

Лекція СПОСОБИ АДРЕСАЦІЇ ОПЕРАНДІВ ТА ФОРМАТИ КОМАНД

3.3 Команди множинного запису/читання

На рисунку 16 наведено формат команд множинного запису/читання.

LDM|STM{<cond>}<addressing_mode> <Rn>{!}, <registers>{^}

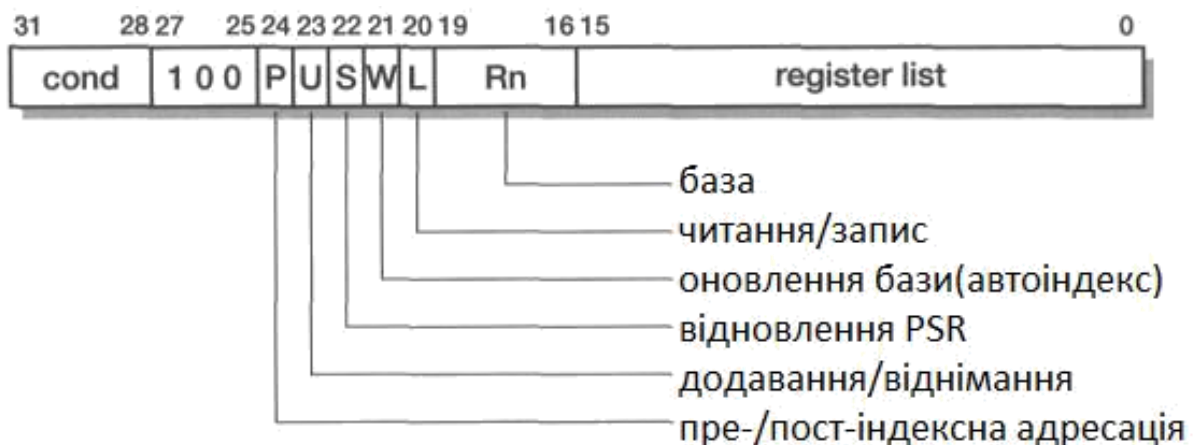


Рисунок 16 – Адресація множинного запису/читання

Нижче наведено пояснення окремих полів формату команд, наведеного на рисунку 16:

- 31 р..28 р (cond) формує компілятор в залежності від префіксу команди (таблиця 1), якщо останній використовується;
- 27 р дорівнює одиниці;
- 25 та 26 р дорівнюють нулю;
- 24 р (P) визначає, чи включене слово, яке адресоване базою, до діапазону комірок пам'яті (0 – так, 1 – ні);
- 23 р (U) визначає, яка операція виконується над базою: додавання (U = 1), або віднімання (U = 0);
- 22 р (S) встановлюється в 1, коли користувацькі регістри копіюються під час роботи в привілейованому режимі;
- 21 р (W) визначає, чи база оновлюється після пересилки;
- 20 р (L) визначає вид операції: завантаження (L=1), або збері-

гання (L=0);

–19 p..16 p (Rn) визначає один з регістрів основного регістрового файлу, який виступає в якості бази;

–15 p..0 p (register list) визначає, які регістри будуть використовуватись.

Зміну бітів L, P, U в залежності від використаних команд відображає таблиця 3.

Таблиця 3 – Зміна бітів L, P, U в залежності від використаних команд

	L	P	U
LDMDA (Decrement After)	1	0	0
LDMIA (Increment After)	1	0	1
LDMDB (Decrement Before)	1	1	0
LDMIB (Increment Before)	1	1	1
STMDA (Decrement After)	0	0	0
STMIA (Increment After)	0	0	1
STMDB (Decrement Before)	0	1	0
STMIB (Increment Before)	0	1	1

