

Алфавитный список команд

A

A.1	Список команд с международными именами	A-2
A.2	Список команд с международными (английскими) именами и их русские эквиваленты	A-5
A.3	Список команд на русском языке с краткими обозначениями SIMATIC	A-9
A.4	Список команд на русском языке и международные (английские) эквиваленты	A-12

A.1 Список команд с международными именами

Таблица A–1 содержит алфавитный список команд FUP упорядоченный по их международным (английским) полным именам, с соответствующим сокращенным именем или мнемоникой и ссылкой на страницу, на которой эта команда описана.

Таблица A–1. Команды FUP, упорядоченные по алфавиту в соответствии с международными (английскими) полными именами с указанием сокращенных наименований

Полное имя	Краткое имя	страница
Add Double Integer	ADD_DI	11–3
Add Integer	ADD_I	11–2
Add Real	ADD_R	12–3
Address Negative Edge Detection	NEG	8–30
Address Positive Edge Detection	POS	8–29
AND	&	8–3
Assign	=	8–9
Assign a Value	MOVE	14–2
BCD to Double Integer	BCD_DI	14–6
BCD to Integer	BCD_I	14–3
Call FB from Box	CALL_FB	20–4
Call FC from Box	CALL_FC	20–4
Call FC SFC (без параметров)	CALL	20–2
Call System FB from Box	CALL_SFB	20–4
Call System FC from Box	CALL_SFC	20–4
Ceiling	CEIL	14–16
Compare Double Integer (>, <, ==, <=, >=)	CMP >=D	13–4
Compare Integer (>, <, ==, <=, >=)	CMP >=I	13–2
Compare Real (>, <, ==, <=, >=)	CMP >=R	13–6
Divide Double Integer	DIV_DI	11–9
Divide Integer	DIV_I	11–8
Divide Real	DIV_R	12–6
Double Integer to BCD	DI_BCD	14–7
Double Integer to Real	DI_R	14–8
Down Counter (операция со счетчиком)	S_CD	10–7
Down Counter (битовая логическая операция)	CD	8–16
Exception Bit Binary Result	BR	19–3
Exception Bit Overflow	OV	19–7
Exception Bit Overflow Stored	OS	19–8
Exception Bit Unordered	UO	19–6
Exclusive OR	XOR	8–6
Extended Pulse S5 Timer (таймерная команда)	S_PEXT	9–7
Extended Pulse Timer (битовая логическая операция)	SE	8–19
Floor	FLOOR	14–17
Form the Absolute Value of a Floating-Point Number	ABS	12–8

Form the Arc Cosine of a Floating-Point Number	ACOS	12-13
Form the Arc Sine of a Floating-Point Number	ASIN	12-13
Form the Arc Tangent of a Floating-Point Number	ATAN	12-13
Form the Cosine of a Floating-Point Number	COS	12-13
Form the Exponential Value of a Floating-Point Number	EXP	12-12
Form the Natural Logarithm of a Floating-Point Number	LN	12-11
Form the Sine of a Floating-Point Number	SIN	12-13
Form the Square of a Floating-Point Number	SQR	12-9
Form the Square Root of a Floating-Point Number	SQRT	12-10
Form the Tangent of a Floating-Point Number	TAN	12-13
Insert Binary Input	---	8-7
Integer to BCD	I_BCD	14-4
Integer to Double Integer	I_DI	14-5
Jump	JMP	18-3
Jump-If-Not	JMPN	18-5
Master Control Relay Activate	MCRA	20-10
Master Control Relay Deactivate	MCRD	20-10
Master Control Relay Off	MCR>	20-13
Master Control Relay On	MCR<	20-13
Midline Output	#	8-10
Multiply Double Integer	MUL_DI	11-7
Multiply Integer	MUL_I	11-6
Multiply Real	MUL_R	12-5
Negate Binary Input	---o	8-8
Negate Real Number	NEG_R	14-13
Negative RLO Edge Detection	N	8-28
Off-Delay S5 Timer (таймерная команда)	S_OFFDT	9-13
Off-Delay Timer (битовая логическая операция)	SF	8-25
On-Delay S5 Timer (таймерная команда)	S_ODT	9-9
On-Delay Timer (битовая логическая операция)	SD	8-21
Ones Complement Double Integer	INV_DI	14-10
Ones Complement Integer	INV_I	14-9
Open Data Block: DB или DI	OPN	17-2
OR	>=1	8-4
Positive RLO Edge Detection	P	8-27
Pulse S5 Timer (таймерная команда)	S_PULSE	9-5
Pulse Timer (битовая логическая операция)	SP	8-17
Reset Output	R	8-13
Reset_Set Flip Flop	RS	8-32
Result Bit Equal 0	==0	19-6
Result Bit Greater Equal 0	>=0	19-4
Result Bit Greater Than 0	>0	19-4
Result Bit Less Equal 0	<=0	19-4

Result Bit Less Than 0	<0	19–4
Result Bit Not Equal 0	<>0	19–4
Retentive On–Delay S5 Timer (таймерная команда)	S_ODTS	9–11
Retentive On–Delay Timer (битовая логическая операция)	SS	8–23
Return	RET	20–8
Return Fraction Double Integer	MOD	11–10
Rotate Left Double Word	ROL_DW	16–10
Rotate Right Double Word	ROR_DW	16–10
Round to Double Integer	ROUND	14–14
Save RLO to BR Memory	SAVE	8–11
Set Output	S	8–12
Set Counter Value	SC	8–14
Set_Reset Flip Flop	SR	8–31
Shift Left Double Word	SHL_DW	16–2
Shift Left Word	SHL_W	16–2
Shift Right Double Integer	SHR_DI	16–2
Shift Right Double Word	SHR_DW	16–2
Shift Right Integer	SHR_I	16–2
Shift Right Word	SHR_W	16–2
Subtract Double Integer	SUB_DI	11–5
Subtract Integer	SUB_I	11–4
Subtract Real	SUB_R	12–4
Truncate Double Integer Part	TRUNC	14–15
Twos Complement Double Integer	NEG_DI	14–12
Twos Complement Integer	NEG_I	14–11
Up Counter (операция со счетчиком)	S_CU	10–5
Up Counter (битовая логическая операция)	CU	8–15
Up–Down Counter	S_CUD	10–3
(Word) AND Double Word	WAND_DW	15–4
(Word) AND Word	WAND_W	15–3
(Word) Exclusive OR Double Word	WXOR_DW	15–8
(Word) Exclusive OR Word	WXOR_W	15–7
(Word) OR Double Word	WOR_DW	15–6
(Word) OR Word	WOR_W	15–5

А.2 Список команд с международными (английскими) именами и их русские эквиваленты

Таблица А–2 содержит алфавитный список команд FUP, упорядоченный в соответствии с международными (английскими) полными именами, их русские эквиваленты и ссылки на страницы, на которых эти команд описаны.

