



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_MEMORY – 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

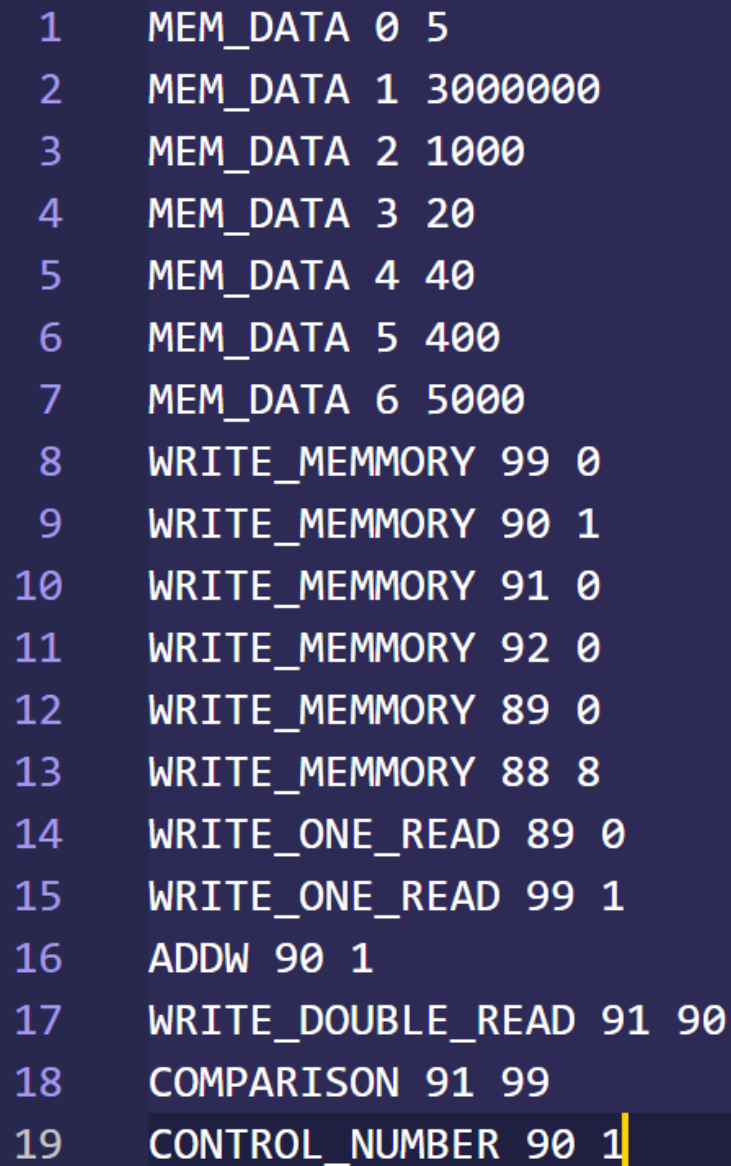
ADDW 90 1

WRITE_DOUBLE_READ 91 90

COMPARISON 91 99

CONTROL_NUMBER 90 1

Ввод данных из txt файла



```
1  MEM_DATA 0 5
2  MEM_DATA 1 3000000
3  MEM_DATA 2 1000
4  MEM_DATA 3 20
5  MEM_DATA 4 40
6  MEM_DATA 5 400
7  MEM_DATA 6 5000
8  WRITE_MEMMORY 99 0
9  WRITE_MEMMORY 90 1
10 WRITE_MEMMORY 91 0
11 WRITE_MEMMORY 92 0
12 WRITE_MEMMORY 89 0
13 WRITE_MEMMORY 88 8
14 WRITE_ONE_READ 89 0
15 WRITE_ONE_READ 99 1
16 ADDW 90 1
17 WRITE_DOUBLE_READ 91 90
18 COMPARISON 91 99
19 CONTROL_NUMBER 90 1
```

Рисунок 1 – Запись команд на языке assemble

Вывод выполнения программы