На рисунку 17 наведено пояснення виконання команд множинного запису/читання

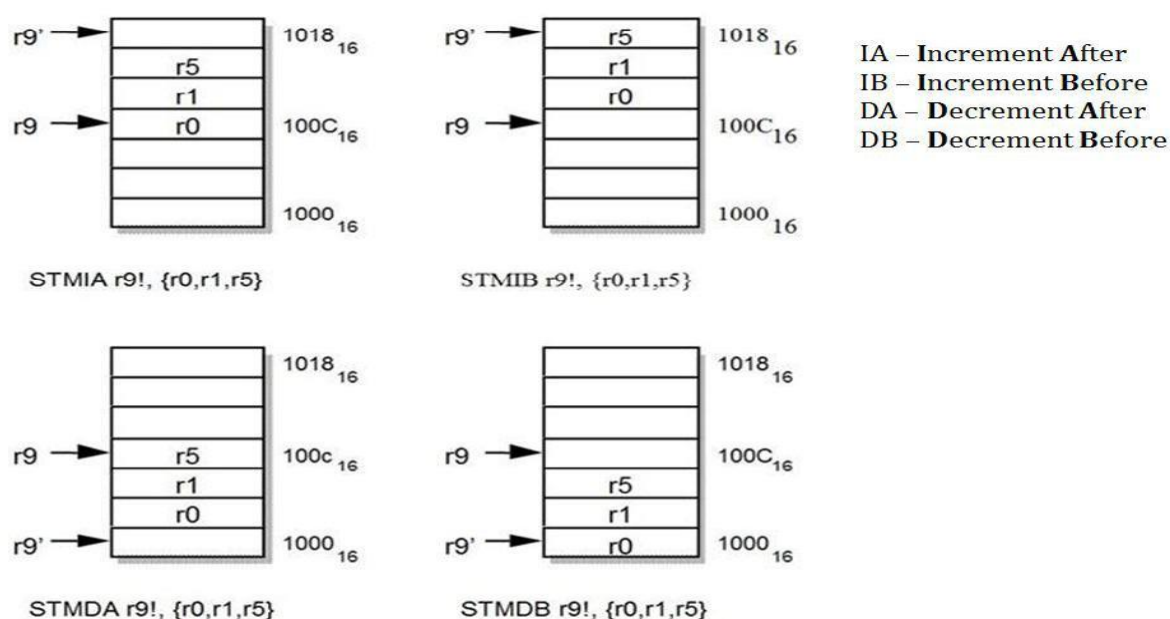


Рисунок 17 – Пояснення виконання команд множинного запису/читання

4.3.3.1 Приклад виконання команди STMIA

STMIA <Rn>!, <registers>

31	30	29	28	27	26	25	24	23	22	21	20	19	16	15	14	13	12	11	10	9	8	7	0
1	1	1	0	1	0	0	0	1	0	1	0	Rn	0	0	0	0	0	0	0	0	0	0	register_list

MOV R0,#0xFFFFFFFFAA ; R0 = 0xFFFFFFFFAA
MOV R1,#0xFFFFFFFFBB ; R1 = 0xFFFFFFFFBB
MOV R2,#0xFFFFFFFFCC ; R2 = 0xFFFFFFFFCC
STMIA R8,{R0,R1,R2} ; M(R8)=0xFFFFFFFFAA, M(R8+4)=0xFFFFFFFFBB,
M(R8+8)=0xFFFFFFFFCC; R8 = R8+12

E8A80007= 1110 1000 1010 1000 0000 0000 0111

3.3.2 Приклад виконання команди STMIB

MOV R0,#0xFFFFFFFFAA ; R0 = 0xFFFFFFFFAA
MOV R1,#0xFFFFFFFFBB ; R1 = 0xFFFFFFFFBB
MOV R2,#0xFFFFFFFFCC ; R2 = 0xFFFFFFFFCC
STMIB R8,{R0,R1,R2} ; M(R8+4)=0xFFFFFFFFAA, M(R8+8)=0xFFFFFFFFBB,
M(R8+12)=0xFFFFFFFFCC; R8 = R8+12

E9A80007= 1110 1001 1010 1000 0000 0000 0111

3.3.3 Приклад виконання команди STMDA

MOV R0,#0xFFFFFFFFAA ; R0 = 0xFFFFFFFFAA
MOV R1,#0xFFFFFFFFBB ; R1 = 0xFFFFFFFFBB
MOV R2,#0xFFFFFFFFCC ; R2 = 0xFFFFFFFFCC
STMDA R8,{R0,R1,R2} ; M(R8)=0xFFFFFFFFAA, M(R8-4)=0xFFFFFFFFBB, M(R8-8)=0xFFFFFFFFCC; R8 = R8-12

E8280007= 1110 1000 0010 1000 0000 0000 0111

3.3.4 Приклад виконання команди STMDB

STMDB SP!, <registers>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	0
1	1	1	0	1	0	0	1	0	0	1	0	1	1	0	1	0	R	0	0	0	0	0	0	0	register_list