Таблица А–2. Команды FUP, упорядоченные по их международным (английским) полным именам, с их русскими эквивалентами

Полное имя	Русский эквивалент	страница
Add Double Integer	Сложить двойные целые числа (32 бита)	11–3
Add Integer	Сложить целые числа (16 бит)	11–2
Add Real	Сложить вещественные числа	12–3
Address Negative Edge Detection	Обнаружить отрицательный фронт по адресу	8–30
Address Positive Edge Detection	Обнаружить положительный фронт по адресу	8–29
AND	Логическая операция И	8–3
Assign	Присваивание	8–9
Assign a Value	Передача значения	14–2
BCD to Double Integer	Преобразование числа в формате BCD в двойное целое число (32 бита)	14–6
BCD to Integer	Преобразование числа в формате BCD в двойное число (16 бит)	14–3
Call FB from Box	Вызов FB	20–4
Call FC from Box	Вызов FC	20–4
Call FC SFC (без параметров)	Вызов FC/SFC без параметров	20–2
Call System FB from Box	Вызов системного FB	20–4
Call System FC from Box	Вызов системной FC	20–4
Ceiling	Округление до ближайшего большего целого числа	14–16
Compare Double Integer (>, <, ==, <=, >=)	Сравнение двойных целых чисел (32 бита)	13–4
Compare Integer (>, <, ==, <=, >=)	Сравнение целых чисел (16 бит)	13–2
Compare Real (>, <, ==, <=, >=)	Сравнение вещественных чисел	13–6
Divide Double Integer	Деление двойных целых чисел (32 бита)	11–9
Divide Integer	Деление целых чисел (16 бит)	11–8
Divide Real	Деление вещественных чисел	12–6
Double Integer to BCD	Преобразование двойного целого числа в число в формате BCD	14–7
Double Integer to Real	Преобразование двойного целого числа в число с плавающей точкой	14–8
Down Counter (операция со счетчиком)	Обратный счет	10–7
Down Counter (битовая логическая операция)	Обратный счет	8–16
Exception Bit Binary Result	Бит ошибки “Двоичный результат”	19–3
Exception Bit Overflow	Бит ошибки “Переполнение”	19–7
Exception Bit Overflow Stored	Бит ошибки “Переполнение с запоминанием”	19–8

Exception Bit Unordered	Бит ошибки “Неупорядочено”	19–6
Exclusive OR	Логическая операция Иключающее ИЛИ	8–6
Extended Pulse S5 Timer (таймерная операция)	Таймер с продленным импульсом (SV)	9–7
Extended Pulse Timer (битовая логическая операция)	Таймер с продленным импульсом SV	8–19
Floor	Округление до ближайшего меньшего целого числа	14–17
Form the Absolute Value of a Floating–Point Number	Образование абсолютного значения числа с плавающей точкой	12–8
Form the Arc Cosine of a Floating–Point Number	Образование арккосинуса числа с плавающей точкой	12–13
Form the Arc Sine of a Floating–Point Number	Образование арксинуса числа с плавающей точкой	12–13
Form the Arc Tangent of a Floating–Point Number	Образование арктангенса числа с плавающей точкой	12–13
Form the Cosine of a Floating–Point Number	Образование косинуса числа с плавающей точкой	12–13
Form the Exponential Value of a Floating–Point Number	Образование экспоненциального значения числа с плавающей точкой	12–12
Form the Natural Algorithm of a Floating–Point Number	Образование натурального логарифма числа с плавающей точкой	12–11
Form the Sine of a Floating–Point Number	Образование синуса числа с плавающей точкой	12–13
Form the Square of a Floating–Point Number	Образование квадрата числа с плавающей точкой	12–9
Form the Square Root of a Floating–Point Number	Образование квадратного корня числа с плавающей точкой	12–10
Form the Tangent of a Floating–Point Number	Образование тангенса числа с плавающей точкой	12–13
Insert Binary Input	Добавление двоичного входа	8–7
Integer to BCD	Преобразование целого числа (16 бит) в число в формате BCD	14–4
Integer to Double Integer	Преобразование целого числа (16 бит) в двойное целое число (32 бита)	14–5
Jump	Переход, если 1	18–3
Jump–If–Not	Переход, если 0	18–5
Master Control Relay Activate	Активизация Master Control Relay	20–10
Master Control Relay Deactivate	Деактивизация Master Control Relay	20–10
Master Control Relay Off	Выключение Master Control Relay	20–13
Master Control Relay On	Включение Master Control Relay	20–13
Midline Output	Коннектор	8–10
Multiply Double Integer	Умножение двойных целых чисел (32 бита)	11–7
Multiply Integer	Умножение целых чисел (16 бит)	11–6
Multiply Real	Умножение вещественных чисел	12–5
Negate Binary Input	Отрицание двоичного входа	8–8
Negate Real Number	Изменение знака вещественного числа	14–13
Negative RLO Edge Detection	Обнаружение отрицательного фронта RLO	8–28

OR	Логическая операция ИЛИ	8–4
Off-Delay S5 Timer (таймерная команда)	Таймер с задержкой выключения (SA)	9–13
Off-Delay Timer (битовая логическая операция)	Таймер с задержкой выключения (SA)	8–25
On-Delay S5 Timer (таймерная команда)	Таймер с задержкой включения (SE)	9–9
On-Delay Timer (битовая логическая операция)	Таймер с задержкой включения (SE)	8–21
Ones Complement Double Integer	Дополнение двойного целого числа (32 бита) до единицы	14–10
Ones Complement Integer	Дополнение целого числа (16 бит) до единицы	14–9
Open Data Block: DB или DI	Открытие блока данных	17–2
Positive RLO Edge Detection	Обнаружение положительного фронта RLO	8–27
Pulse S5 Timer (таймерная команда)	Импульсный таймер (SI)	9–5
Pulse Timer (битовая логическая операция)	Импульсный таймер (SI)	8–17
Reset Output	Сброс выхода	8–13
Reset_Set Flip Flop	Триггер RS	8–32
Result Bit Equal 0	Бит результата в случае равенства 0	19–4
Result Bit Greater Equal 0	Бит результата в случае ≥ 0	19–4
Result Bit Greater Than 0	Бит результата в случае > 0	19–4
Result Bit Less Equal 0	Бит результата в случае ≤ 0	19–4
Result Bit Less Than 0	Бит результата в случае < 0	19–4
Result Bit Not Equal 0	Бит результата в случае неравенства 0	19–4
Retentive On-Delay S5 Timer (таймерная команда)	Таймер с задержкой включения с запоминанием (SS)	9–11
Retentive On-Delay Timer (битовая логическая операция)	Таймер с задержкой включения с запоминанием (SS)	8–23
Return	Возврат	20–8
Return Fraction Double Integer	Получение остатка от деления двойных целых чисел	11–10
Rotate Left Double Word	Циклический сдвиг влево двойного слова	16–10
Rotate Right Double Word	Циклический сдвиг вправо двойного слова	16–10
Round to Double Integer	Округление до двойного целого	14–14
Save RLO to BR Memory	Сохранение RLO в бите BR	8–11
Set Output	Установка выхода	8–12
Set Counter Value	Установка начального значения счетчика	8–14
Set_Reset Flip Flop	Триггер SR	8–31
Shift Left Double Word	Сдвиг влево двойного слова	16–2
Shift Left Word	Сдвиг влево слова	16–2
Shift Right Double Integer	Сдвиг вправо двойного целого числа	16–2
Shift Right Double Word	Сдвиг вправо двойного слова	16–2
Shift Right Integer	Сдвиг вправо целого числа	16–2
Shift Right Word	Сдвиг вправо слова	16–2
Subtract Double Integer	Вычитание двойных целых чисел (32 бита)	11–5
Subtract Integer	Вычитание целых чисел (16 бит)	11–4
Subtract Real	Вычитание вещественных чисел	12–4

Truncate Double Integer Part	Усечение до двойного целого числа	14–15
Twos Complement Double Integer	Дополнение двойного целого числа до двух	14–12
Twos Complement Integer	Дополнение целого числа до двух	14–11
Up Counter (операция со счетчиком)	Прямой счет	10–5
Up Counter (битовая логическая операция)	Прямой счет	8–15
Up–Down Counter	Прямой/обратный счет	10–3
(Word) AND Double Word	Поразрядное И над двойными словами	15–4
(Word) AND Word	Поразрядное И над словами	15–3
(Word) Exclusive OR Double Word	Поразрядное исключающее ИЛИ над двойными словами	15–8
(Word) Exclusive OR Word	Поразрядное исключающее ИЛИ над словами	15–7
(Word) OR Double Word	Поразрядное ИЛИ над двойными словами	15–6
(Word) OR Word	Поразрядное ИЛИ над словами	15–5

А.3 Список команд на русском языке с краткими обозначениями SIMATIC

Таблица А–3 содержит список команд FUP, упорядоченный в порядке русского алфавита, с соответствующими сокращенными обозначениями или мнемоникой SIMATIC и ссылкой на страницу, на которой описана команда.

