ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Ассистент		А.Н.Долидзе
должность, уч.степень,звание	подпись, дата	инициалы,фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

«Программная реализация алгоритма выполнения целочисленной операции для архитектуры набора команд VAX»

по курсу: Организация ЭВМ и систем.

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.	4143		Е.Д.Тегай
		подпись, дата	инициалы,фамилия

Санкт-Петербург
2024

1. Задание

Вариант №2

Вариант исходного алгоритма продемонстрирован на рисунке 1.

4	Умножение целых чисел со знаком в	С коррекцией результата

5	дополнительном коде со сдвигом суммы частичных произведений вправо, неподвижным множимым и анализом множителя, начиная с младших разрядов.	С предварительным изменением знака
6		С преобразованием множителя

Рисунок 1 – Исходный алгоритм

На рисунке 2 продемонстрировано содержание индивидуального варианта.

Тип 1: перенос данных из памяти в регистры. Тип 2: сохранение результата в память

Операнд	Варианты																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Тип1	6	7	8	9	7	8	6	6	7	7	8	9	8	6	6	7	6	8
Тип2	7	9	7	8	6	7	8	7	9	8	7	8	7	9	9	9	8	9

где 6 - косвенная регистровая (простая косвенная) адресация, 8 - автоинкрементная (простая косвенная с автоувеличением), 7 - автодекрементная (простая косвенная с автоуменьшением) и 9 - косвенная автоинкрементная адресация (двойная косвенная с автоувеличением).

Рисунок 2 – Содержимое задания

Промежуточные расчёты:

Адрес начала расположения исходных данных: 19 + 20 = 39.

Адрес начала расположения команд программы: 19 + 120 = 139.

Типы адресации: 1 + 19mod18 = 2.

Перевод исходных данных в шестнадцатеричную систему счисления

Перевод исходных данных продемонстрирован в таблице 1.

Таблица 1

Имя вх./вых. переменной	Десятичное число	Шестнадцатеричный код	Адрес загрузки
A	-18	FFFFFFEE	R0
B	16	00000010	R1
X1 (сравнение номера разряда)	8	08	R7
X2 (загрузка для определения бита)	1	01	R6

Схема алгоритма программы

Исходная схема алгоритма программы показана на рисунке 3.