MOV R0,#0xFFFFFFFFAA ; R0 = 0xFFFFFFFFAA

MOV R1,#0xFFFFFFFFBB ; R1 = 0xFFFFFFFFBB

MOV R2,#0xFFFFFFFFCC ; R2 = 0xFFFFFFFFCC

STMDB R8,{R0,R1,R2}; M(R8-4)=0xFFFFFFFFAA, M(R8-8)=0xFFFFFFFFBB, M(R8-12)=0xFFFFFFFFCC; R8 = R8-12;

E9280007= 1110 1001 0010 1000 0000 0000 0000 0111

4 Адресація команд роботи з співпроцесорами

На рисунку 18 наведено формат команд роботи зі співпроцесорами. Команди цього класу використовуються для читання (LDC) або запису (STC) регістрів співпроцесорів безпосередньо з / в пам'ять.

<LDC|STC>{<cond>}{L} <coproc>,<CRd>,<addressing_mode>

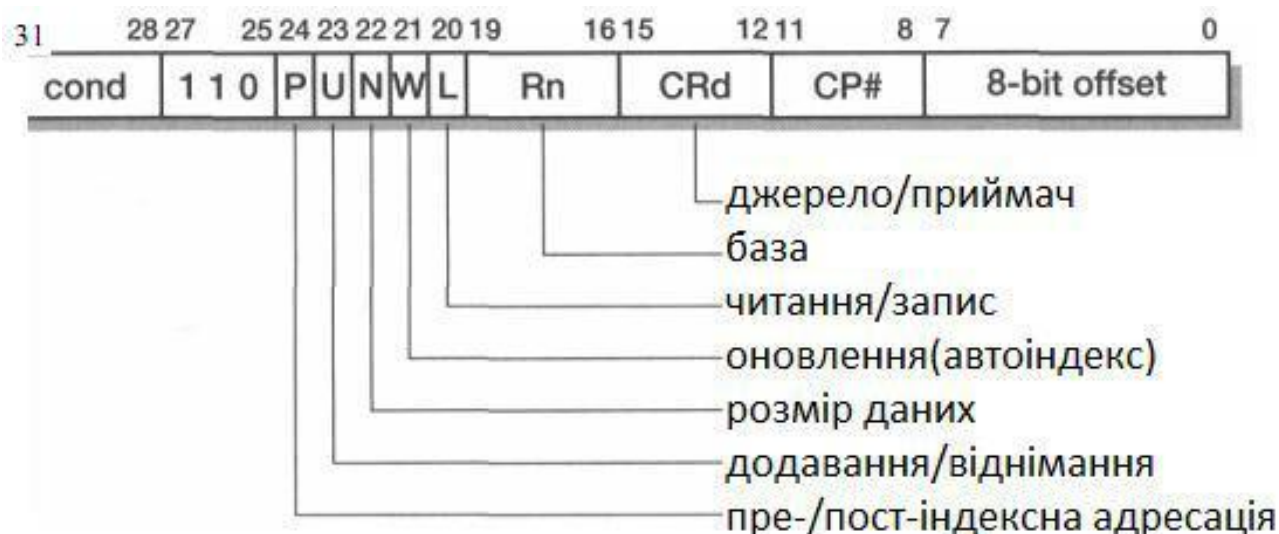


Рисунок 18 – Формат команд роботи з співпроцесорами

Нижче наведено пояснення окремих полів формату команд, наведених на рисунку 18:

- 31 р..28 р (cond) формує компілятор в залежності від префіксу команди (таблиця 4.1), якщо останній використовується;
- 27 р та 26 р дорівнюють одиниці;

- 25 p дорівнює нулю;
- 24 p (P) визначає тип адресації: P=0 – постіндексна (біт W визначає, яка саме), P=1 – преіндексна (біт W визначає, яка саме);
- 23 p (U) визначає яка операція виконується над базою: додавання (U = 1), або віднімання (U = 0);
- 22 p (N) значення довжини передачі задається співпроцесором: 0–1 слово; 1 – більше;
- 21 p (W) визначає, чи база оновлюється після пересилки: 0 – ні; 1 – так;
- 20 p (L) визначає вид операції: завантаження (L=1), або зберігання (L=0);
- 19 p..16 p (Rn) визначає один з регістрів основного реєстрового файлу, який виступає в якості бази;
- 15 p..12 p (CRd) визначає регістр співпроцесора;
- 11 p..8 p (CP#) визначає номер співпроцесора;
- 7 p..0 p (8-bit offset) –8-бітне зміщення.