Таблица А–3. Команды FUP, упорядоченные в соответствии с русским алфавитом, с сокращенными обозначениями SIMATIC

Полное имя	Краткое обозначение SIMATIC	Страница
Активизация Master Control Relay	MCRA	20–10
Бит ошибки “Двоичный результат”	BIE	19–3
Бит ошибки “Неупорядочено”	UO	19–6
Бит ошибки “Переполнение”	OV	19–7
Бит ошибки “Переполнение с запоминанием”	OS	19–8
Бит результата в случае неравенства 0	<>0	19–4
Бит результата в случае равенства 0	=0	19–4
Бит результата в случае >0	>0	19–4
Бит результата в случае >=0	>=0	19–4
Бит результата в случае <0	<0	19–4
Бит результата в случае <=0	<=0	19–4
Включение Master Control Relay	MCR<	20–13
Возврат	RET	20–8
Вызов системного FB	CALL_SFB	20–4
Вызов системной FC	CALL_SFC	20–4
Вызов FB	CALL_FB	20–4
Вызов FC	CALL_FC	20–4
Вызов FC/SFC без параметров	CALL	20–2
Выключение Master Control Relay	MCR>	20–13
Вычитание вещественных чисел	SUB_R	12–4
Вычитание двойных целых чисел (32 бита)	SUB_DI	0–5
Вычитание целых чисел (16 бит)	SUB_I	11–4
Деактивизация Master Control Relay	MCRD	20–10
Деление вещественных чисел	DIV_R	12–6
Деление двойных целых чисел (32 бита)	DIV_DI	11–9
Деление целых чисел (16 бит)	DIV_I	11–8
Добавление двоичного входа	---	8–7
Дополнение двойного целого числа (32 бита) до двух	NEG_DI	14–12
Дополнение двойного целого числа (32 бита) до единицы	INV_DI	14–10
Дополнение целого числа (16 бит) до двух	NEG_I	14–11
Дополнение целого числа (16 бит) до единицы	INV_I	14–9
Изменение знака вещественного числа	NEG_R	14–13
Импульсный таймер (SI) (битовая логическая операция)	SI	8–17
Импульсный таймер (SI) (таймерная операция)	S_IMPULS	9–5
Коннектор	#	8–10

Логическая операция И	&	8–3
Логическая операция ИЛИ	>=1	8–4
Логическая операция исключающее ИЛИ	XOR	8–6
Обнаружение отрицательного фронта по адресу	NEG	8–30
Обнаружение положительного фронта по адресу	POS	8–29
Обнаружение отрицательного фронта RLO	N	8–28
Обнаружение положительного фронта RLO	P	8–27
Образование абсолютного значения числа с плавающей точкой	ABS	12–8
Образование арккосинуса числа с плавающей точкой	ACOS	12–13
Образование арксинуса числа с плавающей точкой	ASIN	12–13
Образование арктангенса числа с плавающей точкой	ATAN	12–13
Образование квадрата числа с плавающей точкой	SQR	12–9
Образование квадратного корня из числа с плавающей точкой	SQRT	12–10
Образование косинуса числа с плавающей точкой	COS	12–13
Образование натурального логарифма числа с плавающей точкой	LN	12–11
Образование синуса числа с плавающей точкой	SIN	12–13
Образование тангенса числа с плавающей точкой	TAN	12–13
Образование экспоненциального значения числа с плавающей точкой	EXP	12–12
Обратный счет (битовая логическая операция)	ZR	8–16
Обратный счет (операция со счетчиком)	Z_RUECK	10–7
Округление до ближайшего большего целого числа	CEIL	14–16
Округление до ближайшего меньшего целого числа	FLOOR	14–17
Округление до двойного целого числа	ROUND	14–14
Открытие блока данных	AUF	17–2
Отрицание двоичного входа	---o	8–8
Передача значения	MOVE	14–2
Переход, если 0	JMPN	18–5
Переход, если 1	JMP	18–3
Получение остатка от деления двойных целых чисел	MOD	11–10
Поразрядное ИЛИ над двойными словами	WOR_DW	15–6
Поразрядное ИЛИ над словами	WOR_W	15–5
Поразрядное И над двойными словами	WAND_DW	15–4
Поразрядное И над словами	WAND_W	15–3
Поразрядное исключающее ИЛИ над двойными словами	WXOR_DW	15–8
Поразрядное исключающее ИЛИ над словами	WXOR_W	15–7
Преобразование двойного целого числа (32 бита) в число в формате BCD	DI_BCD	14–7
Преобразование двойного целого числа (32 бита) в число с плавающей точкой	DI_R	14–8
Преобразование числа в формате BCD в двойное целое число (32 бита)	BCD_DI	14–6
Преобразование числа в формате BCD в целое число (16 бит)	BCD_I	14–3
Преобразование целого числа (16 бит) в двойное целое число (32 бита)	I_DI	14–5
Преобразование целого числа (16 бит) в число в формате BCD	I_BCD	14–4

Присваивание	=	8–9
Прямой/обратный счет	ZAEHLER	10–3
Прямой счет (битовая логическая операция)	ZV	8–15
Прямой счет (операция со счетчиком)	Z_VORW	10–5
Сброс выхода	R	8–13
Сдвиг двойного слова влево	SHL_DW	16–2
Сдвиг двойного слова вправо	SHR_DW	16–2
Сдвиг двойного целого числа (32 бита) вправо	SHR_DI	16–2
Сдвиг слова влево	SHL_W	16–2
Сдвиг слова вправо	SHR_W	16–2
Сдвиг целого числа (16 бит) вправо	SHR_I	16–2
Сложение вещественных чисел	ADD_R	12–3
Сложение двойных целых чисел (32 Bit)	ADD_DI	11–3
Сложение целых чисел (16 бит)	ADD_I	11–2
Сохранение RLO в бите BR	SAVE	8–11
Сравнение вещественных чисел	CMP >=R	13–6
Сравнение двойных целых чисел (32 бита)	CMP >=D	13–3
Сравнение целых чисел (16 бит)	CMP >=I	13–2
Таймер с задержкой включения (SE) (битовая логическая операция)	SE	8–21
Таймер с задержкой включения (SE) (таймерная команда)	S_EVERZ	9–9
Таймер с задержкой включения с запоминанием (SS) (битовая логическая операция)	SS	
Таймер с задержкой включения с запоминанием (SS) (таймерная команда)	S_SEVERZ	9–11
Таймер с задержкой выключения (SA) (битовая логическая операция)	SA	8–25
Таймер с задержкой выключения (SA) (таймерная команда)	S_AVERZ	9–13
Таймер с продленным импульсом (SV) (битовая логическая операция)	SV	8–19
Таймер с продленным импульсом (SV) (таймерная команда)	S_VIMP	9–7
Триггер RS	RS	8–32
Триггер SR	SR	8–31
Умножение вещественных чисел	MUL_R	12–5
Умножение двойных целых чисел (32 бита)	MUL_DI	11–7
Умножение целых чисел (16 бит)	MUL_I	11–6
Усечение до двойного целого числа	TRUNC	14–15
Установка выхода	S	8–12
Установка начального значения счетчика	SZ	8–14
Циклический сдвиг двойного слова влево	ROL_DW	16–10
Циклический сдвиг двойного слова вправо	ROR_DW	16–10

А.4 Список команд на русском языке и международные (английские) эквиваленты

Таблица А–4 содержит алфавитный список команд FUP на русском языке с международными (английскими) эквивалентами и ссылкой на страницу, на которой описана команда.