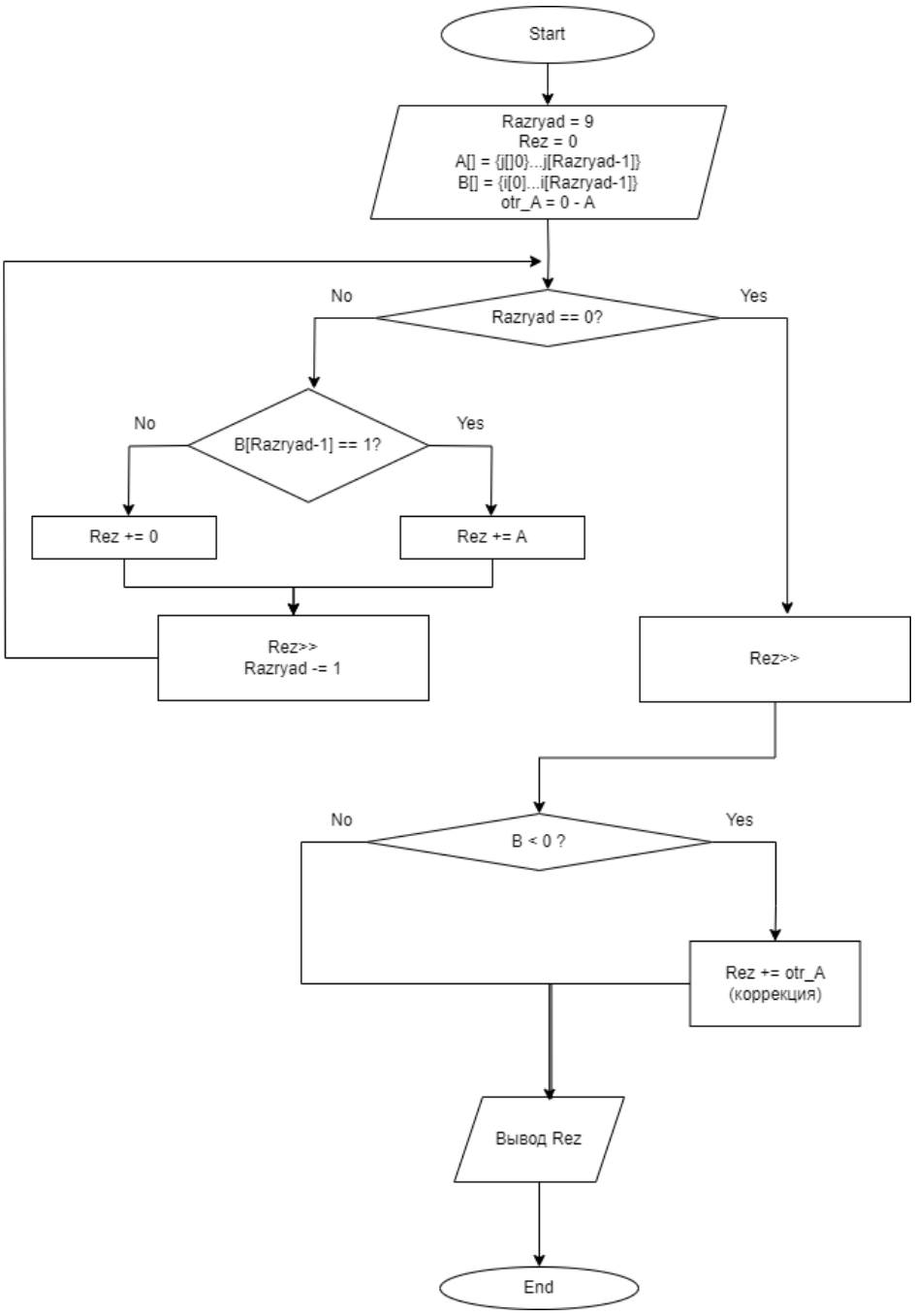

Рисунок 3 – Схема алгоритма

Пример использования автодекрементной адресации

Рассмотрим случай, когда необходимо занести исходное значение A в регистр R0. Для этого разместим исходное число в области памяти, это показано на рисунке 3.1.

	+0	+1	+2	+3
000000C0	00	00	00	00
000000D0	00	00	00	00
000000E0	00	00	00	00
000000F0	EE	FF	FF	FF

Рисунок 3.1 – Исходное значение A

Напишем в области памяти для программы соответствующую команду. Причем укажем, например, адрес, по которому находится исходное число, в регистре R1. Это показано на рисунке 3.2

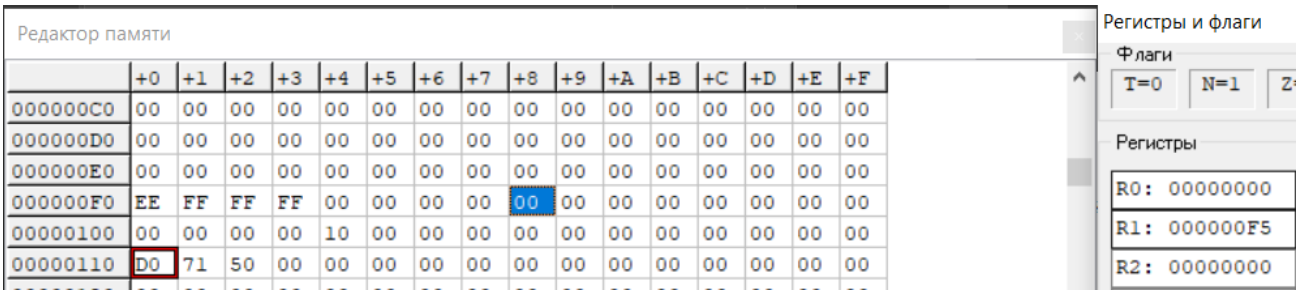

Рисунок 3.2 – Ситуация до работы команды

На рисунке 3.3 продемонстрирован результат работы команды.

