

SAAD 16 и SAAD 18

1 содержит дополнения 16 и содержит дополнение в

Syntax

(op) {cond}, (Rd), Rn, Rm

Где

Орлюбой из

SAAD 16: выполняет 2 16-битных целых сложения

SAAD 8: выполняет 4 16-битных целых сложения

Cond – необязательный условный код (см. Условное выполнение на стр.64)

Rd – регистратор-приемник

Rn –счетчик проведения операций

Rm – второй счетчик проведения операций (действий)

Операции:

Используйте эту инструкцию для представления об сокращений или байте добавленных параллельно

SAAD 16 инструкция(программа)

Добавляйте любое сокращение из первой операции к соответствующему сокращению из первой операции

-напишите результат соответствующий сокращению из места назначения счетчика (регистратора)

SAAD 8 инструкция

1.Добавьте любой бит из 1-ой операции соответствующий бит из второй операции

2. запишите результат соответствующий биту из места регистратора

Ограничения

Не использовать SP и PC

Условные обозначения:

Это инструкция не изменяет обозначения:

SAAD 16, R1, Ro; Добавлено сокращение в Ro к соответствующему сокращению R1 и запишем частичный результат

SAAD 8, R4, Ro, R5: Добавлено байт к Ro к соответствующему байту R5 и записать соответствующий байт в R4.

3.5.11 SHASX SHSAX

Способны складывать и вычитать с обменом. Способен вычитать и добавлять с обменом

Syntax

(op) {cond}, (Rd), Rn, Rm

Где op любой из:

SHASX: прибавляет, вычисляет с обменом, и уменьшает вдвое

SHSAX: вычитает и добавляет вдвое с обменом

Cond – необязательный условный код (см. Условное выполнение на стр.64)

Rd – регистратор-приемник

Rn –счетчик проведения операций

Rm, Rn – регистрируют проходящие 1-ые и 2-ые операции

Операции

SHASX инструкция:

1. Добавляйте первое сокращение из 1-ой операции с концом сокращения второй операции
2. Запишите получившийся результат к первому сокращению реестра назначения, сдвинув на один бит вправо, в результате чего разделите на два, или сократите вдвое
3. Отбросьте первое сокращение второй операции от конца главного слова первой операции
4. Запишите полное сокращение полученное в результате деления в конце сокращения регистратора в результате чего разделите на 2, или сократите вдвое

SHSAX инструкция:

1. Отнимите конечную часть сокращения второй операции от начала сокращения 1-ой операции
2. Запишите получившийся результат, полученный в результате добавления конца сокращения- регистратора приемника, сдвинув на 1 бит вправо, вызывающий сокращение вдвое
3. Добавьте к окончанию сокращения 1 операции начало сокращения второй операции
4. Запишите полученное сокращение, полученное в результате деления в начале сокращения вдвое на регистраторе0приемнике, сдвинув на один бит вправо, вызывающее сокращение вдвое

Ограничения:

Не использовать SP и PC

Условные обозначения:

Это инструкция не изменяет обозначения:

SHASX R4, R7, R2: Добавлено начало сокращения R4 к концу сокращения R2; и записан полученный результат в начале сокращения R7; отнято сокращение в начале R2 конца сокращения R4 и записан сокращенный вариант в конце сокращения R7

SHSAX Ro, R3, R5: Отнят конец сокращения R5 от начала сокращения R3, записан сокращенный результат в начале сокращения Ro, добавлено начало сокращения R5 к концу сокращения R3 и записан сокращенный результат в конце сокращения Ro

3.5.12 SHSUB16 и SHSUB8

Содержит дополнение 16 и дополнение 8

Syntax

(op) {cond}, (Rd), Rn, Rm

Где

op любой из:

SHSUB16: выполняет 2 16-битных вычитания

SHSUB8: выполняет 2 8-битных вычитания

Cond – необязательный условный код (см. Условное выполнение на стр.64)

Rd – регистратор-приемник

Rn –счетчик проведения операций

Rm – второй счетчик проведения операций (действий)

Операции

Используйте эту инструкцию к добавленным 16 и 8-битным данным и затем сокращайте вдвое результат перед записью в регистратор- приемник

SHSUB16 инструкция:

1. Вычитайте каждое сокращение второй операции с соответствующего сокращения 1-ой операции
2. Перестановите результат на один бит вправо, сократите данные
3. Запишите полученную половину сокращения в регистраторе-приемнике

SHSUB8 инструкция:

1. Вычислите каждый бит из второй операции с соответствующего бита 1-ой операции
2. Перестановите результат на один бит вправо, сократите данные
3. Запишите соответствующий подписанный битный результат в регистраторе-приемнике

Ограничения:

Не использовать SP и PC

Условные обозначения:

В этой инструкции сохранены обозначения:

SHSUB16 R1, Ro: вычите сокращение в Ro с соответствующего сокращения из R1 и запишите соответствующее сокращение R1

SHSUB8 Ro, R4, R5: вычите биты из Ro и соответствующего бита в R5 и запишите соответствующий бит в R4

3.5.13 SSUB16 и SSUB 8

Подписанные данные вычисления 16 и вычисление 8

Syntax

(op) {cond}, (Rd), Rn, Rm

Где

op любой из:

SSUB 16: предпочтительно 2 16-битных целочисленное вычитание

SSUB 8: предпочтительно 4 8-битных целочисленное вычитание

Cond – необязательный условный код (см. Условное выполнение на стр.64)

Rd – регистратор-приемник

Rn –счетчик проведения операций

Rm – второй счетчик проведения операций (действий)

Операции:

Используйте эту инструкцию для изменения порядка бита данных

SSUB16 инструкция:

1. Отнимайте каждое сокращение с второй операции с соответствующего сокращения 1-ой операции
2. Запишите разлчие в результате двух подписанных сокращений в соответствующих сокращениях приемника-регистратора

SSUB 8 инструкция:

1. Вычитите каждый бит второй операции из соответствующего бита первой операции
2. Запишите разлчие результата и подписанных битов в соответствующих битах в приемнике-регистраторе

Ограничения:

Не использовать SP и PC

Условные обозначения:

В этой инструкции сохранены обозначения:

SSUB 16, R1, Ro: вычитание сокращения в Ro с соотв. Сокращения в R1 и записывайте соот. Сокращение в R1.

SSUB 8 Ro, R4, R5: вычитайте бит из R5 с соот. Битов в Ro и записывайте результат в соответствующих битах R4

3.5.14 SASX и SSAX

Содержит вычитание и суммирование с обменом и содержит суммирование и вычитание с обменом.

Syntax

(op) {cond}, (Rd), Rn, Rm

Где op любой из:

SHASX: прибавляет, вычисляет с обменом, и уменьшает вдвое

SHSAX: вычитает и добавляет вдвое с обменом

«Cond» – необязательный условный код (см. Условное выполнение на стр.64)

«Rd» – регистратор-приемник

«Rn» – счетчик проведения операций

«Rm, Rn» – счетчик , проходящих первичных и вторичных операций

Операции

SASX инструкция:

1. Добавляйте подписанное начало сокращения первой операции с подписанным концом сокращения второй операции
2. Запишите подписанный полученный результат суммирования начало сокращений регистр-приемника
3. Отнимите подписанный конец сокращения второй операции из начала подписанного полного слова 1-ой операции
4. Запишите полученный вычитанием результат в конец сокращения регистратора-приемника

SSAX инструкция

1. Отнимите подписанный конец сокращения второй операции из начала подписанного полного слова 1-ой операции
2. Добавьте подписанное начало сокращения первой операции в подписанный конец сокращения второй операции
3. Запишите полученный вычитанием результат в начале сокращения регистратора-приемника

Ограничения:

Не использовать SP и PC

Условные обозначения:

Эта инструкция не влияет на условные обозначения:

SASX, Ro, R4, R5: добавьте начало сокращения из R4 и конец сокращения из R5 и запишите начало сокращения в R0. Вычтите конец сокращения R5 из начала сокращения R4 и запишите конец сокращения в Ro.

SSAX R7, R3, R2: вычтите начало сокращения в R2 с конца сокращения в R3 и запишите в конец сокращения R7 добавьте начало сокращения R3 с концом сокращения R2 и запишите начало сокращения R7.

3.5.15TST и TEQ

Тестируем биты и эквиваленты

Syntax:

TST "cond" Rn, Operand 2(операция)

TEQ Rn, Operand 2(операция)

"cond" - необязательный условный код (см. Условное выполнение на стр.64)

«Rn» – регистратор первичных операций

Operand 2(операция) гибкий вторичный оператор(см Гибкие вторичные операторы на стр. 59) деталей опции

Операции:

Эта инструкция тестирует значение в регистраторе против Operand 2. Они обновляют флаги условия на основе результата, но не пишут результат в регистраторе.