Таблица А–4. Команды FUP, упорядоченные в соответствии с русским алфавитом, с международными (английскими) эквивалентами

Полное имя	Международный (английский) эквивалент	стр.
Активизация Master Control Relay	Master Control Relay Activate	20–10
Бит ошибки “Двоичный результат”	Exception Bit Binary Result	19–3
Бит ошибки “Неупорядочено”	Exception Bit Unordered	19–6
Бит ошибки “Переполнение”	Exception Bit Overflow	19–7
Бит ошибки “Переполнение с запоминанием”	Exception Bit Overflow Stored	19–8
Бит результата в случае неравенства 0	Result Bit Not Equal 0	19–4
Бит результата в случае равенства 0	Result Bit Equal 0	19–4
Бит результата в случае >0	Result Bit Greater Than 0	19–4
Бит результата в случае ≥ 0	Result Bit Greater Equal 0	19–4
Бит результата в случае <0	Result Bit Less Than 0	19–4
Бит результата в случае ≤ 0	Result Bit Less Equal 0	19–4
Включение Master Control Relay	Master Control Relay On	20–13
Возврат	Return	20–8
Вызов системного FB	Call System FB from Box	20–4
Вызов системной FC	Call System FC from Box	20–4
Вызов FB	Call FB from Box	20–4
Вызов FC	Call FC from Box	20–4
Вызов FC/SFC без параметров	Call FC SFC (without parameters)	20–2
Выключение Master Control Relay	Master Control Relay Off	20–13
Вычитание вещественных чисел	Subtract Real	12–4
Вычитание двойных целых чисел (32 бита)	Subtract Double Integer	11–5
Вычитание целых чисел (16 бит)	Subtract Integer	11–4
Деактивизация Master Control Relay	Master Control Relay Deactivate	20–10
Деление вещественных чисел	Divide Real	12–6
Деление двойных целых чисел (32 бита)	Divide Double Integer	11–9
Деление целых чисел (16 бит)	Divide Integer	11–8
Добавление двоичного входа	Insert Binary Input	8–7
Дополнение двойного целого числа (32 бита) до единицы	Ones Complement Double Integer	14–10
Дополнение двойного целого (32 бита) числа до двух	Twos Complement Double Integer	14–12
Дополнение целого числа (16 бит) до единицы	Ones Complement Integer	14–9
Дополнение целого числа (16 бит) до двух	Twos Complement Integer	14–11
Изменение знака вещественного числа	Negate Real Number	14–13
Импульсный таймер (битовая логическая)	Pulse Timer	8–17

операция)		
Импульсный таймер (таймерная команда)	Pulse S5 Timer	9–5
Коннектор	Midline Output	8–10
Логическая операция И	AND	8–3
Логическая операция ИЛИ	OR	8–4
Логическая операция Исключающее ИЛИ	Exclusive OR	8–6
Обнаружение отрицательного фронта по адресу	Address Negative Edge Detection	8–30
Обнаружение положительного фронта по адресу	Address Positive Edge Detection	8–29
Обнаружение отрицательного фронта RLO	Negative RLO Edge Detection	8–28
Обнаружение положительного фронта RLO	Positive RLO Edge Detection	8–27
Образование абсолютного значения числа с плавающей точкой	Form the Absolute Value of a Floating-Point Number	12–8
Образование арккосинуса числа с плавающей точкой	Form the Arc Cosine of a Floating-Point Number	12–13
Образование арксинуса числа с плавающей точкой	Form the Arc Sine of a Floating-Point Number	12–13
Образование арктангенса числа с плавающей точкой	Form the Arc Tangent of a Floating-Point Number	12–13
Образование косинуса числа с плавающей точкой	Form the Cosine of a Floating-Point Number	12–13
Образование квадрата числа с плавающей точкой	Form the Square of a Floating-Point Number	12–9
Образование квадратного корня числа с плавающей точкой	Form the Square Root of a Floating-Point Number	12–10
Образование натурального логарифма числа с плавающей точкой	Form the Natural Algorithm of a Floating-Point Number	12–11
Образование синуса числа с плавающей точкой	Form the Sine of a Floating-Point Number	12–13
Образование тангенса числа с плавающей точкой	Form the Tangent of a Floating-Point Number	12–13
Образование экспоненциального значения числа с плавающей точкой	Form the Exponential Value of a Floating-Point Number	12–12
Обратный счет (битовая логическая операция)	Down Counter	8–16
Обратный счет (операция со счетчиком)	Down Counter	10–7
Округление до ближайшего большего целого числа	Ceiling	14–16
Округление до ближайшего меньшего целого числа	Floor	14–17
Округление до двойного целого	Round to Double Integer	14–14
Открытие блока данных	Open Data Block: DB или DI	17–2
Отрицание двоичного входа	Negate Binary Input	8–8
Передача значения	Assign a Value	14–2
Переход, если 0	Jump–If–Not	18–5
Переход, если 1	Jump	18–3
Получение остатка от деления двойных целых чисел	Return Fraction Double Integer	11–10
Поразрядное ИЛИ над двойными словами	(Word) OR Double Word	15–6
Поразрядное ИЛИ над словами	(Word) OR Word	15–5
Поразрядное И над двойными словами	(Word) AND Double Word	15–4
Поразрядное И над словами	(Word) AND Word	15–3
Поразрядное исключающее ИЛИ над двойными словами	(Word) Exclusive OR Double Word	15–8

Поразрядное исключающее ИЛИ над словами	(Word) Exclusive OR Word	15–7
Преобразование двойного целого числа в число в формате BCD	Double Integer to BCD	14–7
Преобразование двойного целого числа в число с плавающей точкой	Double Integer to Real	14–8
Преобразование целого числа (16 бит) в двойное целое число (32 бита)	Integer to Double Integer	14–5
Преобразование целого числа (16 бит) в число в формате BCD	Integer to BCD	14–4
Преобразование числа в формате BCD в двойное целое число (32 бита)	BCD to Double Integer	14–6
Преобразование числа в формате BCD в целое число (16 бит)	BCD to Integer	14–3
Присваивание	Assign	8–9
Прямой счет (битовая логическая операция)	Up Counter	8–15
Прямой счет (операция со счетчиком)	Up Counter	10–5
Прямой/обратный счет	Up–Down Counter	10–3
Сброс выхода	Reset Output	8–13
Сдвиг влево двойного слова	Shift Left Double Word	16–2
Сдвиг влево слова	Shift Left Word	16–2
Сдвиг вправо двойного слова	Shift Right Double Word	16–2
Сдвиг вправо двойного целого числа	Shift Right Double Integer	16–2
Сдвиг вправо слова	Shift Right Word	16–2
Сдвиг вправо целого числа	Shift Right Integer	16–2
Сохранение RLO в бите BR	Save RLO to BR Memory	8–11
Сложение вещественных чисел	Add Real	12–3
Сложение двойных целых чисел (32 бита)	Add Double Integer	11–3
Сложение целых чисел (16 бит)	Add Integer	11–2
Сравнение вещественных чисел	Compare Real (>, <, ==, <=, >=)	13–4
Сравнение двойных целых чисел (32 бита)	Compare Double Integer (>, <, ==, <=, >=)	13–3
Сравнение целых чисел (16 бит)	Compare Integer (>, <, ==, <=, >=)	13–2
Таймер с задержкой включения (битовая логическая операция)	On–Delay Timer	8–21
Таймер с задержкой включения с запоминанием (битовая логическая операция)	Retentive On–Delay Timer	8–23
Таймер с задержкой включения (таймерная команда)	On–Delay S5 Timer	9–9
Таймер с задержкой включения с запоминанием (таймерная команда)	Retentive On–Delay S5 Timer	9–11
Таймер с задержкой выключения (битовая логическая операция)	Off–Delay Timer	8–25
Таймер с задержкой выключения (таймерная команда)	Off–Delay S5 Timer	9–13
Таймер с продленным импульсом (битовая логическая операция)	Extended Pulse Timer	8–19
Таймер с продленным импульсом (таймерная операция)	Extended Pulse S5 Timer	9–7

Триггер RS	Reset_Set Flip Flop	8–32
Триггер SR	Set_Reset Flip Flop	8–31
Умножение вещественных чисел	Multiply Real	12–5
Умножение двойных целых чисел (32 бита)	Multiply Double Integer	11–7
Умножение целых чисел (16 бит)	Multiply Integer	11–6
Усечение до двойного целого числа	Truncate Double Integer Part	14–15
Установка выхода	Set Output	8–12
Установка начального значения счетчика	Set Counter Value	8–14
Циклический сдвиг влево двойного слова	Rotate Left Double Word	16–10
Циклический сдвиг вправо двойного слова	Rotate Right Double Word	16–10