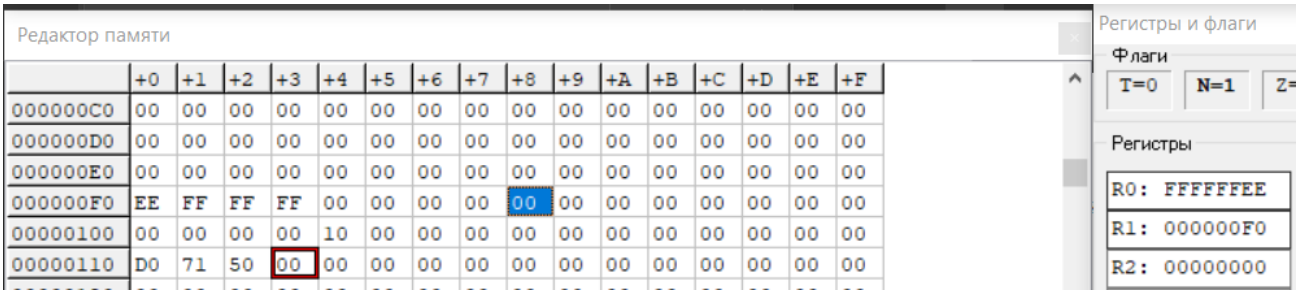

Рисунок 3.3 – Результат работы команды

Далее рассмотрим использование косвенной автоинкрементной адресации. Рассмотрим ситуацию, когда результат необходимо сохранить в памяти. Пусть результат хранится в регистре R4, а после работы команды он будет лежать в ячейке 0000000A. Исходное состояние показано на рисунке 3.4

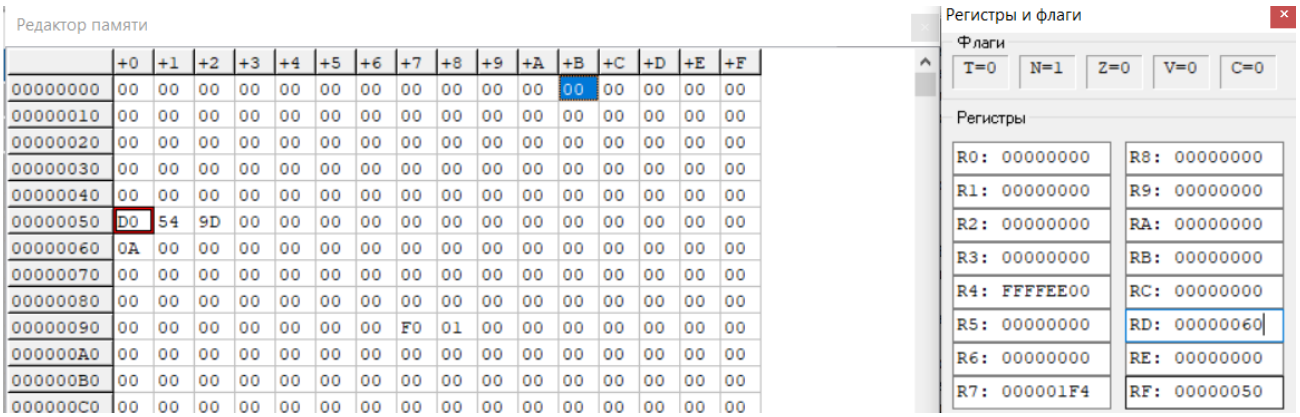

Рисунок 3.4 – Состояние до работы команды

На рисунке 3.5 показан результат.