Инструкция TST выполняет поразр. Операцию и на значение в Rn и в значении Operand 2. Это совпадает с инструкцией ANDS, за исключением того, что она отбрасывает результат

Чтобы протестировать, является ли Rn 0 или 1, используйте инструкцию TST с Operand 2 константой, у которой есть тот набор битов к 1 и все другие биты, очищенные к 0.

Инструкция TEQ выполняет работу битового исключающего «или» на значениях в Rn и значении Operand 2. Это совпадает с инструкцией EORS, за исключением того, что она отображает результат. Используйте инструкцию TEQ чтобы протестировать если 2 значения равны, не влияет на V и флаги C.

TEQ также полезен для тестирования знака значения. После сравнения флага N - логическое монополюное или знаковых битов этих 2-х операций.

Ограничения:

Не использовать SP и PC

Условные обозначения:

Эта инструкция:

-загружает N и Z флаги в соответствии с результатом

-может загружать флаг C в процессе расчета Operand 2 (см. гибкие вторичные операторы на стр. 59)

-не влияет на флаг V

Пример:

TST R0, #0X3F8: представляет bitwise and в R0 значение 0X3F8

APSR загружает но результат отброшен.

3.5.16 UADD16 и UADD8

Без знака добавляют 16, и Без знака добавляют 8

Syntax

op {cond} {Rd.}, Rn, Rm

где:

- op - один из:

UADD16: Выполняет два 16-разрядных дополнения целого без знака.

UADD8: Выполняет четыре 8-разрядных дополнения целого без знака.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр
- 'Rn' - первый регистр, содержащий операнд
- 'Rm' - второй регистр, содержащий операнд

Работа

Используйте эти инструкции, чтобы добавить 16-и 8-разрядные данные без знака:

Инструкция UADD16:

1. Добавляет каждое сокращение от первого операнда до соответствующего сокращения второго операнда.
2. Пишет результат без знака в соответствующих сокращениях целевого регистра.

Инструкция UADD16:

1. Добавляет каждый байт первого операнда к соответствующему байту второго операнда.
2. Пишет результат без знака в соответствующем байте целевого регистра.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не изменяют флаги.

Примеры

UADD16 R1, R0;

Добавляют сокращение в R0 к соответствующему сокращению R1,; записи к соответствующему сокращению R1

UADD8 R4, R0, R5;

Добавляют байты R0 к соответствующему байту в R5 и записях ; к соответствующему байту в R4.

3.5.17 UASX и USAX

Добавьте и вычитите с обменом и Вычитите и добавьте с обменом.

Syntax

ор {cond} {Rd.}, Rn,Rm

где:

- ор - один из:

UASX: Добавьте и вычитите с обменом.

USAX: Вычитите и добавьте с обменом.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр
- 'Rn' 'Rm' являются регистрами, содержащими первые и вторые операнды

Работа

Инструкция UASX:

1. Вычитает главное сокращение второго операнда от конец сокращения первого операнда.
2. Пишет результат без знака вычитания в конец сокращения целевого регистра.
3. Добавляет главное сокращение первого операнда с нижним сокращением второго операнда.
4. Пишет результат без знака дополнения к главному сокращению целевого регистра.

Инструкция USAX:

1. Добавляет нижнее сокращение первого операнда с главным сокращением второго операнда.
2. Пишет результат без знака дополнения к нижнему сокращению целевого регистра.
3. Вычитает нижнее сокращению второго операнда от главного сокращению первого операнда.
4. Пишет результат без знака вычитания в главное сокращение целевого регистра.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не влияют на флаги кода условия.

Примеры

UASX R0, R4, R5;

Добавляет главное сокращение R4, к концу сокращения R5 и записи к главному сокращению R0

Вычитает нижнее сокращение R5 от главного сокращения R0 и записи, к концу сокращения R0

USAX R7, R3, R2;

Вычитает главное сокращение R2 от нижнего сокращения R3; и записи к концу сокращения R7

Добавляет главное сокращение R3, к концу сокращения R2 и записи к главному сокращению R7.

3.5.18 UNADD16 и UNADD8

Сокращение вдвое без знака добавляет 16, и сокращение вдвое Без знака добавляют 8

Syntax

op {cond} {Rd.}, Rn,Rm

где:

- op - любой из:

UNADD16: сокращение вдвое Без знака добавляет 16.

UNADD8: сокращение вдвое Без знака добавляет 8.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр
- 'Rn' - регистр, содержащий первый операнд
- 'Rm' - регистр, содержащий второй операнд

Работа

Используйте эти инструкции, чтобы добавить 16-и 8-разрядные данные и затем разделить на два результат прежде, чем записать результат в целевой регистр:

Инструкция UNADD16:

1. Добавляет каждое сокращение от первого операнда до соответствующего сокращения второго операнда.
2. Переставляет результат сокращения на один бит вправо, деля на два данные.
3. Пишут результаты без знака в соответствующее сокращение в целевом регистре.

Инструкция UNADD8:

1. Добавляет каждый байт первого операнда к соответствующему байту второго операнда.
2. Переставляет результат байта на один бит вправо, деля на два данные.
3. Пишут результаты без знака в соответствующем байте в целевом регистре.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не изменяют флаги.

Примеры

UNADD16 R7, R3;

Добавляют полуслова в R7 к соответствующему соукращению R3 & записи разделили на два результат к соответствующему сокращению в R7

UNADD8 R4, R0, R5;

Добавляют байты R0 к соответствующему байту в R5 и записях; разделенный на два результат к соответствующему байту в R4.

3.5.19 UHASX и UHSAX

Сокращение вдвое без знака, добавляет и вычитает с обменом и сокращением вдвое без знака, вычитают и добавляют с обменом.

Syntax

op{cond} {Rd}, Rn, Rm

где:

- op - один из:

UHASX: Добавьте и вычитите с обменом и сокращением вдвое.

UHSAX: Вычитите и добавьте с обменом и сокращением вдвое.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр
- 'Rn' 'Rm' являются регистрами, содержащими первые и вторые операнды

Работа

Инструкция UHASX:

1. Добавляет главное сокращение первого операнда с нижним сокращением второго операнда.
2. Смещает результат на один бит к праву, вызывающему деление на два, или сокращение вдвое.
3. Пишет результат сокращения в дополнении к главному сокращению у целевого регистра.
4. Вычитает главное сокращение второго операнда от нижней части целого слова первого операнда.
5. Смещает результат на один бит к праву, вызывающему деление на два, или сокращение вдвое.
6. Результат сокращения записывается в нижнем сокращении целевого регистра.

Инструкция UHSAX:

1. Вычитает нижнее сокращение второго операнда от вершины целого слова первого операнда.
2. Смещает результат на один бит к праву, вызывающему деление на два, или сокращение вдвое.
3. Результат сокращения записывается в главном сокращении целевого регистра.
4. Добавляет нижнее сокращение первого операнда с главным сокращением второго операнда.
5. Смещает результат на один бит к праву, вызывающему деление на два, или сокращение вдвое.
6. Результат сокращений записывают к нижнему сокращению целевого регистра.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не влияют на флаги кода условия.

Примеры

UHASX R7, R4, R2;

Добавляет главное сокращение R4 с нижним сокращением R2 ; и записи разделили на два результат превысить сокращение R7 ; Вычитает главное сокращение R2 от нижнего сокращения ; R7 и записи разделили на два результат записывают в сокращении R7

UHSAX R0, R3, R5;

Вычитает нижнее сокращение R5 от главного сокращения; R3 и записи разделили на два результат превысить сокращения R0 Добавляет главное сокращение R5, записывают R3 и записи разделили на два результат записывают в R0.

3.5.20 UNSUB16 и UNSUB8

Сокращение вдвое без знака вычитает 16, и сокращение вдвое Без знака вычитают 8

Syntax

op{cond}{Rd,} Rn, Rm

где:

- op - любой из:

UNSUB16:

Выполняет два 16-разрядных целочисленных дополнения без знака, половины результаты, и пишет результаты в целевой регистр.

UNSUB8:

Выполняет четыре 8-разрядных целочисленных дополнения без знака, половины результаты, и пишет результаты в целевой регистр.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр
- 'Rn' - первый регистр, содержащий операнд
- 'Rm' - второй регистр, содержащий операнд

Работа

Используйте эти инструкции, чтобы добавить 16-разрядные и 8-разрядные данные и затем разделить на два результат прежде, чем записать результат в целевой регистр:

Инструкция UNSUB16:

1. Вычитает каждое полуслово второго операнда от соответствующего полуслова первого операнда.
2. Перестановки каждый результат полуслова вправо на один бит, деля на два данные.
3. Записи каждый результат полуслова без знака к соответствующему полуслову в целевом регистре.

Инструкция UNSUB8:

1. Вычитает каждый байт второго операнда от соответствующего байта первого операнда.
2. Перестановки каждый результат байта на один бит вправо, деля на два данные.
3. Пишут результаты байта без знака в соответствующий байт целевого регистра.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не изменяют флаги.