Таблица А–5. Команды FUP, приведенные в данном руководстве, с их международными полными и сокращенными именами и с сокращенными именами SIMATIC

Международное полное имя	Международное краткое имя	Сокращенное имя SIMATIC	стр.
Down Counter (операция со счетчиком)	S_CD	Z_RUECK	10–7
Down Counter (битовая логическая операция)	CD	ZR	8–16
Exception Bit Binary Result	BR	BIE	19–3
Extended Pulse S5 Timer (таймерная команда)	S_PEXT	S_VIMP	9–7
Extended Pulse Timer (битовая логическая операция)	SE	SV	8–19
Off-Delay S5 Timer (таймерная команда)	S_OFFDT	S_AVERZ	9–13
Off-Delay Timer (битовая логическая операция)	SF	SA	8–25
On-Delay S5 Timer (таймерная команда)	S_ODT	S_EVERZ	9–9
On-Delay Timer (битовая логическая операция)	SD	SE	8–21
Open Data Block: DB или DI	OPN	AUF	17–2
Pulse S5 Timer (таймерная команда)	S_PULSE	S_IMPULS	9–5
Pulse Timer (битовая логическая операция)	SP	SI	8–17
Retentive On-Delay S5 Timer (таймерная команда)	S_ODTS	S_SEVERZ	9–11
Retentive On-Delay Timer (битовая логическая операция)	SS	SS	8–23
Set Counter Value	SC	SZ	8–14
Up Counter (операция со счетчиком)	S_CU	Z_VORW	10–5
Up Counter (битовая логическая операция)	CU	ZV	8–15
Up-Down Counter	S_CUD	ZAHLER	10–3

Примеры программирования

В

В.1	Обзор	В-2
В.2	Битовые логические операции	В-3
В.3	Таймерные команды	В-7
В.4	Операции счета и сравнения	В-11
В.5	Арифметические операции с целыми числами	В-13
В.6	Поразрядные логические операции со словами	В-14

В.1 Обзор

Практические применения

Каждая команда FUP, описанная в данном руководстве выполняет определенную функцию. Комбинируя эти команды в программе, Вы можете реализовать широкий спектр задач. Эта глава предлагает следующие примеры практического применения команд FUP:

- Управление транспортером с использованием битовых логических операций
- Обнаружение направления движения ленты транспортера с использованием битовых логических операций
- Генерирование тактовых импульсов с помощью таймерных команд
- Слежение за пространством для хранения с помощью операций счета и сравнения
- Решение задачи с использованием арифметических операций с целыми числами
- Установка интервала времени для нагревания печи

Используемые команды

Примеры в этой главе используют следующие команды:

- Сложение целых чисел (ADD_I)
- Логическая операция И (AND)
- Присваивание (=)
- Передача значения (MOVE)
- Сравнение целых чисел (CMP_I>=)
- Сравнение целых чисел (CMP_I<=)
- Деление целых чисел (DIV_I)
- Обратный счет (S_CD)
- Таймер с продленным импульсом (SE)
- Переход, если 0 (JMPN)
- Умножение целых чисел (MUL_I)
- Логическая операция ИЛИ (OR)
- Обнаружение положительного фронта RLO (P)
- Сброс выхода (R)
- Возврат (RET)
- Установка выхода (S)
- Прямой счет (S_CU)
- Поразрядное И со словами (WAND_W)
- Поразрядное ИЛИ со словами (WOR_W)

В.2 Битовые логические операции

Управление лентой транспортера

На рис. В–1 показана лента транспортера, которая может приводиться в движение с помощью электричества. В начале транспортера имеются две кнопки: S1 для запуска и S2 для останова. В конце транспортера тоже имеются две кнопки: S3 для запуска и S4 для останова. Транспортер можно запускать или останавливать с любого конца. Датчик S5 останавливает транспортер, когда предмет, находящийся на ленте, достигает конца.

Символическое программирование

Вы можете программировать управление лентой транспортера, показанного на рис. В–1, используя символы, представляющие различные компоненты конвейерной системы. Если Вы выберете этот метод, то Вам потребуется создать таблицу символов для установления связи между выбранными Вами символами и абсолютными адресами (см. табл. В–1). В этой таблице Вы определяете символы (см. *Руководство пользователя* /231/).

Таблица В–1. Элементы символического программирования для конвейерной системы

Компонент системы	Абсолютный адрес	Символ	Таблица символов
Кнопка “Пуск”	И.1	S1	И.1 S1
Кнопка “Стоп”	И.2	S2	И.2 S2
Кнопка “Пуск”	И.3	S3	И.3 S3
Кнопка “Стоп”	И.4	S4	И.4 S4
Датчик	И.5	S5	И.5 S5
Двигатель	Q4.0	MOTOR_ON	Q4.0 MOTOR_ON

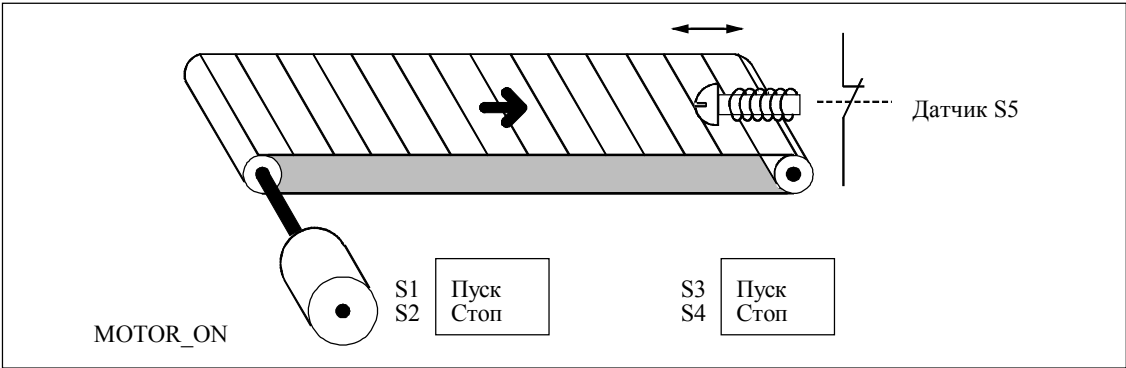


Рис. В-1. Конвейерная система

Абсолютное программирование

Вы можете написать программу для управления лентой транспортера, показанного на рис. В–1, используя абсолютные значения, представляющие различные компоненты конвейерной системы (см. табл. В–2). На рис. В–2 показана программа на языке программирования FUP для управления лентой транспортера.

Таблица В–2. Элементы абсолютного программирования для конвейерной системы

Компонент системы	Абсолютный адрес
Кнопка “Пуск”	П.1
Кнопка “Стоп”	П.2
Кнопка “Пуск”	П.3
Кнопка “Стоп”	П.4
Датчик	П.5
Двигатель	Q4.0