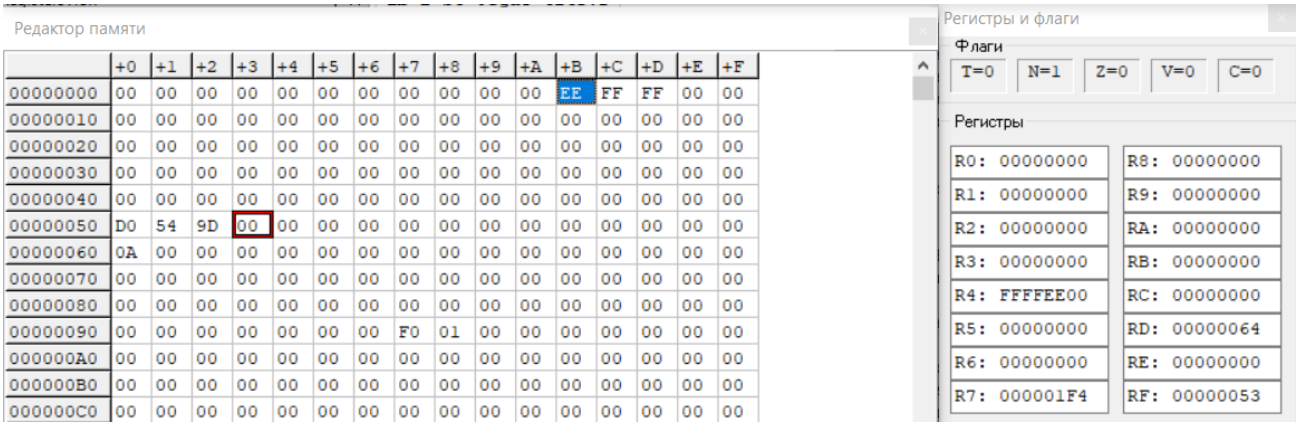

Рисунок 3.5 - Результат

Текст программы в мнемонических кодах

MOVL #0xFFFFFFEE, R0 ;Загружаем значение A в регистр R0

MOVL #0x00000010, R1;Загружаем значение B в регистр R1

ASHL #8, R0, R0 ;Сдвигаем на 8 бит влево значение в регистре R0

SUBL3 R0, R6, R5 ;Находим отр. A

MOVB #09, R7 ;Загружаем значение в регистр для дальнейшего сравнения

CMPL R3, R7 ;Сравниваем значения в регистрах (флаг N станет 1, если разряд не 8, иначе – Z станет 0)

BNEQU ; Переход, если Z не равно 0

BGEQ ; Переход в зависимости от значения флагов N и V

MOVB #1, R6 ;Загрузка единицы в регистр для определения бита

ASHL R3, R6, R6 ; Соответствующий сдвиг в зависимости от итерации

BITL R1, R6 ; Логическое «И»

BEQL ;Переход, если Z = 1

MOVL R2, R4 ; Сохранение результата

ASHL #8, R2, R2 ; Сдвиг на байт влево

INCL R3 ; Инкремент счётчика цикла

BRB ; Возвращение назад на проверку разряда

ADDL2 R0, R2 ;Сложение промежуточного результата и A

MOVL R2, R4 ; Сохранение результата

ASHL #8, R2, R2 ; Сдвиг на байт влево

INCL R3 ; Инкремент счётчика цикла

BRB ; Возвращение назад на проверку разряда

MOVL R2, R4 ; Сохранение результата

CMPL R1, #0 ; Определение отрицательности или положительности B

BLSS ; Переход, если B - отр

ADDL2 R4, R5 ; сложение с итоговым результатом отр. A

Текст программы в машинных кодах

D0 8F EE FF FF FF 50
D0 8F 10 00 00 00 51
78 8F 08 50 50
C3 50 56 55

D0 8F 08 00 00 00 57
D1 53 57
13 54

18 22 90 8F 01 56
78 53 56 56
D3 51 56
13 35

D0 52 54
78 8F FF 52 52
D6 53
31 A3 FF

C0 50 52
D0 52 54
78 8F FF 52 52
D6 53
31 D5 FF

D0 52 54
D1 8F 00 51
19

C0 54 55

Карта распределения памяти под команды и данные

Карта распределения памяти продемонстрирована на рисунке 4.