Примеры

UNSUB16 R1, R0; Вычитает полуслова в R0 от соответствующего полуслова R1 ;

и записи разделили на два результат к соответствующему полуслову в R1

UNSUB8 R4, R0, R5; Вычитает байты R5 от соответствующего байта в R0 и ;

записи разделили на два результат к соответствующему байту в R4.

3.5.21 SEL

Выберите байты. Выбирает каждый байт его результата или его первый операнд или его второй операнд, согласно значениям флагов GE.

Syntax

SEL{<c>}{<q>} {<Rd>}, <Rn>, <Rm>

где:

- <c>, <q> - стандартные ассемблерные поля синтаксиса.
- <Rd> - целевой регистр.
- <rn> - первый регистр операнда.
- <Rm> - второй регистр операнда.

Работа

Инструкция SEL:

1. Читает значение каждого бита APSR.GE.
2. В зависимости от значения APSR.GE, присваивается, место назначения регистрируют значение или первого или второго регистра операнда.

Ограничения

Ничего.

Флаги условия

Эти инструкции не изменяют флаги.

Примеры

SADD16 R0, R1, R2; биты GE Набора на основе результата

SEL R0, R0, R3; Выберите байты из R0 или R3, на основе GE.

3.5.22 USAD8

Сумма без знака абсолютных разностей

Syntax

USAD8{cond}{Rd,} Rn, Rm

где:

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр
- 'Rn' - первый регистр операнда
- 'Rm' - второй регистр операнда

Работа

Инструкция USAD8:

1. Вычитает каждый байт второго регистра операнда от соответствующего байта первого регистра операнда.
2. Добавляют абсолютные значения различий вместе.
3. Пишет результат в целевой регистр.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не изменяют флаги.

Примеры

USAD8 R1, R4, R0; Вычитает каждый байт в R0 от соответствующего байта; R4 добавляет различия и пишет в R1

USAD8 R0, R5; Вычитает байты R5 от соответствующего байта в R0; добавляют различия и пишут в R0.

3.5.23 USADA8

Сумма без знака абсолютных разностей и накопления

Syntax

USADA8{cond}{Rd,} Rn, Rm, Ra

где:

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'rd.' - целевой регистр
- 'Rn' - первый регистр операнда
- 'Rm' - второй регистр операнда
- 'Ra' - регистр, который содержит значение накопления.

Работа

Инструкция USADA8:

1. Вычитает каждый байт второго регистра операнда от соответствующего байта первого регистра операнда.
2. Добавляют абсолютные разности без знака вместе.
3. Добавляет, что накопление оценивает сумме абсолютных разностей.
4. Пишет результат в целевой регистр.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не изменяют флаги.

Примеры

USADA8 R1, R0, R6;

Вычитает байты в R0 от соответствующего сокращения R1; добавляют различия, увеличивает значение R6, пишет в R1

USADA8 R4, R0, R5, R2;

Вычитает байты R5 от соответствующего байта в R0; добавляют различия, увеличивает значение R2, пишет в R4.

3.5.24 USUB16 и USUB8

вычитают 16, и вычитают 8

Syntax

op{cond}{Rd,} Rn, Rm

где:

- op - любой из:

USUB16: вычитают 16.

USUB8: вычитают 8.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр
- 'Rn' - регистр, содержащий операнд
- 'Rm' - второй регистр операнда

Работа

Используйте эти инструкции, чтобы вычесть 16-разрядные и 8-разрядные данные прежде, чем записать результат в целевой регистр:

Инструкция USUB16:

1. Вычитает каждое полуслово из второго регистра операнда от соответствующего полуслова первого регистра операнда.
2. Пишет результат без знака в соответствующих полусловах целевого регистра.

Инструкция USUB8:

1. Вычитает каждый байт второго регистра операнда от соответствующего байта первого регистра операнда.
2. Пишет результат байта без знака в соответствующем байте целевого регистра.

Ограничения

Не используйте SP или PC.

Флаги условия

Эти инструкции не изменяют флаги.

Примеры

USUB16 R1, R0; Вычитает полуслова в R0 от соответствующего полуслова; R1 и записи к соответствующему полуслову в R1

USUB8 R4, R0, R5; Вычитает байты R5 от соответствующего байта в R0 и; записи к соответствующему байту в R4.

3.6

Таблица умножения и деления

Сокращение	Краткое описание	Вид
MLA	Умножение с суммированием, 32-битный результат	MUL, MLA, и MLS на стр 109
MLS	Умножает и вычитает 32-бит результат(32-разрядг)	MUL, MLA, and MLS на стр 109
MUL	Умножает, 32 разрд результат	MUL, MLA, and MLS на стр 109
SDIV	Со знаком делит	SDIV and UDIV нас тр 123
SMAL[B,T]	Умножает с суммирован(сокращ)	SMLA and SMLAW на стр 111
SMLAD, SMLADX	Со знаком суммирует умножает в двое	SMLAD на стр 113
SmLAL	Умножается и суммируется (32*23->64) 64 битный результат	SMLAL and SMLALD на стр 114
SMLAL[B,T]	Суммируется при умножении (сокращается)	SMLAL and SMLALD на стр 114
SMLALD, SMLALDX	Со знаком умножает вдвое, суммируется	SMLAL and SMLALD на стр 114
SMLAW[B,T]	Со знаком умножает, суммируется	SMLA and SMLAW на стр 111
SMLSd	Со знаком умножает, двойное вычитание	SMLSd and SMLSd на стр 116
SMLSd	Со знаком умножает, двойное вычитание	SMLSd and SMLSd на стр 116
SMMLA	Старшее значащее слово со знаком умножают, суммируются	SMMLA and SMMLS на стр 118
SMMLS, SMMLSR	Умножает старшее значение, умножает со знаком, вычитает	SMMLA and SMMLS на стр 118
SMUAD, SMUADX	Со знаком двойное умножение, суммирование	SMUAD and SMUSD на стр 120
SMUL[B,T]	Умножает со знаком	SMUL and SMULW на стр 121
SMMUL, SMMULR	Умножает старшее значение со знаком	SMMUL на стр 119
SMULL	Умножает со знаком 64- битный результат	SMMUL на стр 119
SMULWB, SMULWT	Умножает со знаком	SMUL and SMULW на стр 121
SMUSD, SMUSDx	Со знаком двойное умножение, вычитание	SMUAD and SMUSD на стр 120
UDIV	Деление без знака	SMLA and SMLAW на стр 111
UMAAL	Суммирует умножение 64-битный результат	UMULL, UMAAL and UMLAL на стр 110
UMLAL	Суммирует умножение 64-битный результат	UMULL, UMAAL and UMLAL на стр 110
UMULL	Умножение без знака 64-битный результат	UMULL, UMAAL and UMLAL на стр 110

3.6.1 MUL, MLA и MLS

Умножение, умножение с суммированием и умножением с вычитанием, используя 32-разрядные операнды, и приводя к 32-разрядному результату.

Syntax

MUL {S} {cond} {Rd.}, Rn, Rm; умножение

MLA {cond} Rd., Rn, Rm, Ra; Умножение с суммированием

MLS {cond} Rd., Rn, Rm, Ra; Умножение с вычитанием

где:

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'S' - дополнительный суффикс. Если S определен, флаги кода условия обновлены на результате работы (см. Условное выполнение на странице 64).
- 'Rd.' - целевой регистр. Если Rd. опущена, целевой регистр - Rn
- 'Rn', 'Rm' - регистры, содержащие значения, которые будут умножены
- 'Ra' - регистр, содержащий значение, которое будет добавлено к или вычтено из

Работа

Инструкция MUL умножает значения от Rn и Rm, и помещает младшие значащие 32 бита результата в Rd.

Инструкция MLA умножает значения от Rn и Rm, добавляет значение от Ra и помещает младшие значащие 32 бита результата в Rd.

Инструкция MLS умножает значения от Rn и Rm, вычитает продукт из значения от Ra и помещает младшие значащие 32 бита результата в Rd.

Результаты не зависят от того, подписаны ли операнды или без знака.

Ограничения

В этих инструкциях не используйте SP и не используйте PC.

Если Вы используете суффикс S с инструкцией MUL:

- Rd., Rn и Rm должны все быть в диапазоне R0 к R7
- Rd. должна совпасть с Rm
- Вы не должны использовать суффикс cond

Флаги условия

Если S определен, инструкция MUL:

- Обновляет флаги N и Z согласно результату
- Не влияет на C и V флагов

Примеры

MUL R10, R2, R5; умножение, $R10 = R2 \times R5$

MLA R10, R2, R1, R5; умножение с суммированием, $R10 = (R2 \times R1) + R5$

MULS R0, R2, R2; умножением с обновлением флага, $R0 = R2 \times R2$

MULLT R2, R3, R2; условно умножение, $R2 = R3 \times R2$

MLS R4, R5, R6, R7; умножение с вычитанием, $R4 = R7 - (R5 \times R6)$

3.6.2 UMULL, UMAAL и UMLAL

Без знака умножение, с дополнительным суммированием, 32-разрядные операнды, приводя к 64-разрядному результату.