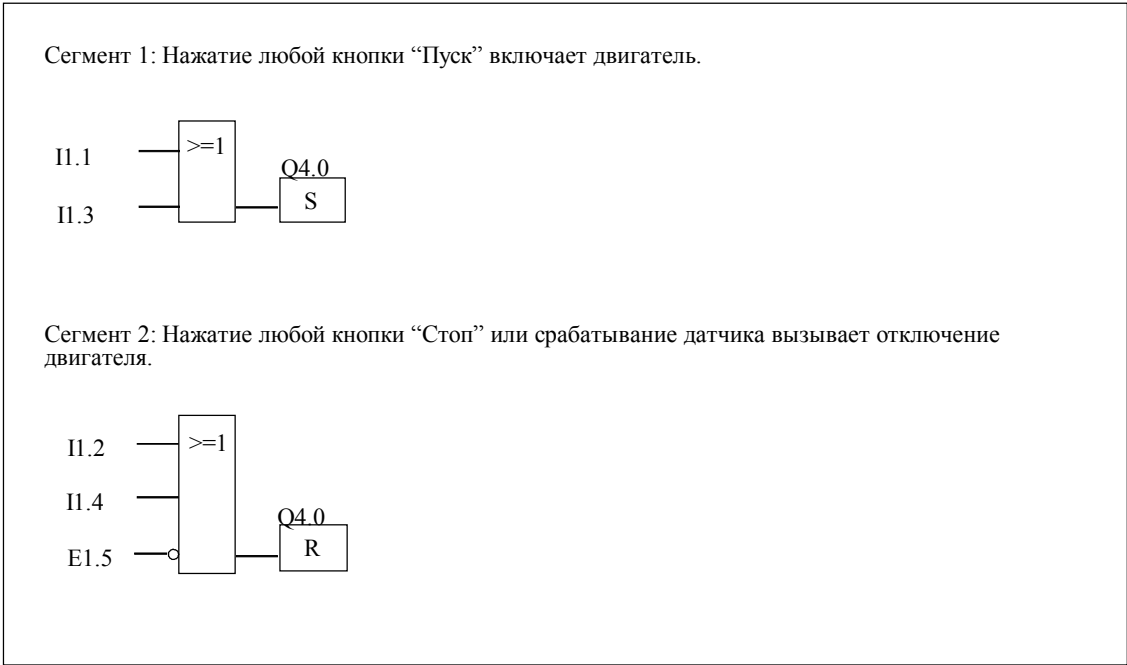


Рис. В-2. Функциональный план для управления лентой транспортера

Определение направления движения ленты транспортера

На рис. В–3 показана лента транспортера, снабженная двумя фотоэлектрическими датчиками (РЕВ1 И РЕВ2), которые спроектированы для определения направления, в котором движется пакет, расположенный на ленте.

Символическое программирование

Вы можете написать программу, активизирующую отображение направления движения ленты транспортера конвейерной системы, показанной на рис. В–3, используя символы, которые представляют различные компоненты конвейерной системы, включая фотоэлектрические датчики для определения направления. Если Вы выберете этот метод, то Вам потребуется создать таблицу символов для установления связи между выбранными Вами символами и абсолютными адресами (см. табл. В–3). В этой таблице Вы определяете символы (см. *Руководство пользователя /231/*).

Таблица В–3. Элементы символического программирования для определения направления

Компонент системы	Абсолютный адрес	Символ	Таблица символов
Фотоэлектрический датчик 1	I0.0	PEB1	I0.0 PEB1
Фотоэлектрический датчик 2	I0.1	PEB2	I0.1 PEB2
Индикатор движения направо	Q4.0	RIGHT	Q4.0 RIGHT
Индикатор движения налево	Q4.1	LEFT	Q4.1 LEFT
Тактовый меркер 1	M0.0	PM1	M0.0 PM1
Тактовый меркер 2	M0.1	PM2	M0.1 PM2

Абсолютное программирование

Вы можете написать программу для активизации отображения направления движения ленты транспортера, показанного на рис. В–3, используя абсолютные значения, представляющие фотоэлектрические датчики для определения направления (см. табл. В–4). На рис. В–4 показана программа на языке программирования FUP для управления отображением направления движения лентой транспортера.

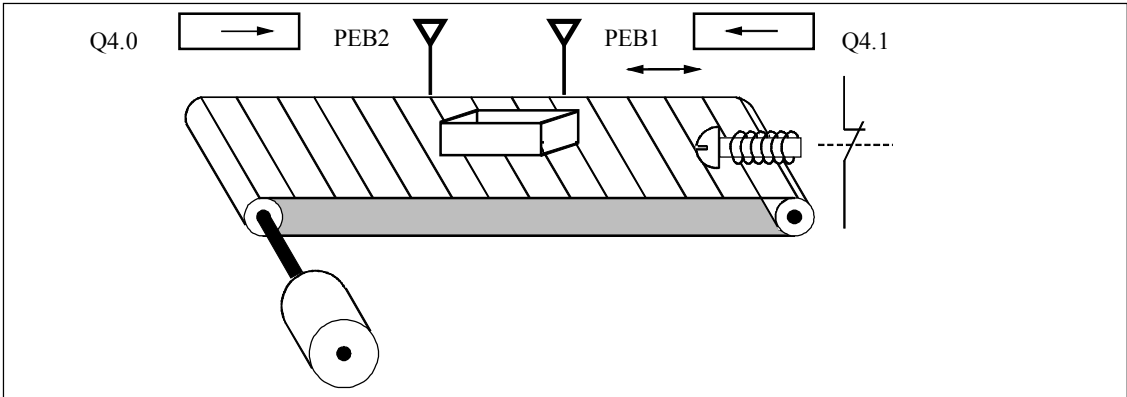


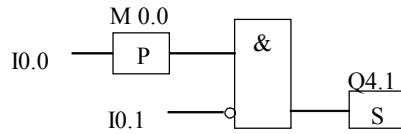
Рис. В-3. Конвейерная система с фотоэлектрическими датчиками для определения направления

Таблица В–4. Элементы абсолютного программирования для определения направления

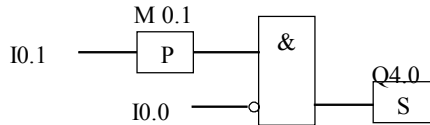
Компонент системы	Абсолютный адрес
Фотоэлектрический датчик 1	I0.0
Фотоэлектрический датчик 2	I0.1
Индикатор движения направо	Q4.0
Индикатор движения налево	Q4.1
Тактовый меркер 1	M0.0
Тактовый меркер 2	M0.1



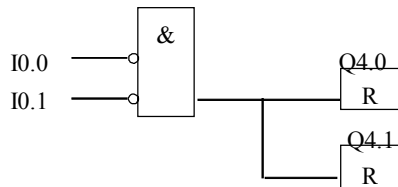
Сегмент 1: Если на входе I0.0 сигнал изменяется с 0 на 1 (нарастающий фронт), и при этом состояние сигнала на входе I0.1 равно 0, то пакет на ленте транспортера движется влево.



Сегмент 2: Если на входе I0.1 сигнал изменяется с 0 на 1 (нарастающий фронт), и при этом состояние сигнала на входе I0.0 равно 0, то пакет на ленте транспортера движется вправо.



Сегмент 3. Если один из фотодатчиков прерывается, то это значит, что пакет находится между датчиками.



В.3 Таймерные команды

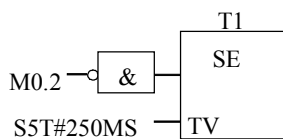
Генератор тактовых импульсов

Для создания периодически повторяющегося сигнала Вы можете использовать генератор тактовых импульсов или реле мигания. Генераторы тактовых импульсов обычно используются в системах сигнализации, управляющих миганием индикаторных ламп.

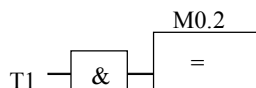
Если Вы используете S7-300, то Вы можете реализовать функцию генератора тактовых импульсов, применив исполнение программы, управляемое временем, в специальных организационных блоках. Пример, показанный в следующей программе FUP, иллюстрирует использование таймерных функций для генерации тактовых импульсов.

Следующий пример показывает, как реализуется тактовый генератор в режиме свободных колебаний с помощью таймера (скважность импульсов 1:1). Значения частоты приведены в таблице В -5.

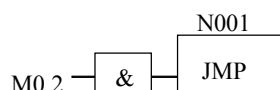
Сегмент 1: Если состояние сигнала таймера T1 равно 0, загрузить значение времени 250 мс в T1, запустить T1 как таймер с продленным импульсом.



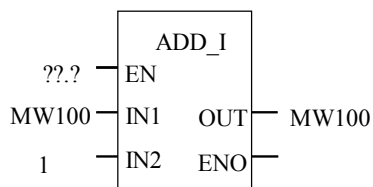
Сегмент 2: Состояние таймера временно сохраняется во вспомогательном меркере.



Сегмент 3: Если состояние сигнала таймера T1 равно 1, перейти на метку N001.