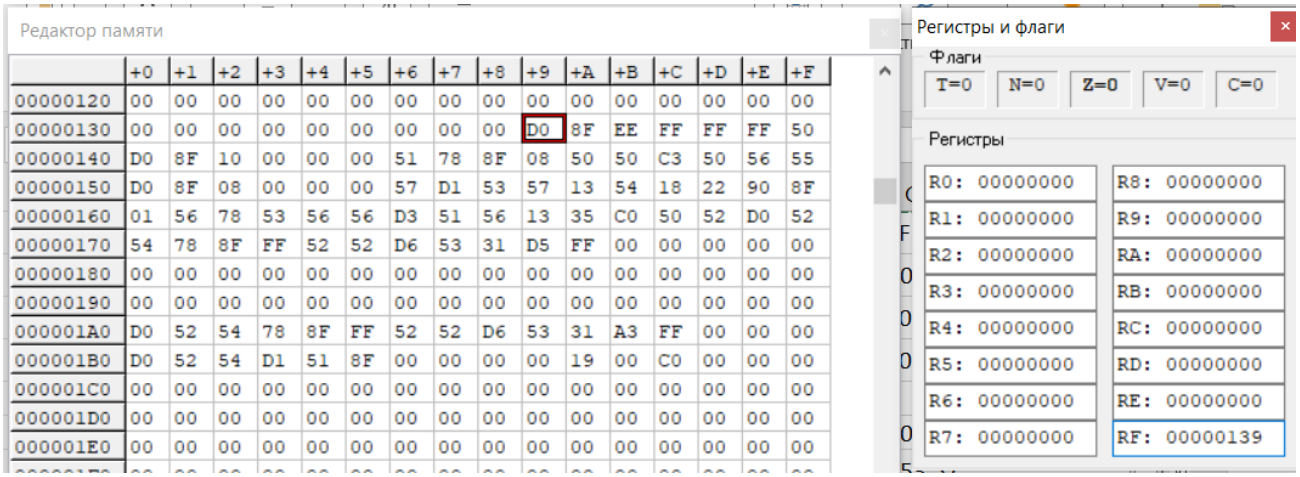

Рисунок 4 – Карта распределения памяти

Выполнение команды

На рисунке 5 показан скриншот в момент выполнения программы.