Syntax

op {cond} RdLo, RdHi, Rn, Rm

где:

- 'op' - один из:

UMULL: Без знака умножение

UMAAL: Без знака умножаются, с суммированием, суммируют UMLAL: Без знака умножаются, с суммированием

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'RdHi, RdLo' являются целевыми регистрами. Они также содержат суммирующимся значение.
- 'Rn,Rm' является регистрами, содержащими первые и вторые операнды

Работа

Инструкция UMULL:

1. Умножает без знака в первых и вторых операндах.
2. Пишут младшие значащие 32 бита результата в RdLo.
3. Пишут старшие значащие 32 бита результата в RdHi.

Инструкция UMAAL:

1. Умножает два 32-разрядных целых числа без знака в первых и вторых операндах.
2. Добавляет 32-разрядное целое число без знака в RdHi к 64-разрядному результату умножения.
3. Добавляет 32-разрядное целое число без знака в RdLo к 64-разрядному результату дополнения.
4. Пишут лучшие 32 бита результата к RdHi.
5. Пишут более низкие 32 бита результата к RdLo.

Инструкция UMLAL:

1. Умножает без знака в первых и вторых операндах.
2. Добавляет 64-разрядный результат к 64-разрядному целому без знака, содержавшемуся в RdHi и RdLo.
3. Записывает результат обратно к RdHi и RdLo.

Ограничения

В этих инструкциях:

4.6.1 Не используйте или SP или PC

4.6.2 RdHi и RdLo должны быть различными регистрами

Флаги условия

Эти инструкции не влияют на флаги кода условия.

Примеры

UMULL R0, R4, R5, R6; Умножает R5 и R6, пишет лучшие 32 бита в R4
; и нижняя часть 32 бита к R0

UMAAL R3, R6, R2, R7; Умножает R2 и R7, добавляет R6, добавляет R3, пишет
; лучшие 32 бита к R6 и нижняя часть 32 бита к R3

UMLAL R2, R1, R3, R5; Умножает R5 и R3, добавляет R1:R2, пишет в R1:R2.

3.6.3 SMLA и SMLAW

Умножение, суммирование (сокращения)

Syntax

op {XY} {cond} Rd., Rn, Rm

op {Y} {cond} Rd., Rn, Rm, Ra

где

- op - один из:

SMLA: Умножение, суммирование (сокращения), X и Y определяет, какая половина источника регистрирует Rn, и Rm используются в качестве первого, и вторые умножают операнд.

– Если X B, то нижнее полуслово, биты [15:0], Rn используется.

– Если X T, то главное полуслово, биты [31:16], Rn используется.

– Если Y - B, то нижнее полуслово, биты [15:0], Rm используется.

– Если Y - T, то главное полуслово, биты [31:16], Rm используется.

SMLAW: умножаются, накапливаются (слово сокращение). Y определяет, какая половина источника регистрирует, Rm используется, поскольку вторые умножают операнд.

– Если Y - T, то главное полуслово, биты [31:16] Комнаты используются.

– Если Y - B, то нижнее полуслово, биты [15:0] Комнаты используются.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр. Если Rd. опущена, целевой регистр - Rn
- 'Rn', 'Rm' - регистры, содержащие значения, которые будут умножены
- 'Ra' - регистр, содержащий значение, которое будет добавлено к или вычтено из

Работа

SMALBB, SMLABT, SMLATB, инструкции SMLATT:

1. Умножьте указанное полуслово со знаком, вершину или нижнюю часть, значения от Rn и Rm.
2. Добавьте значение в Ra к получающемуся 32-разрядному продукту.
3. Запишите результат умножения и дополнения в Rd.
4. Неуказанные полуслова исходных регистров проигнорированы.

SMLAWB и инструкции SMLAWT:

1. Умножьте 32-разрядные значения со знаком в Rn с:
 - а) Вершина подписала сокращение Rm, T суффикс инструкции.
 - б) Нижняя часть подписала сокращение Rm, B суффикс инструкции.
2. Добавьте 32-разрядное значение со знаком в Ra к лучшим 32 битам 48-разрядного продукта
3. Запишите результат умножения и дополнения в Rd.
4. Нижняя часть 16 битов 48-разрядного продукта проигнорирована.

5. Если переполнение происходит во время добавления накапливать значения, системы команд флаг Q в APSR. Никакое переполнение не может произойти во время умножения.

Ограничения

В этих инструкциях не используйте SP или PC.

Флаги условия

Если переполнение обнаружено, флаг Q установлен.

Примеры

SMLABB R5, R6, R4, R1; Умножает нижние полуслова R6 и R4, добавляет
; R1 и записи к R5

SMLATB R5, R6, R4, R1; Умножает главное полуслово R6 с нижним полусловом
; из R4 добавляет R1 и пишет в R5

SMLATT R5, R6, R4, R1; Умножает главные полуслова R6 и R4, добавляет
; R1 и записи сумма к R5

SMLABT R5, R6, R4, R1; Умножает нижнее полуслово R6 с главным полусловом
; из R4 добавляет R1 и пишет в R5

SMLABT R4, R3, R2; Умножает нижнее полуслово R4 с главным полусловом
; из R3 добавляет R2 и пишет в R4

SMLAWB R10, R2, R5, R3; Умножает R2 с нижним полусловом R5, добавляет
; R3 к результату и записям лучшие 32 бита к R10

SMLAWT R10, R2, R1, R5; Умножает R2 с главным полусловом R1, добавляет R5
; и пишет лучшие 32 бита в R10.

3.6.4 SMLAD

Умножает вдвое с накоплением

Syntax

op {X} {cond} Rd., Rn, Rm, Ra;

где:

- op - один из:

SMLAD: Умножает вдвое с накоплением

SMLADX: Умножает вдвое с накоплением. X определяет, какое сокращение источника регистрируют, Rn используется в качестве множителя операнда.

Если X опущен, умножение - нижняя часть × нижняя часть и вершина × вершина. Если X присутствует, умножение - нижняя часть × вершина и вершина × нижняя часть.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'Rn' - первый регистр операнда, содержащий значения, которые будут умножены
- 'Rm' - второй регистр операнда
- 'Ra' - накапливает значение

Работа

SMLAD и инструкции SMLADX расценивают эти два операнда как четыре полуслова 16-разрядные значения. SMLAD и инструкции SMLADX:

1. Также:

а) Если X не присутствует, умножить, вершина подписалась, значение полуслова в Rn с вершиной подписало полуслово Rm, и нижняя часть подписалась, значения полуслова в Rn с нижней частью подписали полуслово Комнаты.

а) Если X присутствует, умножить, вершина подписалась, значение полуслова в Rn с нижней частью подписало полуслово Rm, и нижняя часть подписалась, значения полуслова в Rn с вершиной подписали полуслово Rm.

2. Добавьте оба результата умножения к 32-разрядному значению со знаком в Ra.

3. Запишите 32-разрядный результат со знаком умножения и дополнения к Rd.

Ограничения

Не используйте SP или PC.

Флаги условия

Эти инструкции не изменяют флаги.

Примеры

SMLAD R10, R2, R1, R5; Умножает два значения полуслова в R2 с соответствующие полуслова в R1, добавляют R5 и записи к R10

SMLALDX R0, R2, R4, R6; Умножает главное полуслово R2 с нижним полусловом из R4, умножает нижнее полуслово R2 с вершиной полуслово R4, добавляет R6 и пишет в R0.

3.6.5 SMLAL и SMLALD

Умножение с долгим накоплением(сохранением),накопление сокращения,умножают с вычитанием

Syntax

op{cond} RdLo, RdHi, Rn, Rm

op{XY}{cond} RdLo, RdHi, Rn, Rm

op{X}{cond} RdLo, RdHi, Rn, Rm

где:

- op - один из:

– SMLAL: умножаются, накапливаются долго

– SMLAL: умножаются, накапливаются долго (полуслова, X и Y). X и Y определяют, какое полуслово источника регистрирует Rn, и Rm используются в качестве первого, и вторые умножают операнд: Если X B, то нижнее полуслово, биты [15:0], Rn используется. Если X T, то главное полуслово, биты [31:16], Rn используется.

Если Y - B, то нижнее полуслово, биты [15:0], Rm используется. Если Y - T, то главное полуслово, биты [31:16], Rm используется.

– SMLALD: умножаются, накапливаются долго двойной

– SMLALDX: умножаются, накапливаются долго двойной инвертированный

Если эти X опущены, умножение - нижняя часть × нижняя часть и вершина × вершина. Если X присутствует, умножение - нижняя часть × вершина и вершина × нижняя часть.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'RdHi, RdLo' являются целевыми регистрами. RdLo составляет более низкие 32 бита, и RdHi составляет верхние 32 бита 64-разрядного целого числа. Для SMLAL, SMLALBB, SMLALBT, SMLALTB, SMLALTT, SMLALD и SMLALDX, они также содержат накапливающееся значение.
- 'Rn', 'Rm' - регистры, содержащие первые и вторые операнды

Работа

Инструкция SMLAL:

1. Умножается дополнение значения слова от Rn и Rm.
2. Добавляет 64-разрядное значение в RdLo и RdHi к получающемуся 64-разрядному продукту.
3. Пишет 64-разрядный результат умножения и дополнения в RdLo и RdHi.

