

Содержание главы 10

<u>10</u>	<u>Арифметические функции</u>	4
<u>10.1</u>	<u>Обработка арифметической функции</u>	4
<u>10.2</u>	<u>Вычисления с типом данных INT</u>	8
<u>10.3</u>	<u>Вычисления с типом данных DINT</u>	9
<u>10.4</u>	<u>Вычисления с типом данных REAL</u>	11

10 Арифметические функции

Арифметические функции комбинируют два значения в соответствии с основными арифметическими операциями сложения, вычитания, умножения и деления. Вы можете применять арифметические функции к переменным типов INT, DINT и REAL (таблица 10.1).

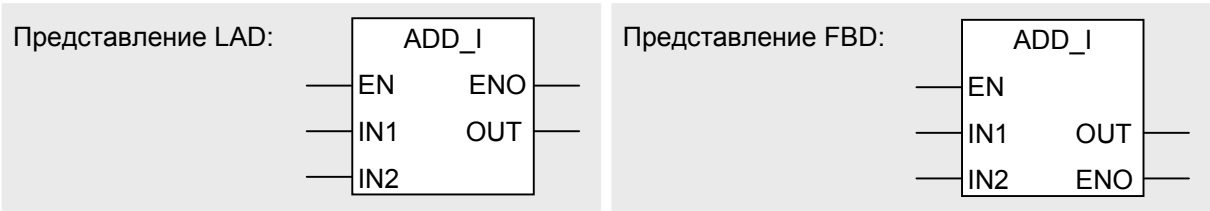
Таблица 10.1 Обзор арифметических функций

Арифметическая функция	С типом данных		
	INT	DINT	REAL
Сложение	ADD_I	ADD_DI	ADD_R
Вычитание	SUB_I	SUB_DI	SUB_R
Умножение	MUL_I	MUL_DI	MUL_R
Деление с частным в качестве результата	DIV_I	DIV_DI	DIV_R
Деление с остатком в качестве результата	-	MOD_DI	-

10.1 Обработка арифметической функции

Представление

Арифметический блочный элемент
(в примере: сложение в соответствии с INT)



Кроме разрешающего входа (enable input) EN и разрешающего выхода (enable output) ENO блочный элемент арифметической функции имеет два входа IN1 и IN2 и выход OUT. «Заголовок» в блочном элементе идентифицирует исполняемое арифметическое действие (ADD_I, например, означает сложение чисел типа INT).

Комбинируемые значения подаются на входы IN1 и IN2, результат вычисления находится на выходе OUT. Входы и выход могут иметь различные типы данных в зависимости от арифметической функции. К примеру, в случае арифметической функции ADD_R (сложение чисел типа REAL) входы и выход отнесены к типу REAL. Используемые переменные должны быть того же типа данных, что и входы или выход. Если вы используете для операндов абсолютные адреса, то размеры операндов

должны соответствовать типам данных. Так, вы можете применить операнд размером в слово для типа данных INT.

Описание отдельных битов в каждом формате данных вы можете найти параграфе 3.5.4 «Простые типы данных».

Функция

Арифметическая функция выполняется, если на разрешающем входе присутствует «1» (во входе EN течет ток). Если во время вычисления возникает ошибка, то разрешающий выход устанавливается в «0»; в противном случае он устанавливается в «1». Если выполнение функции не разрешено (EN = «0»), то вычисление не производится, и ENO также обнуляется.

Если главное реле управления (MCR) активировано, то выход OUT устанавливается в нуль, когда арифметическая функция обрабатывается (EN = «1»). MCR не влияет на выход ENO.

ЕСЛИ EN == “1” или не используется		
ТО		ИНАЧЕ
OUT := IN1 Cfct IN2		
ЕСЛИ возникла ошибка		
ТО	ИНАЧЕ	
ENO := “0”	ENO := “1”	ENO := “0”

Cфct – арифметическая функция

Во время выполнения арифметической функции могут возникнуть следующие ошибки:

- Выход за пределы диапазона (переполнение) в вычислениях с типами INT и DINT;
- Исчезновение значащих разрядов и переполнение в вычислениях с типом REAL;
- Недействительное (недопустимое) число REAL в вычислениях с типом REAL.

За информацией о том, как арифметические функции устанавливают различные биты состояния, обратитесь к главе 15 «Биты состояния».

Примеры

На рисунке 10.1 приведены примеры для каждого типа данных. Арифметическая функция выполняет вычисление в соответствии с определенными параметрами, даже если не были объявлены типы данных при использовании операндов с абсолютными адресами.

В случае пошагового программирования арифметические функции вы можете найти в каталоге программных элементов (Program Element Catalog), вызываемого командой меню View → Catalog (Вид → Каталог) [Ctrl + K] или Insert → Program Elements (Вставка → Программные элементы), в разделе «Integer Math» («Вычисления с целыми числами», вычисления с типами данных INT и DINT) и в разделе «Floating-Point Math» («Вычисления с числами с плавающей запятой», вычисления с типом данных REAL).

