_DATA ='MEM\_DATA'# 0x100 # запись числа в память

**Программа**

MEM\_DATA 0 5

MEM\_DATA 1 3000000

MEM\_DATA 2 1000

MEM\_DATA 3 20

MEM\_DATA 4 40

MEM\_DATA 5 400

MEM\_DATA 6 5000

WRITE\_MEMMORY 99 0

WRITE\_MEMMORY 90 1

WRITE\_MEMMORY 91 0

WRITE\_MEMMORY 92 0

WRITE\_MEMMORY 89 0

WRITE\_MEMMORY 88 8

WRITE\_ONE\_READ 89 0

WRITE\_ONE\_READ 99 1

ADDW 90 1

WRITE\_DOUBLE\_READ 91 90

COMPARISON 91 99

CONTROL\_NUMBER 90 1

**Ввод данных из txt файла**

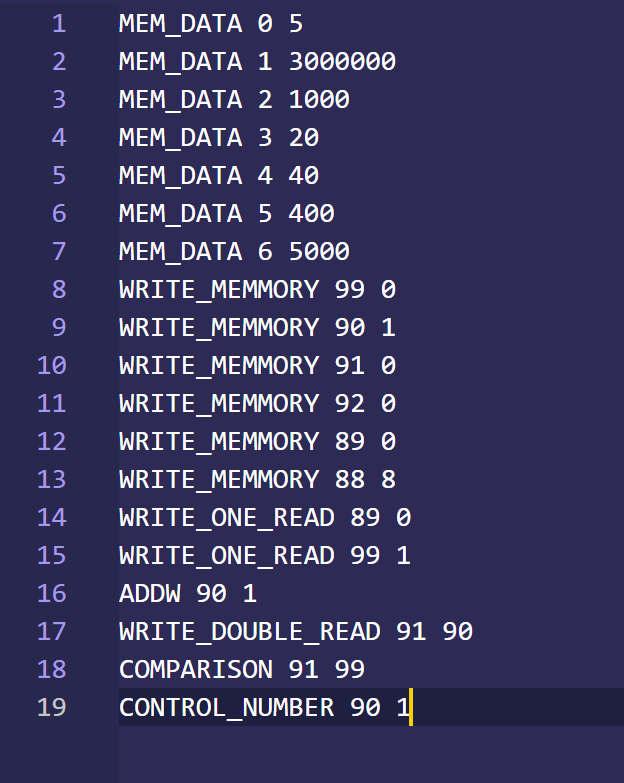


Рисунок 1 – Запись команд на языке assemble

**Вывод выполнения программы**

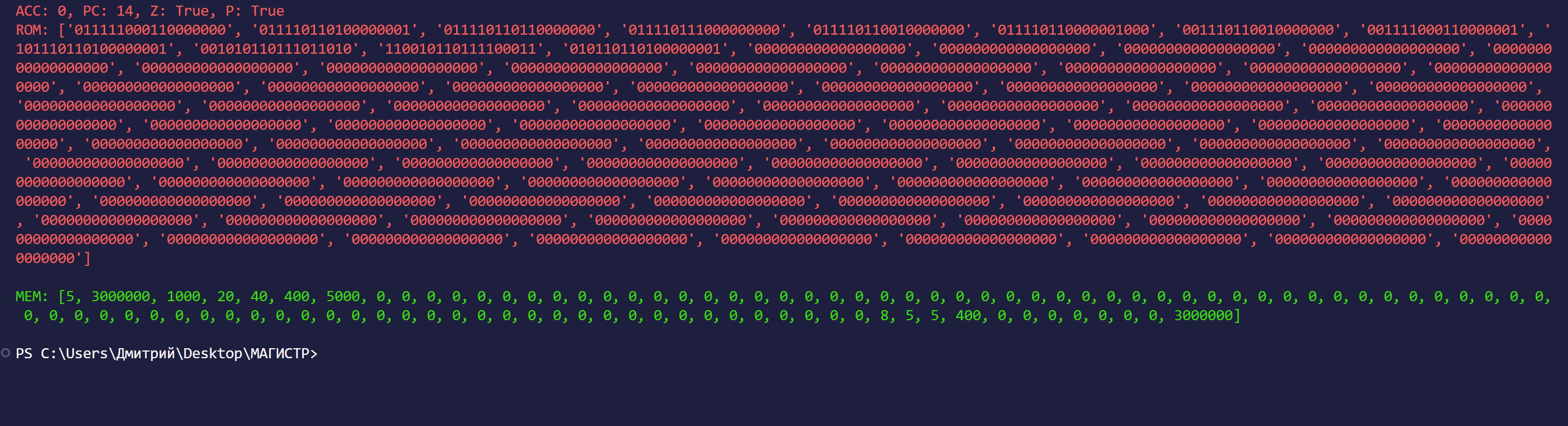


Рисунок 2 – Вывод эмулируемого процессора

На рисунке 2 показан вывод программы где в первом массиве указаны выполняемые команды, а во втором массиве вывод данных.

**Вывод**

Описали модели регистров и памяти процессора, а также проверили модели путем составления тестовой программы с предсказуемым результатом исполнения.