SMLALBB, SMLALBT, SMLALTB и инструкции SMLALTT:

1. Умножает указанное полуслово со знаком, вершину или нижнюю часть, значения от Rn и Rm.
2. Добавляет получающийся расширенный до знака 32-разрядный продукт к 64-разрядному значению в RdLo и RdHi.

3. Пишет 64-разрядный результат умножения и дополнения в RdLo и RdHi.

Неуказанные полуслова исходных регистров проигнорированы.

SMLALD и инструкции SMLALDX интерпретируют значения от Rn и Rm, поскольку четыре дополнения второго полуслова подписали 16-разрядные целые числа. Эти инструкции:

- Если X не присутствует, умножьтесь, вершина подписалась, значение полуслова Rn с вершиной подписало полуслово Rm, и нижняя часть подписалась, значения полуслова Rn с нижней частью подписали полуслово Rm.
- Или если X присутствует, умножьтесь, вершина подписалась, значение полуслова Rn с нижней частью подписало полуслово Rm, и нижняя часть подписалась, значения Rn с вершиной подписали Rm.
- Добавьте два результата умножения к 64-разрядному значению со знаком в RdLo и RdHi, чтобы создать получающийся 64-разрядный продукт.
- Запишите 64-разрядный продукт в RdLo и RdHi.

Ограничения

В этих инструкциях:

Не используйте SP и не используйте PC.

RdHi и RdLo должны быть различными регистрами.

Флаги условия

Эти инструкции не влияют на флаги кода условия.

Примеры

SMLAL R4, R5, R3, R8; Умножает R3 и R8, добавляет R5:R4 и пишет в ; R5:R4

SMLALBT R2, R1, R6, R7; Умножает нижнее полуслово R6 с вершиной полуслова R7, знак расширяется на 32-разрядный, добавляет R1:R2 и записи к R1:R2 SMLALTB R2, R1, R6, R7; Умножает главное полуслово R6 с нижней частью полуслова R7, знак расширяется на 32-разрядный, добавляет R1:R2 и записи к R1:R2

SMLALD R6, R8, R5, R1; Умножает главные полуслова в R5 и R1 и нижней части полуслова R5 и R1, добавляет R8:R6 и пишет в R8:R6 SMLALDX R6, R8, R5, R1; Умножает главное полуслово в R5 с нижней частью полуслова R1 и нижнее полуслово R5 с главное полуслово R1, добавляет R8:R6 и пишет в R8:R6.

3.6.6 SMLSD и SMLSLD

умножаются, двойное вычитание, и Со знаком умножаются

Syntax

op{X}{cond} Rd, Rn, Rm, Ra

где:

- op - один из:

SMLSD: умножаются, вычитают двойной. SMLSDX: Подписанный умножаются, вычитают двойной, инвертировал

SMLSLD: Подписанный умножаются, вычитают долго двойной.

SMLSLDX: Подписанный умножаются, вычитают двойной инвертированный.

– Если X присутствует, умножение - нижняя часть × вершина и вершина × нижняя часть.

– Если эти X опущены, умножение - нижняя часть × нижняя часть и вершина × вершина.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'Rn', 'Rm' - регистры, содержащие первые и вторые операнды
- 'Ra' - регистр, содержащий накапливать значение

Работа

Инструкция SMLSD интерпретирует значения от первых и вторых операндов как четыре полуслова со знаком. Эта инструкция:

1. Дополнительно поворачивает полуслова второго операнда.
2. Выступает два 16 × 16-разрядного умножения полуслова.
3. Вычитает результат верхнего умножения полуслова от результата более низкого умножения полуслова.
4. Добавляет, что со знаком накапливают значение до результата вычитания.
5. Пишет результат дополнения к целевому регистру.

Инструкция SMLSLD интерпретирует значения от Rn и Rm как четыре полуслова со знаком. Эта инструкция:

1. Дополнительно поворачивает полуслова второго операнда.
2. Выступает два, подписал 16 × 16-разрядного умножения полуслова.
3. Вычитает результат верхнего умножения полуслова от результата более низкого умножения полуслова.
4. Добавляет 64-разрядное значение в RdHi и RdLo к результату вычитания.
5. Пишет 64-разрядный результат дополнения к RdHi и RdLo.

Ограничения

В этих инструкциях: не используйте SP или PC. Флаги условия

Это системы команд флаг Q, если накапливать работа переполняется. Переполнение не может произойти во время умножения или вычитания.

Для системы команд Ползунка эти инструкции не влияют на флаги кода условия.

Примеры

SMLS R0, R4, R5, R6; Умножает нижнее полуслово R4 с нижней частью ; полуслово R5, умножает главное полуслово R4 ; с главным полусловом R5, вычитает второй из ; во-первых, добавляет R6, пишет в R0

SMLS DX R1, R3, R2, R0; Умножает нижнее полуслово R3 с вершиной ; полуслово R2, умножает главное полуслово R3 ; с нижним полусловом R2, вычитает второй из ; во-первых, добавляет R0, пишет в R1

SMLS LD R3, R6, R2, R7; Умножает нижнее полуслово R6 с нижней частью полуслово R2, умножает главное полуслово R6 с главным полусловом R2, вычитает второй из во-первых, добавляет R6:R3, пишет в R6:R3

SMLS LD X R3, R6, R2, R7; Умножает нижнее полуслово R6 с вершиной полуслово R2, умножает главное полуслово R6 с нижним полусловом R2, вычитает второй из во-первых, добавляет R6:R3, пишет в R6:R3.

3.6.7 SMMLA и SMMLS

Старшее значащее слово со знаком умножается, накапливаются / умножаются, вычитают
Syntax

op{R}{cond} Rd, Rn, Rm, Ra

где:

- op - один из:

SMMLA: старшее значащее слово Со знаком умножается, накапливаются. SMMLS: старшее значащее слово Со знаком умножается, вычитают.

- R - флаг погрешности округления. Если R определен, результат округлен вместо того, чтобы быть усеченным, 0x80000000 добавлен к продукту, прежде чем высокое слово будет извлечено.
- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'Rn', 'Rm' - регистры, содержащие первые, и вторые умножения операции
- 'Ra' - регистр, содержащий накапливать значение

Работа

Инструкция SMMLA интерпретирует значения от Rn и Rm как подписанные 32-разрядные слова:

1. Умножает значения в Rn и Rm.
2. Дополнительно округляет результат, добавляя 0x80000000.
3. Извлекает старшие значащие 32 бита результата.
4. Добавляет значение Ra к извлеченному значению со знаком.
5. Пишет результат дополнения в Ул.

Инструкция SMMLS интерпретирует значения от Rn и Rm как подписанные 32-разрядные слова:

1. Умножает значения в Rn и Rm.
2. Дополнительно округляет результат, добавляя 0x80000000.
3. Извлекает старшие значащие 32 бита результата. 4. Вычитает извлеченное значение результата значения в Ra.
5. Пишет результат вычитания в Rd.

Ограничения

В этих инструкциях: не используйте SP или PC.

Флаги условия

Эти инструкции не влияют на флаги кода условия.

Примеры

SMMLA R0, R4, R5, R6; Умножает R4 и R5, лучшие 32 бита извлечений, ; добавляет R6, усекает и пишет в R0

SMMLAR R6, R2, R1, R4; Умножает R2 и R1, лучшие 32 бита извлечений, ; добавляют R4, раунды и пишут в R6 SMMLSR R3, R6, R2, R7; Умножает R6 и R2, лучшие 32 бита извлечений, ; вычитает R7, раунды и пишет в R3 .

SMMLS R4, R5, R3, R8; Умножает R5 и R3, лучшие 32 бита извлечений, ; вычитает R8, усекает и пишет в R4

3.6.8 SMMUL

Старшее значащее слово со знаком умножается

Syntax

op{R}{cond} Rd, Rn, Rm

где:

- op - один из:

SMMUL: старшее значащее слово со знаком умножается

R: флаг погрешности округления. Если R определен, результат округлен вместо того, чтобы быть усеченным. В этом случае постоянный 0x80000000 добавлен к продукту, прежде чем высокое слово будет извлечено.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'Rn', 'Rm' - регистры, содержащие первые и вторые операнды

Работа

Инструкция SMMUL интерпретирует значения от Rn и Rm как дополнительные 32-разрядные целые числа со знаком . Инструкция SMMUL:

1. Умножает значения от Rn и Rm.
2. Дополнительно округляет результат, иначе усекает результат.
3. Пишут старшие значащие 32 бита со знаком результата в Ул.