Поле CP # використовується для ідентифікації того співпроцесора, який повинен передати або прийняти дані, при цьому команду виконає тільки той співпроцесор, номер якого збігається з номером у полі CP #.

Регістр CRd завжди виконує роль реєстра, який має бути переданий (або перший регістр зі списку при блоковій передачі), а біт N визначає довжину передачі: при N = 0 обмін проводиться тільки з одним регістром (CRd), при N = 1 – зі всіма регістрами (для перемикання контексту завдань).

Нижче наведено приклади деяких команд розглянутого формату:

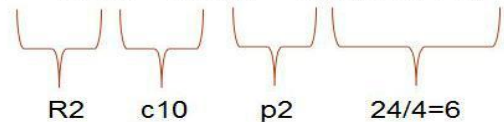
4.1 Команда з безпосереднім зміщенням з преіндексною адресацією без зміни бази

[<Rn>, #+/-<offset_8>*4]



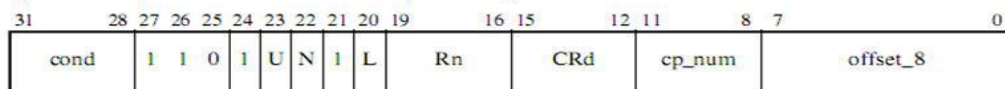
STC p2, c10, [R2, #24]

ED82A206 = 1110 1101 1000 0010 1010 0010 0000 0110



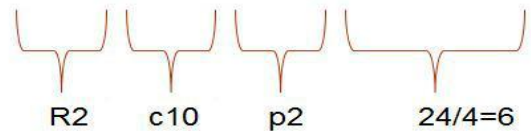
4.2 Команда з безпосереднім зміщенням з преіндексною адресацією зі зміною бази

[<Rn>, #+/-<offset_8>*4]!



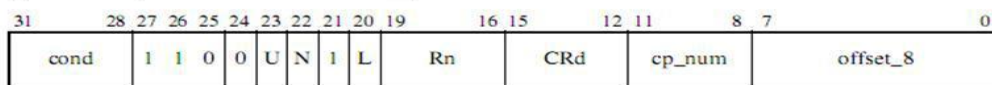
STC p2, c10, [R2, #24] !

EDA2A206 = 1110 1101 1010 0010 1010 0010 0000 0110



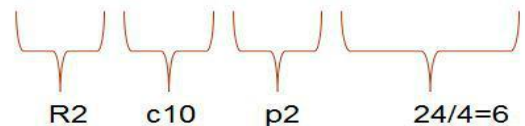
4.3 Команда з безпосереднім зміщенням з постіндексною адресацією

[<Rn>], #+/-<offset_8>*4



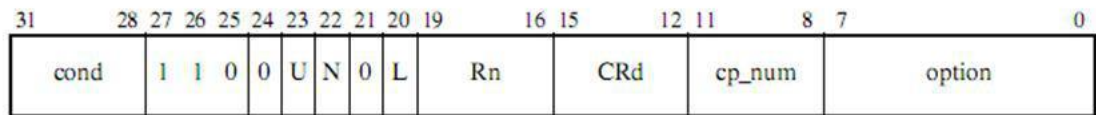
STC p2, c10, [R2], #24

ECA2A206 = 1110 1100 1010 0010 1010 0010 0000 0110



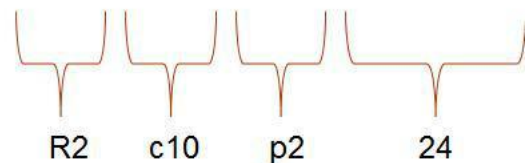
4.4 Команда з безіндексною адресацією

[<Rn>], <option>



STC p2, c10, [R2], {24}

EC82A218 = 1110 1100 1000 0010 1010 0010 0001 1000



5 Приклади способів адресацій

1 Команда: ADDS R1, R2, #0xFF00.

Коментар до команди: $R1 \leftarrow R2 + 0xFF00$, запис до регістра R1 результату додавання R2 та 0xFF00.

Способи адресації операндів: лівий – регістрова, правий – безпосередня, отримувач результату – регістрова.

2 Команда: LDRSB R0, [R1, R2].