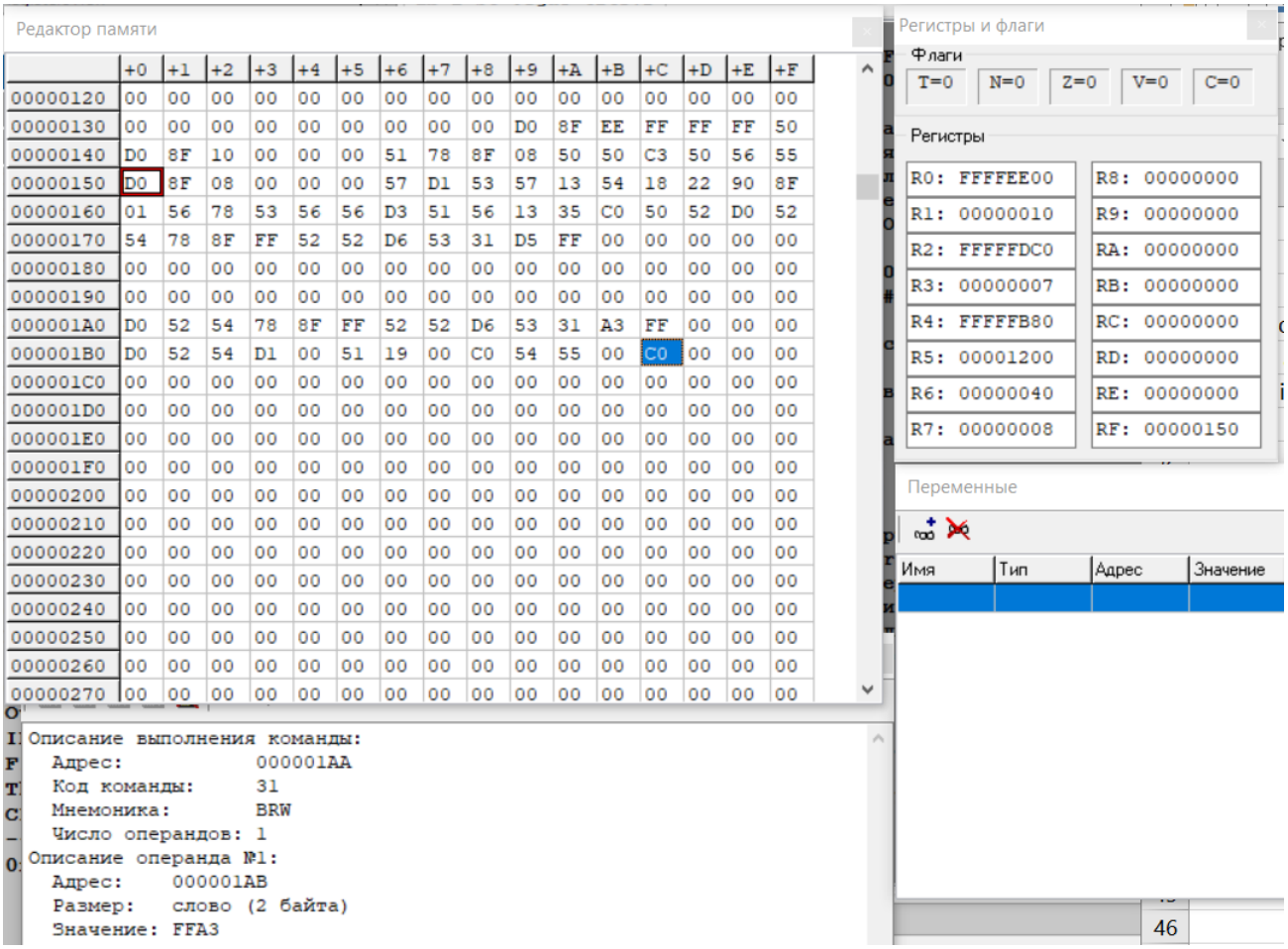

Рисунок 5 – Выполнение программы

Сравнение наборов команд ARM и VAX

Рассмотрим более подробно разницу команд. Это показано в таблице 2.

Таблица 2

ARM	**VAX**
MOV	MOVL
LSL	ASHL
SUB	SUBL3
CMP	CMPL
AND	BITL
ADD #1	INCL
ADD	ADDL2

Таблица трассировки

Искомая таблица продемонстрирована в таблице 3 для одной итерации цикла.

Таблица 3

Команда	Состояние памяти и регистров	
	Было	Стало
D0 8F EE FF FF FF 50	R0 = 00000000 RF = 00000139 N = 0	R0 = FFFFFFEE RF = 00000140 N = 1
D0 8F 10 00 00 00 51	R1 = 00000000 RF = 00000140 N = 1	R1 = 00000010 RF = 00000147 N = 0
78 8F 08 50 50	R0 = FFFFFFEE RF = 00000147 N = 0	R0 = FFFFEE00 RF = 0000014C N = 1

	V = 0	V = 1
C3 50 56 55	R5 = 00000000 RF = 0000014C N = 1 V = 1 C = 0	R5 = 00001200 RF = 00000150 N = 0 V = 0 C = 1
D0 8F 08 00 00 00 57	R7 = 00000000 RF = 00000150	R7 = 00000008 RF = 00000157
D1 53 57	RF = 00000157 N = 0	RF = 0000015A N = 1
13 54	RF = 0000015A	RF = 0000015C
18 22	RF = 0000015C	RF = 0000015E
90 8F 01 56	R6 = 00000000 RF = 0000015E N = 1	R6 = 00000001 RF = 00000162 N = 0
78 53 56 56	RF = 00000162 V = 0	RF = 00000166 V = 1
D3 51 56	RF = 00000166 Z = 0 V = 1	RF = 00000169 Z = 1 V = 0
13 35	RF = 00000169	RF = 000001A0
D0 52 54	RF = 000001A0	RF = 000001A3
78 8F FF 52 52	RF = 000001A3	RF = 000001A8
D6 53	RF = 000001A8 R3 = 00000000 Z = 1	RF = 000001AA R3 = 00000001 Z = 0
31 A3 FF	RF = 000001AA	RF = 00000150