Ограничения

В этой инструкции: не используйте SP или PC.

Флаги условия

Эта инструкция не влияет на флаги кода условия.

Примеры

SMULL R0, R4, R5; Умножает R4 и R5, усекает лучшие 32 бита ; и записи к R0

SMULLR R6, R2; Умножает R6 и R2, округляет лучшие 32 бита ; и записи к R6.

3.6.9 SMUAD и SMUSD

Со знаком двойное умножение, суммирование, и Со знаком двойное умножение, вычитают

Syntax

op{X}{cond} Rd, Rn, Rm

где:

- op - один из:

SMUAD: Со знаком двойной умножение, добавляют. SMUADX: Со знаком двойной умножение, добавляют инвертированный. SMUSD: Со знаком двойной умножение, вычитают.

SMUSDX: Со знаком двойное умножение, вычитают инвертированный.

– Если X присутствует, умножение - нижняя часть × вершина и вершина × нижняя часть. Если эти X опущены, умножение - нижняя часть × нижняя часть и вершина × вершина.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'Rn', 'Комната' - регистры, содержащие первые и вторые операнды

Работа

SMUAD интерпретирует первые и вторые значения операнда как два полуслова со знаком:

1. Дополнительно поворачивает полуслова второго операнда.
2. Выступает два 16 × 16-разрядного умножения.
3. Добавляет, что два умножения заканчивается вместе.
4. Пишет результат дополнения к целевому регистру.

SMUSD интерпретирует значения от первых и вторых операндов как дополнительные целые числа со знаком :

1. Дополнительно поворачивает полуслова второго операнда.
2. Выступает два 16 × 16-разрядного умножения.
3. Вычитает результат главного умножения полуслова от результата нижнего умножения полуслова.
4. Пишет результат вычитания к целевому регистру.

Ограничения

В этих инструкциях: не используйте SP или PC.

Флаги условия

Устанавливает флаг Q, если дополнение переполняется. Умножение не может переполниться.

Примеры

SMUAD R0, R4, R5; Умножает нижнее полуслово R4 с нижней частью полуслова R5, добавляет умножение главного полуслова из R4 с главным полусловом R5, пишет в R0
SMUADX R3, R7, R4; Умножает нижнее полуслово R7 с главным полусловом ; из R4 добавляет умножение главного полуслова R7 ; с нижним полусловом R4, пишет в R3
SMUSD

R3, R6, R2; Умножает нижнее полуслово R4 с нижним полусловом ; из R6, вычитает умножение главного полуслова R6 ; с главным полусловом R3, пишет в R3

SMUSDX R4, R5, R3; Умножает нижнее полуслово R5 с главным полусловом R3, вычитает умножение главного полуслова R5 с нижним полусловом R3, пишет в R4.

3.6.10 SMUL и SMULW

умножаются (полуслова), и Со знаком умножаются (слово полусловом)

Syntax

op{XY}{cond} Rd, Rn, Rm

op{Y}{cond} Rd. Rn, Rm

- op - один из:

SMUL {XY} Со знаком умножаются (полуслова). X и Y определяют, какое полуслово источника регистрирует Rn, и Rm используется в качестве первого, и вторые умножают операнд.

Если X - B, то нижнее полуслово, биты [15:0] Rn используются. Если X - T, то главное полуслово, биты [31:16] Rn используются.

Если Y - B, то нижнее полуслово, биты [15:0], Комнаты используется. Если Y - T, то главное полуслово, биты [31:16], Rm используется.

SMULW {Y} Со знаком умножаются (слово полусловом). Y определяет, какое полуслово источника регистрируют, Rm используется, поскольку вторые умножают операнд.

Если Y - B, то нижнее полуслово (биты [15:0]) Rm используется. Если Y - T, то главное полуслово (биты [31:16]) Rm используется.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'Rn', 'Rm' - регистры, содержащие первые и вторые операнды

Работа

SMULBB SMULTB, SMULBT и инструкции SMULTT интерпретируют значения от Rn и Rm как четыре 16-разрядных целых числа со знаком. Эти инструкции:

1. Умножьте указанное полуслово со знаком, вершину или нижнюю часть, значения от Rn и Rm.
2. Запишите 32-разрядный результат умножения в Rd.

SMULWT и инструкции SMULWB интерпретируют значения от Rn как 32-разрядное целое число со знаком и Rm как два полуслова 16-разрядные целые числа со знаком. Эти инструкции:

1. Умножьте первый операнд и вершину, T суффикс или нижняя часть, B суффикс, полуслово второго операнда.

2. Запишите 32 старших значащих бита со знаком 48-разрядного результата в целевом регистре.

Ограничения

Не используйте SP и не используйте PC.

Примеры

SMULBT R0, R4, R5; Умножает нижнее полуслово R4 с вершиной ; полуслово R5, умножает результаты и пишет в R0

SMULBB R0, R4, R5; Умножает нижнее полуслово R4 с нижней частью ; полуслово R5, умножает результаты и пишет в R0

SMULTT R0, R4, R5; Умножает главное полуслово R4 с вершиной ; полуслово R5, умножает результаты и пишет в R0

Ред. 4 121/245 DocID022708

Система команд Кортис-М4 STM32 PM0214

SMULTB R0, R4, R5; Умножает главное полуслово R4 с нижней частью ; полуслово R5, умножает результаты и пишет в R0

SMULWT R4, R5, R3; Умножает R5 с главным полусловом R3, ; извлекает лучшие 32 бита и пишет в R4

SMULWB R4, R5, R3; Умножает R5 с нижним полусловом R3, извлекает лучшие 32 бита и пишет в R4.

3.6.11 UMULL, UMLAL, SMULL и SMLAL

Без знака умножаются, Накапливаются, используя 32-разрядные операнды и приводя к 64-разрядному результату.

Syntax

op{cond} RdLo, RdHi, Rn, Rm

где:

- op' является одним из:

UMULL: Без знака умножаются.

UMLAL: Без знака умножаются, и накапливаются. SMULL: Подписываемый долго умножаются.

SMLAL: Подписываемый долго умножаются и накапливаются.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'RdHi, RdLo' являются целевыми регистрами. Для UMLAL и SMLAL они также содержат накапливающееся значение.
- 'Rn', 'Rm' - регистры, содержащие операнды

Работа

Инструкция UMULL интерпретирует значения от Rn и Rm как целые без знака. Это умножает эти целые числа и помещает младшие значащие 32 бита результата в RdLo и старшие значащие 32 бита результата в RdHi.

Инструкция UMLAL интерпретирует значения от Rn и Rm как целые без знака. Это умножает эти целые числа, добавляет 64 разрядных результата к 64-разрядному целому без знака, содержащемуся в RdHi и RdLo, и записывают результат обратно к RdHi и RdLo.

Инструкция SMULL интерпретирует значения от Rn и Rm как дополнительные целые числа со знаком. Это умножает эти целые числа и помещает младшие значащие 32 бита результата в RdLo и старшие значащие 32 бита результата в RdHi.

Инструкция SMLAL интерпретирует значения от Rn и Rm как дополнительные целые числа со знаком. Это умножает эти целые числа, добавляет 64-разрядный результат к целому числу со знаком на 64-биты, содержавшемуся в RdHi и RdLo, и записывает результат обратно к RdHi и RdLo.

Ограничения

В этих инструкциях:

- Не используйте SP или PC
- RdHi и RdLo должны быть различными регистрами.

Флаги условия

Эти инструкции не влияют на флаги кода условия.

Ред. 4 122/245 DocID022708

PM0214 система команд Кору-М4 STM32

Примеры

UMULL R0, R4, R5, R6; Без знака $(R4, R0) = R5 \times R6$

SMLAL R4, R5, R3, R8; Подписанный $(R5, R4) = (R5, R4) + R3 \times R8$

3.6.12 SDIV и UDIV

делятся и делаж без знака.

Syntax

SDIV{cond} {Rd,} Rn, Rm

UDIV{cond} {Rd,} Rn, Rm

где:

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр. Если Rd. опущена, целевой регистр - Rn
- 'Rn', регистр, содержащий значение, которое будет разделено
- 'Rm' - регистр, содержащий делитель

Работа

SDIV выполняет подразделение целого числа со знаком значения в Rn значением в Rm. UDIV выполняет подразделение целого без знака значения в Rn значением в Rm.

Для обеих инструкций, если значение в Rn не делимое значением в Rm, результат округлен по направлению к нулю.

Ограничения

Не используйте или SP или PC.

Флаги условия

Эти инструкции не изменяют флаги.