Коментар до команди: завантаження байта зі знаком у регістр R0 з пам'яті за адресою: $(R1 + R2)$.

Способи адресації операндів: джерело – преіндексна з регістровим зміщенням без оновлення бази, отримувач – регістрова.

3 Команда: LDR R0, [R1], R2, LSR#2.

Коментар до команди: завантаження слова: $R0 \leftarrow M(R1)$; $R1 \leftarrow R1 + (R2 \cdot 4)$.

Способи адресації операндів: джерело – постіндексна з масштабованим регістровим зміщенням, отримувач – регістрова.

4 Команда: ADD R1, R2, #0xFF00.

Коментар до команди: $R1 \leftarrow R2 + 0xFF00$, запис до регістра R1 результату додавання R2 та 0xFF00.

Способи адресації операндів: перший операнд – регістрова – R2, другий операнд – безпосередня – 0xFF00, отримувач – регістрова – R1.

5 Команда: ADDS R9, R4, R5, LSR R1.

Коментар до команди: $R9 \leftarrow R4 + R5/2^{R1}$, запис до регістра R9 результату додавання R4 та $R5/2^{R1}$.

Способи адресації операндів: перший операнд – регістрова, другий – регістрова з масштабуванням, отримувач – регістрова.

6 Команда: STR R0, [R2, #12].

Коментар до команди: зберігання слова: $M(R1 + 12) \leftarrow R0$.

Способи адресації операндів: джерело – регістрова, приймач – преіндексна з безпосереднім зміщенням без оновлення бази.

7 Команда: STR R0, [R1, #0xAB]!.

Коментар до команди: $R1 \leftarrow R1 + 0xAB$; $M(R1) \leftarrow R0$ (зберігання слова).

Способи адресації операндів: джерело – регістрова, отримувач – преіндексна з безпосереднім зміщенням та оновленням бази.

8 Команда: STR R5, [R2], #25.

Коментар до команди: $M(R2) \leftarrow R5$; $R2 \leftarrow R2 + 25$ (зберігання слова).

Способи адресації операндів: джерело – регістрова, отримувач – постіндексна з безпосереднім зміщенням.

9 Команда: STR R0, [R1, R2].

Коментар до команди: $M(R1 + R2) \leftarrow R0$ (зберігання слова).

Способи адресації операндів: джерело – регістрова, приймач – преіндексна з регістровим зміщенням без оновлення бази.

10 Команда: STR R1, [R2].

Коментар до команди: $M(R2) \leftarrow R1$ (зберігання слова)

Способи адресації операндів: джерело – регістрова, отримувач – непряма.

11 Команда: STR R0, [R1], R2.

Коментар до команди: $M(R1) \leftarrow R0$; $R1 \leftarrow R1 + R2$ (зберігання слова)

Способи адресації операндів: джерело – регістрова, отримувач – постіндексна з регістровим зміщенням.

12 Команда: STR R0, [R1, R2, LSL#2].

Коментар до команди: $M(R1 + R2 * 2^2) \leftarrow R0$ (зберігання слова).

Способи адресації операндів: джерело – регістрова, отримувач – преіндексна з масштабованим регістровим зміщенням без оновлення бази.

13 Команда: STR R0, [R1, R2, LSL#2]!.

Коментар до команди: $R1 \leftarrow R1 + R2 * 2^2$; $M(R1) \leftarrow R0$ (зберігання слова).

Способи адресації операндів: джерело – регістрова, отримувач – преіндексна з масштабованим регістровим зміщенням та оновленням бази.

14 Команда: STR R0, [R1], R2 LSL#2.

Коментар до команди: $M(R1) \leftarrow R0$; $R1 \leftarrow R1 + R2 * 2^2$ (зберігання слова).

Способи адресації операндів: джерело – регістрова, отримувач – постіндексна з масштабованим регістровим зміщенням.

15 Команда: STR R2, [R1].

Коментар до команди: $M(R1) \leftarrow R2$ (зберігання слова).

Способи адресації операндів: джерело – регістрова, отримувач – непряма.

16 Команда: ANDS R1, R0, #0x0FA00000.

Коментар до команди: $R1 \leftarrow R0 \& 0x0FA00000$, запис до регістра R1 результату логічного множення R0 та 0x0FA00000.

Способи адресації операндів: перший операнд – регістрова, другий операнд – безпосередня, отримувач – регістрова.