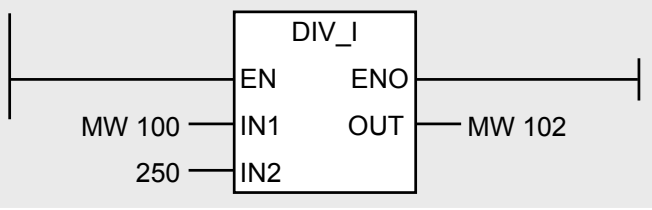
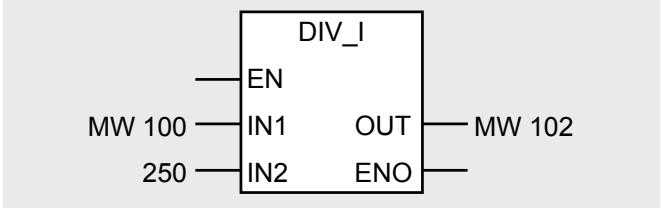
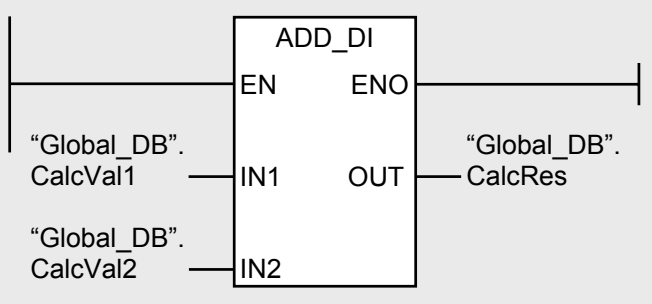
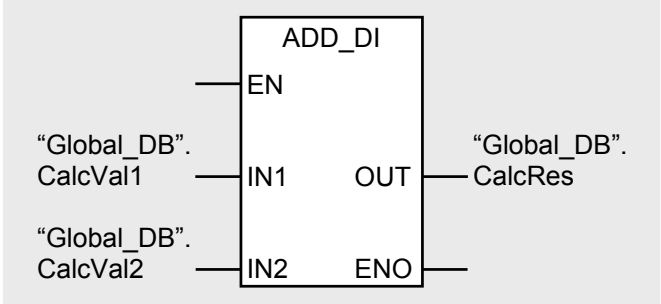
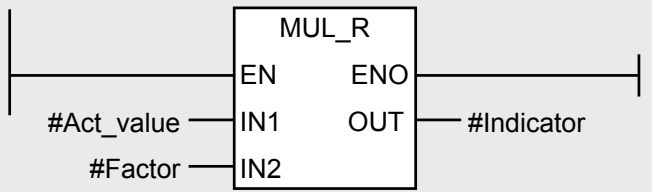
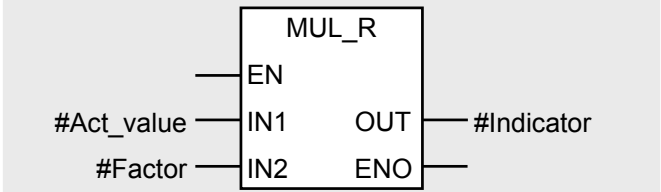
Вычисление в соответствии с INT	Значение слова памяти (меркеров) MW 100 делится на 250; целочисленный результат сохраняется в слове памяти (меркеров) MW 102.
	
Вычисление в соответствии с DINT	Значения переменных «CalcVal1» и «CalcVal2» складываются, результат сохраняется в переменной «CalcRes». Все переменные находятся в блоке данных «Global_DB».
	
Вычисление в соответствии с REAL	Переменная #Act_value умножается на переменную #Factor, результат пересылается в переменную #Indicator.
	

Рисунок 10.1 примеры арифметических функций

Арифметические функции в цепи (LAD)

Вы можете поставить контакты, соединив их последовательно или параллельно, перед входом EN и после выхода ENO.

Сам арифметический блочный элемент может быть помещен после Т-ветви и в ветви, ведущей непосредственно к левой питающей направляющей (шине питания). Эта ветвь также может иметь контакты перед входом EN и не может быть самой верхней ветвью.

Прямое соединение с левой питающей направляющей означает, что вы можете подключить арифметические блочные элементы параллельно. Когда блочные элементы подключаются параллельно, требуется катушка для завершения цепи. Если вы не производите проверку на наличие ошибки, назначьте катушке «пустой» операнд, к примеру, бит временных локальных данных.

Можно соединить арифметические блочные элементы последовательно. Если выход ENO предыдущего блочного элемента соединен с входом EN следующего, то последний срабатывает только, если предыдущий элемент был обработан без ошибок. Если вы хотите использовать результат из предыдущего блочного элемента в качестве входного значения для следующего блока, то переменные из области временных локальных данных предоставляют удобные промежуточные буферы.