Примеры

SDIV R0, R2, R4; подписанный делятся, $R0 = R2/R4$

UDIV R8, R8, R1; делаж без знака, $R8 = R8/R1$

3.7 таблица

В этом разделе описываются инструкции насыщения

SSAT	Насыщает со знаком	SSAT и USAT на стр 125
SSAT16	Насыщают сокращение со знаком	SSAT16 и USAT16 на стр 126
USAT	Насыщает без знака	SSAT и USAT на стр 125
USAT16	Насыщают сокращение без знака	SSAT16 и USAT16 на стр 126
QADD	Суммирует, вычисляет	QADD и QSUB на стр 127
QSUB	Вычитает	QADD и QSUB на стр 127
QSUB16	Суммирует и вычисляет с обменом	QADD и QSUB на стр 127
QASX	Суммирует и вычисляет с обменом	QASX и QSAX на стр 128
QSAX	Умножение двойное и суммирование	QASX и QSAX на стр 128
QDADD		QDADD и QDSUB на стр 129
QDSUB	Насыщает без знака и суммирует	QDADD и QDSUB на стр 129
UQADD16	Насыщает без знака и суммирует	UQADD и UQSUB на стр 131
UQADD8	Насыщает без знака, вычитает и суммирует с обменом	UQADD и UQSUB на стр 131
UQSAX	Насыщает без знака	UQASX и UQSAX на стр 130
UQSUB16	Насыщает без знака, вычитает и суммирует с обменом	UQADD и UQSUB на стр 131
UQSUB8	Насыщает без знака, вычитает и суммирует с обменом	UQADD и UQSUB на стр 131

Для n-bit насыщенности со знаком это означает что:

- если значение, которое будет насыщаться, является меньше, чем -2^{n-1} , результат возвратился, -2^{n-1}
- если значение, которое будет насыщаться, больше, чем $2^{n-1}-1$, результат возвратился, $2^{n-1}-1$
- иначе, результат возвратился, совпадает со значением, которое будет насыщаться.

Для n -bit насыщенности без знака это означает что:

- если значение, которое будет насыщаться, является меньше чем 0, результат возвратился, 0
- если значение, которое будет насыщаться, больше, чем 2^{n-1} , результат возвратился, 2^{n-1}
- иначе, результат возвратился, совпадает со значением, которое будет насыщаться.

Если возвращенный результат отличается от значения, которое будет насыщаться, это вызывают насыщенностью. Если насыщенность происходит, системы команд флаг Q к 1 в APSR. Иначе, это оставляет флаг Q неизменным. Чтобы очистить флаг Q к 0, Вы должны использовать инструкцию MSR, видеть MSR на странице 174.

Чтобы считать состояние флага Q, используйте инструкцию MRS, посмотрите MRS на странице 173.

3.7.1 SSAT и USAT

Насыщают, и Без знака Насыщают к любой позиции двоичного разряда, с дополнительным сдвигом перед насыщением.

Syntax

op{cond} Rd, #n, Rm {, shift #s}

где:

- op' является одним из:

SSAT: Насыщает значение со знаком к диапазону со знаком. USAT: Насыщает значение со знаком к диапазону без знака.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'n' определяет позицию двоичного разряда, чтобы насыщать к: n колеблется от 1 до 32 для SSAT

n колеблется от 0 до 31 для USAT.

- 'Комната' - регистр, содержащий значение, чтобы насыщать.
- 'shift #s' является дополнительным сдвигом, применялся к Rm перед насыщением. Это должно быть одно из следующего:

ASR #s: где s находится в диапазоне 1 - 31 LSL #s: где s находится в диапазоне от 0 до 31.

Работа

Эти инструкции насыщают к значению n -bit без знака или со знаком.

Инструкция SSAT применяет указанный сдвиг, затем насыщает к диапазону со знаком $-2^{n-1} \leq x$

$\leq 2^{n-1}-1$.

Инструкция USAT применяет указанный сдвиг, затем насыщает к диапазону без знака $0 \leq x \leq 2^n-1$.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не влияют на флаги кода условия.

Если насыщенность происходит, эти инструкции устанавливают флаг Q в 1.

Примеры

SSAT R7, № 16, R7, LSL № 4; Логический сдвиг оставил значение в R7 4, тогда насыщайте его как 16-разрядное значение со знаком и запишите его обратно к R7

USATNE R0, № 7, R5; Условно насыщайте значение в R5 как ; 7 битовых значений без знака и запись это к R0.

0: SSAT16 и USAT16

Подписанный Насыщают, и Без знака Насыщают к любой позиции двоичного разряда для двух полуслов.

Syntax

op{cond} Rd, #n, Rm

где:

- op' является одним из:

SSAT16 Насыщает значение полуслова со знаком к диапазону со знаком. USAT16 Насыщает значение полуслова со знаком к диапазону без знака.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'n' определяет позицию двоичного разряда, чтобы насыщать к: n колеблется от 1 до 16 для SSAT. n колеблется от 0 до 15 для USAT.
- 'Rm' - регистр, содержащий значение, чтобы насыщать.

Работа

Инструкция SSAT16:

1. Насыщает два 16-разрядных значения полуслова со знаком регистра со значением, чтобы насыщать от выбранной позицией двоичного разряда в n.
2. Пишут результаты как два 16-разрядных полуслова со знаком к целевому регистру.

Инструкция USAT16:

1. Насыщает два 16-разрядных значения полуслова без знака регистра со значением, чтобы насыщать от выбранной позицией двоичного разряда в п.

2. Пишет результаты как два полуслова без знака в целевом регистре.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не влияют на флаги кода условия.

Если насыщенность происходит, эти инструкции устанавливают флаг Q в 1.

Примеры

SSAT16 R7, № 9, R2; Насыщает вершину и нижнюю часть R2 как 9-разрядные значения, пишет в соответствующее полуслово из R7

USAT16NE R0, № 13, R5; Условно насыщает вершину и нижнюю часть полуслова R5 как 13-разрядные значения, пишет в соответствующее полуслово R0.

3.7.3 QADD и QSUB

Насыщение Добавляет, и Насыщение Вычитают,

Syntax

op{cond} {Rd}, Rn, Rm

op{cond} {Rd}, Rn, Rm

где:

- op' является одним из:

QADD: 32-разрядное Насыщенное суммирование

QADD8: Насыщение четырех 8-разрядных целочисленных дополнений. QADD16: Насыщение двух 16-разрядных целочисленных дополнений. QSUB: Насыщение 32-разрядного вычитания.

QSUB8: Насыщение четырех 8-разрядных целочисленных вычитаний. QSUB16: Насыщение двух 16-разрядных целочисленных вычитаний.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'Rn, Rm' является регистрами, содержащими первые и вторые операнды.

Работа

Эти инструкции добавляют или вычитают два, четыре или восемь значений от первых и вторых операндов и затем пишут важное значение со знаком в целевом регистре.

QADD и инструкции QSUB применяются, указанные добавляют или вычитают, и затем насыщают результат к диапазону со знаком $-2^{n-1} \leq x \leq 2^{n-1}-1$, где x дан числом битов, примененных в

инструкция, 32, 16 или 8.

Если возвращенный результат отличается от значения, которое будет насыщаться, это вызывают насыщенностью. Если насыщенность происходит, QADD и инструкции QSUB устанавливают флаг APSR Q в 1. Иначе, флаг Q неизменен. 8-разрядный и 16-разрядный QADD и инструкции QSUB всегда оставляют флаг Q неизменным.

Чтобы очистить флаг Q к 0, Вы должны использовать инструкцию MSR, видеть MSR на странице 174. Чтобы считать состояние флага Q, используйте инструкцию MRS, посмотрите MRS на странице 173.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не влияют на флаги кода условия.

Если насыщенность происходит, эти инструкции устанавливают флаг Q в 1.

Примеры

QADD16 R7, R4, R2; Добавляют полуслова R4 с соответствующим полусловом
; R2, насыщает к 16 битам и пишет в соответствие
; полуслово R7

QADD8 R3, R1, R6; Добавляют байты R1 к соответствующим байтам R6, насыщает
; к 8 битам и записи к соответствующему байту R3

QSUB16 R4, R2, R3; Вычитает полуслова R3 от соответствующего полуслова
; из R2, насыщает к 16 битам, пишет в соответствие ; полуслово R4

QSUB8 R4, R2, R5; Вычитает байты R5 от соответствующего байта в R2; насыщает к 8 битам, пишет в соответствующий байт ofR4.

3.7.4 QASX и QSAX

Насыщение Добавляет и Вычитает с обменом, и Насыщение Вычитают и Добавляют обменом ,
Syntax

op{cond} {Rd}, Rm, Rn

где:

- op' является одним из:

QASX добавляют и вычитают обменом и насыщают. QSAX вычитают и добавляют с обменом и насыщают.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'Rn, Rm' является регистрами, содержащими первые и вторые операнды.

Работа

Инструкция QASX:

1. Добавляет главное полуслово исходного операнда с нижним полусловом второго операнда.
2. Вычитает главное полуслово второго операнда от нижней части первого операнда.
3. Насыщает результат вычитания и пишет 16-разрядное целое число со знаком в диапазоне $-2^{15} \leq x \leq 2^{15} - 1$, где x равняется 16 к нижнему полуслову целевого регистра.
4. Насыщает результаты суммы и пишет 16-разрядное целое число со знаком в диапазоне

5. $-215 \leq x \leq 215 - 1$, где x равняется 16 к главному полуслову целевого регистра.