Сегмент 4: Когда время таймера T1 истекает, меркерное слово 100 увеличивается на 1.



Сегмент 5: Команда MOVE дает возможность выводить различные частоты импульсов на выходы от Q12.0 до Q13.7.

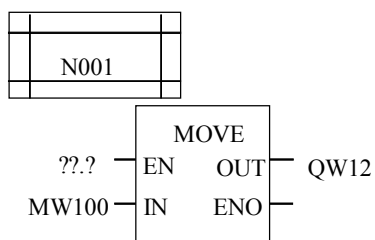


Рис. В-5. Функциональный план для генерирования тактовых импульсов

Опрос сигнала таймера T1 определяет результат логической операции (RLO, см. раздел 6.3), показанный на рис. В-6.

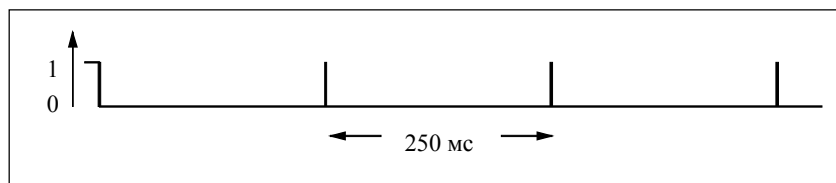


Рис. В-6. RLO для инверсного входного параметра AN T1 в примере генератора тактовых импульсов

Как только время истекает, таймер перезапускается. Вследствие этого опрос сигнала, выполняемый командой AN M0.2, выдает состояние сигнала "1" лишь кратковременно

На рис. В-7 показан инвертированный бит RLO.

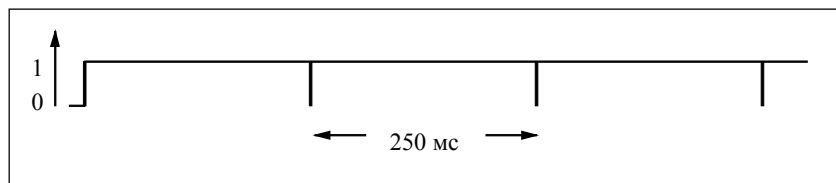


Рис. В-7. Инвертированный бит RLO таймера T1 в примере тактового генератора

Каждые 250 мс бит RLO равен 0. Переход игнорируется, и содержимое меркерного слова MW100 увеличивается на 1.

Достижение определенной частоты

В таблице В-5 перечислены частоты, которые Вы можете получить из отдельных битов меркерных байтов MB101 и MB100. Сегмент 5 в функциональном плане, показанном на рис. В-5, иллюстрирует, как команда MOVE позволяет увидеть различные частоты импульсов на выходах от Q12.0 до Q13.7.

Таблица В-5. Частоты для примера генератора тактовых импульсов

Биты в MB101/MB100	Частота в Гц	Длительность
M101.0	2.0	0,5 с (250 мс вкл/250 мс выкл)
M101.1	1.0	1 с (0,5 с вкл/0,5 с выкл)
M101.2	0.5	2 с (1 с вкл/1 с выкл)
M101.3	0.25	4 с (2 с вкл/2 с выкл)
M101.4	0.125	8 с (4 с вкл/4 с выкл)
M101.5	0.0625	16 с (8 с вкл/8 с выкл)
M101.6	0.03125	32 с (16 с вкл/16 с выкл)
M101.7	0.015625	64 с (32 с вкл/32 с выкл)
M100.0	0.0078125	128 с (64 с вкл/64 с выкл)
M100.1	0.0039062	256 с (128 с вкл/128 с выкл)
M100.2	0.0019531	512 с (256 с вкл/256 с выкл)
M100.3	0.0009765	1024 с (512 с вкл/512 с выкл)
M100.4	0.0004882	2048 с (1024 с вкл/1024 с выкл)
M100.5	0.0002441	4096 с (2048 с вкл/2048 с выкл)
M100.6	0.000122	8192 с (4096 с вкл/4096 с выкл)
M100.7	0.000061	16384 с (8192 с вкл/8192 с выкл)

В табл. В-6 перечислены состояния сигнала битов меркерного байта MB101. На рис. В-8 показано состояние сигнала меркера M101.1.

Таблица В-6. Состояния сигнала битов меркерного байта MB101

Цикл	Состояния сигнала битов меркерного байта MB101								Время в мс
	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	250
1	0	0	0	0	0	0	0	1	250
2	0	0	0	0	0	0	1	0	250
3	0	0	0	0	0	0	1	1	250
4	0	0	0	0	0	1	0	0	250
5	0	0	0	0	0	1	0	1	250
6	0	0	0	0	0	1	1	0	250
7	0	0	0	0	0	1	1	1	250
8	0	0	0	0	1	0	0	0	250
9	0	0	0	0	1	0	0	1	250
10	0	0	0	0	1	0	1	0	250
11	0	0	0	0	1	0	1	1	250
12	0	0	0	0	1	1	0	0	250

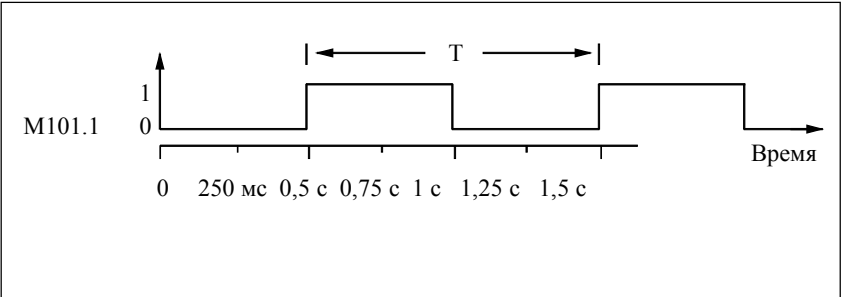


Рис. В-8. Состояние сигнала бита 1 меркерного байта MB101 (M101.1)

В.4 Операции счета и сравнения

Зона складирования со счетчиком и компаратором

На рис. В-9 показана система с двумя конвейерами и зоной временного складирования между ними. Конвейер 1 поставляет пакеты в зону складирования. Фотоэлектрический датчик в конце конвейера 1 вблизи зоны складирования определяет, сколько пакетов доставлено в эту зону. Конвейер 2 транспортирует пакеты из зоны временного складирования к погрузочной площадке, где пакеты грузятся на грузовые автомобили для доставки клиентам. Фотоэлектрический датчик в конце конвейера 2 вблизи зоны складирования определяет, сколько пакетов покидают зону складирования для отправки на погрузочную площадку.

Информационное табло с пятью лампами отображает уровень заполнения зоны временного складирования. На рис. В-10 показана программа FUP, активизирующая индикаторные лампы на информационном табло.