Если вы komponуете несколько арифметических блочных элементов в одной цепи (параллельно левой питающей направляющей и в дальнейшем последовательно), блочные элементы самой верхней ветви обрабатываются слева направо, затем элементы второй ветви слева направо и так далее.

Библиотека «LAD_Book» на дискете, прилагаемой к книге, содержит примеры представления и группировки арифметических функций (FB 110 в программе «Digital Functions», «Числовые функции»).

Арифметические функции в логической цепи (FBD)

EN и ENO могут быть не назначены. Если требуется обработать арифметический блочный элемент согласно определенным условиям, то можно скомпоновать логические операции перед входом EN. Можно соединить выход ENO с бинарными входами других функций. Например, вы можете выстроить арифметические блочные элементы последовательно, таким образом, выход ENO предыдущего блочного элемента будет подключен к входу EN следующего. Если вы хотите использовать результат вычислений из предыдущего блочного элемента в качестве входного значения для последующего элемента, то переменные из области временных локальных данных предоставляют удобные промежуточные буферы.

Библиотека «FBD_Book» на дискете, прилагаемой к книге, содержит примеры представления и группировки арифметических функций (FB 110 в программе «Digital Functions», «Числовые функции»).

10.2 Вычисления с типом данных INT

Сложение с типом INT

Функция ADD_I интерпретирует значения на входах IN1 и IN2 как числа типа данных INT (целочисленный тип). Она складывает два числа и сохраняет сумму на выходе OUT.

После выполнения вычисления биты состояния CC0 и CC1 указывают на отрицательную, нулевую или положительную сумму. Биты состояния OV и OS индицируют любой выход за границы диапазона.

Вычитание с типом INT

Функция SUB_I интерпретирует значения на входах IN1 и IN2 как числа типа данных INT. Она вычитает значение на IN2 из значения на IN1 и сохраняет разность на выходе OUT.

После вычисления биты состояния CC0 и CC1 показывают, какова разность - отрицательная, нулевая или положительная. Биты состояния OV и OS сигнализируют о любом нарушении границ диапазона.

Умножение с типом INT

Функция MUL_I интерпретирует значения на входах IN1 и IN2 как числа типа данных INT. Она умножает два значения и сохраняет произведение на выходе OUT.

После выполнения вычисления биты состояния CC0 и CC1 указывают, каково произведение - отрицательное, равно нулю или положительное. Биты состояния OV и OS сигнализируют о любом нарушении границ диапазона типа INT.

Деление с типом INT

Функция DIV_I интерпретирует значения на входах IN1 и IN2 как числа типа данных INT. Она делит значение на входе IN1 (делимое) на значение на входе IN2 (делитель) и передает частное на выход OUT. Этот результат деления является целым. Частное равно нулю, если делимое равно нулю, а делитель не равен нулю, или если абсолютное значение делимого меньше абсолютного значения делителя. Частное отрицательное, если делитель отрицательный.

После вычисления биты состояния CC0 и CC1 показывают, частное отрицательное, равно нулю или положительное. Биты состояния OV и OS индицируют любой выход за границы диапазона. Деление на ноль выдает нулевое частное и устанавливает биты состояния CC0, CC1, OV и OS в «1».

10.3 Вычисления с типом данных DINT

Сложение с типом DINT

Функция ADD_DI интерпретирует значения на входах IN1 и IN2 как числа типа данных DINT (двойное целое). Она складывает два числа и сохраняет сумму на выходе OUT.

После выполнения вычисления биты состояния CC0 и CC1 указывают на отрицательную, нулевую или положительную сумму. Биты состояния OV и OS индицируют любой выход за границы диапазона.

Вычитание с типом DINT

Функция SUB_DI интерпретирует значения на входах IN1 и IN2 как числа типа данных DINT. Она вычитает значение на IN2 из значения на IN1 и сохраняет разность на выходе OUT.

После вычисления биты состояния CC0 и CC1 показывают, какова разность - отрицательная, нулевая или положительная. Биты состояния OV и OS сигнализируют о любом нарушении границ диапазона.

Умножение с типом DINT

Функция MUL_DI интерпретирует значения на входах IN1 и IN2 как числа типа данных DINT. Она умножает два значения и сохраняет произведение на выходе OUT.

После выполнения вычисления биты состояния CC0 и CC1 указывают, каково произведение - отрицательное, равно нулю или положительное. Биты состояния OV и OS сигнализируют о любом нарушении границ диапазона типа INT.