17 ORRS R3, R0, #0xAB.

Команда:

Коментар до команди: $R3 \leftarrow R0 \vee 0xAB$, запис до регістра R3 результату логічного додавання R0 та 0xAB.

Способи адресації операндів: перший операнд – регістрова, другий операнд – безпосередня, отримувач – регістрова.

18 EORS R3, R0, #0x35.

Команда:

Коментар до команди: $R3 \leftarrow R0 \oplus 0x35$, запис до регістра R3 результату логічного додавання за модулем 2: R0 та 0x35.

Способи адресації операндів: перший операнд – регістрова, другий – безпосередня, отримувач – регістрова.

19 Команда: SBCS R3, R2, #0xDE.

Коментар до команди: $R3 \leftarrow R2 - 0xDE - C_F$, запис до регістра R3 результату віднімання від регістра R2 безпосереднього операнда 0xDE та інверсії C_F (C_F).

Способи адресації операндів: перший операнд – регістрова, другий – безпосередня, C_F – неявна, отримувач – регістрова

20 Команда: LDRSB R0, [R1, #35].

Коментар до команди: завантаження байта зі знаком –

$R0 \leftarrow M(R1 + 35)$.

Способи адресації операндів: джерело – преіндексна з безпосереднім зміщенням без оновлення бази, отримувач – регістрова.

21 Команда: LDRB R0, [R1, 0x35]!.

Коментар до команди: завантаження байта –

$R1 \leftarrow R1 + 0x35$; $R0 \leftarrow M(R1)$.

Способи адресації операндів: джерело – преіндексна з безпосереднім зміщенням та оновленням бази, отримувач – регістрова.

22 Команда: LDR R0, [R1], #0x78.

Коментар до команди: завантаження слова – $R0 \leftarrow M(R1)$; $R1 \leftarrow R1 + 0x78$.

Способи адресації операндів: джерело – постіндексна з безпосереднім зміщенням, отримувач – регістрова.

23 Команда: LDR R0, [R1, R2].

Коментар до команди: завантаження слова – $R0 \leftarrow M(R1 + R2)$.

Способи адресації операндів: джерело – преіндексна з регістровим зміщенням без оновлення бази, отримувач – регістрова.

24 Команда: LDR R0, [R1, R2]!.

Коментар до команди: завантаження слова – $R1 \leftarrow R1 + R2$; $R0 \leftarrow M(R1)$.

Способи адресації операндів: джерело – преіндексна з регістровим зміщенням та оновленням бази, отримувач – регістрова.

25 Команда: LDR R0, [R1], R2.

Коментар до команди: завантаження слова – $R0 \leftarrow M(R1)$; $R1 \leftarrow R1 + R2$.

Способи адресації операндів: джерело – постіндексна з регістровим зміщенням, отримувач – регістрова.

26 Команда: LDR R0, [R1, R2, LSL#2].

Коментар до команди: завантаження слова – $R0 \leftarrow M(R1 + R2 * 2^2)$.

Способи адресації операндів: джерело – преіндексна з масштабованим регістровим зміщенням без оновлення бази, отримувач – регістрова.

27 Команда: LDR R0, [R1, R2, LSL#2]!.

Коментар до команди: завантаження слова – $R1 \leftarrow R1 + R2 * 2^2$;

$R0 \leftarrow M(R1).$

Способи адресації операндів: джерело – преіндексна з масштабованим регістровим зміщенням та оновленням бази, отримувач – регістрова.

28 Команда: $LDR R0, [R1], R2, LSL\#2.$

Коментар до команди: завантаження слова – $R0 \leftarrow M(R1);$

$R1 \leftarrow R1 + R2 * 2^2.$

Способи адресації операндів: джерело – постіндексна з масштабованим регістровим зміщенням, отримувач – регістрова.

29 Команда: $LDR R2, [R1].$

Коментар до команди: завантаження слова – $R2 \leftarrow M(R1).$

Способи адресації операндів: джерело – непряма, отримувач – регістрова.

30 Команда: $RSCS R3, R1, R2.$

Коментар до команди: $R3 \leftarrow R2 - R1 -$, запис до регістра R3 результату зворотного віднімання від регістра R2 регістра R1 та інверсії пра-порця C_F ().

Способи адресації операндів: перший операнд – регістрова, другий – регістрова, прапорець C_F – неявна, отримувач – регістрова.