Инструкция QSAX:

1. Вычитает нижнее полуслово второго операнда от вершины первого операнда.
2. Добавляет нижнее полуслово исходного операнда с главным полусловом второго операнда.
3. Насыщает результаты суммы и пишет 16-разрядное целое число со знаком в диапазоне
4. $-215 \leq x \leq 215 - 1$, где x равняется 16 к нижнему полуслову целевого регистра.
5. Насыщает результат вычитания и пишет 16-разрядное целое число со знаком в диапазоне $-215 \leq x \leq 215 - 1$, где x равняется 16 к главному полуслову целевого регистра.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не влияют на флаги кода условия.

Примеры

QASX R7, R4, R2; Добавляет главное полуслово R4, чтобы оснастить полуслово R2, насыщает к 16 битам, пишет в главное полуслово R7. Вычитает вершину R2 от нижнего полуслова R4, насыщает к 16 битам и пишет в нижнее полуслово из R7.

QSAX R0, R3, R5; Вычитает нижнее полуслово R5 от главного полуслова R3, насыщает к 16 битам, пишет в главное полуслово R0. Добавляет нижнее полуслово R3 к главному полуслову R5, насыщает к 16 битам, пишет в нижнее полуслово R0.

3.7.5 QDADD и QDSUB

Двойное насыщение с суммированием и вычитанием

Syntax

op{cond} {Rd}, Rm, Rn

где:

- op' является одним из: QDADD, насыщающие дважды и, добавляют. QDSUB, насыщающие дважды и, вычитают.
- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'Rn, Rm' является регистрами, содержащими первые и вторые операнды.

Работа

Инструкция QDADD:

1. Удваивает второе значение операнда.
2. Добавляет результат удвоения до влажного значения со знаком в первом операнде.
3. Пишет результат в целевой регистр.

Инструкция QDSUB:

1. Удваивает второе значение операнда.
2. Вычитает удвоенное значение из важного значения со знаком в первом операнде.
3. Пишет результат в целевой регистр.

И удвоение и целое число со знаком располагаются –231 флаг в APSR. дополнению или вычитанию насыщали их результаты к 32-битам $\leq x \leq 231 - 1$. Если насыщенность происходит в любой работе, она устанавливает Q

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Если насыщенность происходит, эти инструкции устанавливают флаг Q в 1.

Примеры

QDADD R7, R4, R2; Удваивает и насыщает R4 к 32 битам, добавляет R2, ; насыщает к 32 битам, пишет в R7 QDSUB R0, R3, R5; Вычитает R3, удвоенный и влажный до 32 битов ; от R5, насыщает к 32 битам, пишет в R0.

3.7.6 UQASX и UQSAX

Двоеное насыщение с суммированием и вычитанием включающее обмен

Syntax

op{cond} {Rd}, Rm, Rn

где:

- op' является одним из:

UQASX добавляют и вычитают с обменом и насыщают. UQSAX вычитают и добавляют с обменом и насыщают.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'Rn,Rm' является регистрами, содержащими первые и вторые операнды.

Работа

Инструкция UQASX:

1. Добавляет нижнее полуслово исходного операнда с главным полусловом второго операнда.
2. Вычитает нижнее полуслово второго операнда от вершины первого операнда.
3. Насыщает результаты суммы и пишет 16-разрядное целое без знака в диапазоне
4. $0 \leq x \leq 216 - 1$, где x равняется 16 к главному полуслову целевого регистра.
5. Насыщает результат вычитания и пишет 16-разрядное целое без знака в диапазоне $0 \leq x \leq 216 - 1$, где x равняется 16 к нижнему полуслову целевого регистра.

Инструкция UQSAX:

1. Вычитает нижнее полуслово второго операнда от вершины первого операнда.
2. Добавляет нижнее полуслово первого операнда с главным полусловом второго операнда.
3. Насыщает результат вычитания и пишет 16-разрядное целое без знака в диапазоне $0 \leq x \leq 2^{16} - 1$, где x равняется 16 к главному полуслову целевого регистра.
4. Насыщает результаты дополнения и пишет 16-разрядное целое без знака в диапазоне $0 \leq x \leq 2^{16} - 1$, где x равняется 16 к нижнему полуслову целевого регистра.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не влияют на флаги кода условия.

Примеры

UQASX R7, R4, R2; Добавляет главное полуслово R4 с нижним полусловом R2, насыщает к 16 битам, пишет в главное полуслово R7. Вычитает главное полуслово R2 от нижнего полуслова R4, насыщает к 16 битам, пишет в нижнее полуслово R7.

UQSAX R0, R3, R5; Вычитает нижнее полуслово R5 от главного полуслова; R3, насыщает к 16 битам, пишет в главное полуслово R0. Добавляет нижнее полуслово R4 к главному полуслову R5, насыщает к 16 битам, пишет в нижнее полуслово R0.

3.7.7 UQADD и UQSUB

Насыщение добавляет, и насыщение вычитают без знака.

Syntax

op{cond} {Rd}, Rn, Rm

op{cond} {Rd}, Rn, Rm

где:

- op' является одним из:

UQADD8, Насыщающий четыре 8-разрядных целочисленных дополнения без знака. UQADD16, Насыщающий два 16-разрядных целочисленных дополнения без знака. UDSUB8, Насыщающий четыре 8-разрядных целочисленных вычитания без знака. UQSUB16, Насыщающий два 16-разрядных целочисленных вычитания без знака.

- 'cond' - дополнительный код условия (см. Условное выполнение на странице 64),
- 'Rd.' - целевой регистр.
- 'Rn, rm' является регистрами, содержащими первые и вторые операнды.

Работа

Эти инструкции добавляют или вычитают два или четыре значения, и затем пишет значение без знака в целевом регистре.

Инструкция UQADD16:

1. Добавляют соответствующие полуслова вершины и нижней части первых и вторых операндов.
2. Насыщает результат дополнений для каждого полуслова в целевом регистре к диапазону без знака $0 \leq x \leq 2^{16}-1$, где x равняется 16.

Инструкция UQADD8:

1. Добавляет каждый соответствующий байт первых и вторых операндов.
2. Насыщает результат дополнения для каждого байта в целевом регистре к диапазону без знака $0 \leq x \leq 2^8-1$, где x равняется 8.

Инструкция UQSUB16:

0: Вычитает оба полуслова второго операнда от соответствующих полуслов первого операнда.

- 1: Насыщает результат различий в целевом регистре к диапазону без знака $0 \leq x \leq 2^{16}-1$, где x равняется 16.

Инструкции UQSUB8:

1. Вычитает соответствующие байты второго операнда от соответствующих байтов первого операнда.
2. Насыщает результаты различий для каждого байта в целевом регистре к диапазону без знака $0 \leq x \leq 2^8-1$, где x равняется 8.

Ограничения

Не используйте SP и не используйте PC.

Флаги условия

Эти инструкции не влияют на флаги кода условия.

Примеры

UQADD16 R7, R4, R2; Добавляют полуслова в R4 к соответствующему полуслову в R2,

; насыщает к 16 битам, пишет в соответствующее полуслово ; из R7

UQADD8 R4, R2, R5; Добавляют байты R2 к соответствующему байту R5, насыщает

; к 8 битам, пишет в соответствующие байты R4

UQSUB16 R6, R3, R0; Вычитает полуслова в R0 от соответствующего полуслова

; в R3, насыщает к 16 битам, пишет в соответствие

; полуслово в R6

UQSUB8 R1, R5, R6; Вычитает байты в R6 от соответствующего байта R5,

; насыщает к 8 битам, пишет в соответствующий байт

R1.

3.8

PKH	Сокращение	PKHBT and PKHTB
SXTAB	Расширяют 8-битов на 32 и суммируют	SXTA и UXTA стр 135
SXTAB16	Расширяют на 8 битов и на 16 битов и суммируют	SXTA и UXTA стр 135
SXTAH	Расширяют на 16 и 32 битов и суммируют	SXTA и UXTA стр 135
SXTB	Расширяет байт	SXT и UXT стр 134
SXTB16	Расширяют на 8 битов и на 16 и суммируют	SXT и UXT стр 134
SXTH	Расширяет сокращение	SXT и UXT стр 134
UXTAB	Расширяют 8-битов на 32 и суммируют	SXTA и UXTA стр 135

UXTAB16	Расширяют на 8 битов и на 16 битов и суммируют	SXTA и UXTA стр 135
UXTAN	Расширяют на 16 и 32 битов и суммируют	SXTA и UXTA стр 135
UXTB	Расширяет байту	SXT и UXT стр 134
UXTB16	Расширяют на 8 битов и на 16 и суммируют	SXT и UXT стр 134
UXTN	Расширяет сокращение	SXT и UXT стр 134