Деление с типом DINT, в качестве результата – частное

Функция DIV_DI интерпретирует значения на входах IN1 и IN2 как числа типа данных DINT. Она делит значение на входе IN1 (делимое) на значение на входе IN2 (делитель) и сохраняет частное на выходе OUT. Этот результат деления является целым. Частное равно нулю, если делимое равно нулю, а делитель не равен нулю, или если абсолютное значение делимого меньше абсолютного значения делителя. Частное отрицательное, если делитель меньше нуля.

После вычисления биты состояния CC0 и CC1 показывают, частное отрицательное, равно нулю или положительное. Биты состояния OV и OS индицируют любой выход за границы диапазона. Деление на ноль выдает нулевое частное и устанавливает биты состояния CC0, CC1, OV и OS в «1».

Деление с типом DINT, в качестве результата – остаток

Функция MOD_DI интерпретирует значения на входах IN1 и IN2 как числа типа данных DINT. Она делит значение на входе IN1 (делимое) на значение на входе IN2 (делитель) и сохраняет остаток на выходе OUT. Остаток – это то, что осталось от деления; он не соответствует десятичным разрядам. Если делимое отрицательное, то остаток также меньше нуля.

После вычисления биты состояния CC0 и CC1 показывают, каков остаток - отрицательный, равен нулю или положительный. Биты состояния OV и OS индицируют любой выход за границы диапазона. Деление на ноль возвращает нулевой остаток и устанавливает биты состояния CC0, CC1, OV и OS в «1».

10.4 Вычисления с типом данных REAL

Числа типа REAL (вещественные) представляются внутренне как числа с плавающей десятичной точкой (запятой) с двумя диапазонами чисел. Один диапазон с полной точностью («нормализованные» числа с плавающей точкой) и один диапазон с ограниченной точностью («ненормализованные» числа с плавающей точкой; обратитесь также к параграфу 3.5.4 «Простые типы данных»). CPU S7-400 производят вычисления в обоих диапазонах, CPU S7-300 – только в диапазоне с полной точностью. Если CPU S7-300 выполняют вычисление, результат которого находится в диапазоне с ограниченной точностью, то в качестве результата возвращается нуль, и сообщается о выходе из диапазона.

Сложение с типом REAL

Функция ADD_R интерпретирует значения на входах IN1 и IN2 как числа типа данных REAL. Она складывает два числа и сохраняет сумму на выходе OUT.

После выполнения вычисления биты состояния CC0 и CC1 указывают на отрицательную, нулевую или положительную сумму. Биты состояния OV и OS индицируют любой выход за границы диапазона.

В случае недопустимого вычисления (одно из входных значений является действительным числом типа REAL, или вы пытаетесь сложить $+\infty$ и $-\infty$) ADD_R возвращает действительное значение на выход OUT и устанавливает биты состояния CC0, CC1, OV и OS в «1».

Вычитание с типом REAL

Функция SUB_R интерпретирует значения на входах IN1 и IN2 как числа типа данных REAL. Она вычитает значение на входе IN2 из значения на входе IN1 и сохраняет разность на выходе OUT.

После вычисления биты состояния CC0 и CC1 показывают, какова разность - отрицательная, нулевая или положительная. Биты состояния OV и OS сигнализируют о любом нарушении границ диапазона.

В случае недопустимого вычисления (одно из входных значений является действительным числом типа REAL, или вы пытаетесь вычесть $+\infty$ из $-\infty$) SUB_R возвращает действительное значение на выход OUT и устанавливает биты состояния CC0, CC1, OV и OS в «1».

Умножение с типом REAL

Функция MUL_R интерпретирует значения на входах IN1 и IN2 как числа типа данных REAL. Она умножает два значения и сохраняет произведение на выходе OUT.

После выполнения вычисления биты состояния CC0 и CC1 указывают, каково произведение - отрицательное, равно нулю или положительное. Биты состояния OV и OS сигнализируют о любом нарушении границ диапазона.

В случае недопустимого вычисления (одно из входных значений является недействительным числом типа REAL, или вы пытаетесь умножить ∞ на 0) MUL_R возвращает недействительное значение на выход OUT и устанавливает биты состояния CC0, CC1, OV и OS в «1».

Деление с типом REAL

Функция DIV_R интерпретирует значения на входах IN1 и IN2 как числа типа данных REAL. Она делит значение на входе IN1 (делимое) на значение на входе IN2 (делитель) и передает частное на выход OUT.

После вычисления биты состояния CC0 и CC1 показывают, частное отрицательное, равно нулю или положительное. Биты состояния OV и OS индицируют любой выход за границы диапазона. Деление на ноль выдает нулевое частное и устанавливает биты состояния CC0, CC1, OV и OS в «1».

В случае недопустимого вычисления (одно из входных значений является недействительным числом типа REAL, или вы пытаетесь поделить ∞ на ∞ или 0 на 0) DIV_R возвращает недействительное значение на выход OUT и устанавливает биты состояния CC0, CC1, OV и OS в «1».