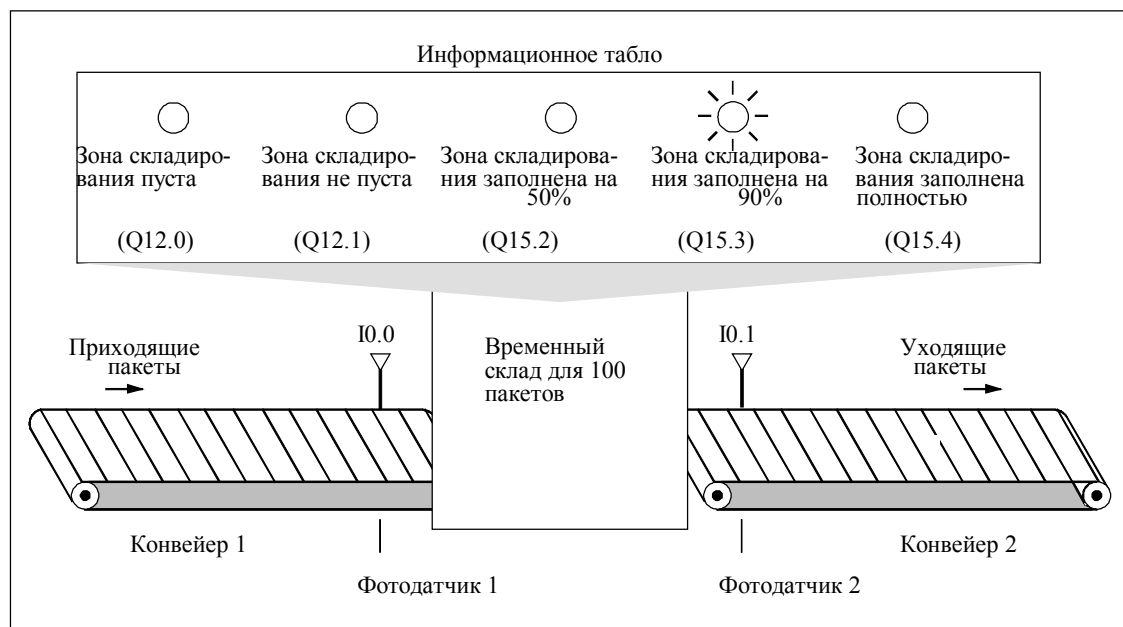
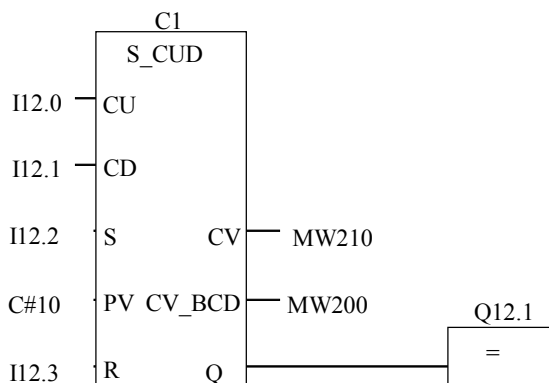
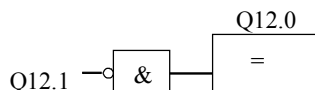


Рис. В-9. Зона складирования со счетчиком и компаратором

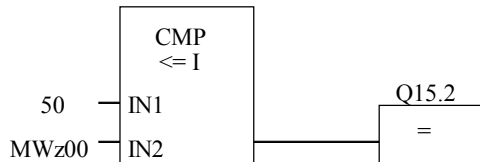
Сегмент 1: Счетчик C1 увеличивает значение при каждом изменении сигнала с 0 на 1 на входе CU и уменьшает значение при каждом изменении сигнала с 1 на 0 на входе CD. При изменении сигнала с 0 на 1 на входе S счетчик принимает значение PV. Изменение сигнала с 0 на 1 на входе R сбрасывает значение счетчика в 0. MW200 содержит текущее значение C1. Q12.1 показывает, что зона складирования не пуста.



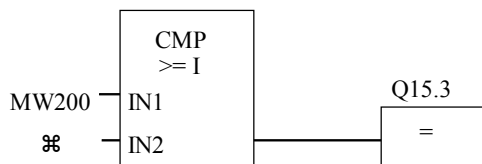
Сегмент 2: Q12.0 показывает, что зона складирования пуста



Сегмент 3: Если 50 меньше или равно значению счетчика (иными словами, если текущее значение счетчика больше 50), то загорается индикаторная лампа “Зона складирования заполнена на 50%”



Сегмент 4: Если значение счетчика больше или равно 90, то горит индикаторная лампа “Зона складирования заполнена на 90%”.



Сегмент 5: Если значение счетчика больше или равно 100, то горит индикаторная лампа “Зона складирования заполнена полностью”. Используйте выход Q14.4 для блокирования конвейера 1.

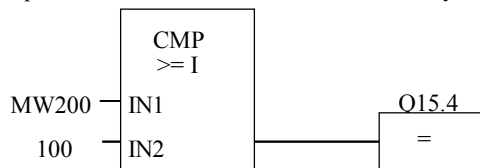


Рис. В-10. Функциональный план для активизации индикаторных ламп на информационном табло

В.5 Арифметические операции с целыми числами

Следующий пример программы показывает, как использовать арифметические операции с целыми числами и команды L и T для получения того же результата, который дает следующее уравнение:

$$MW4 = \frac{(IW0 + DBW3) \times 15}{MW0}$$

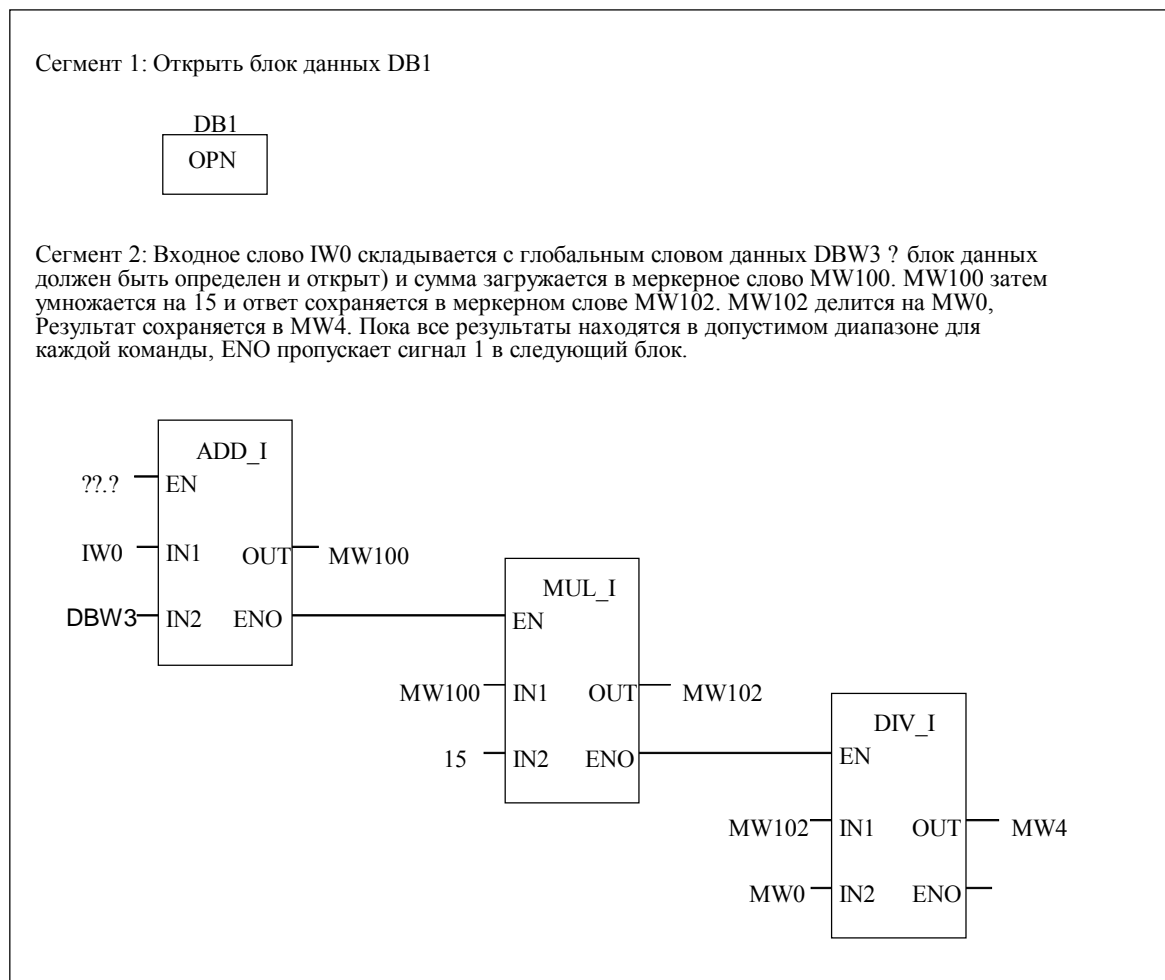
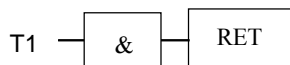


Рис. В-11. Функциональный план для арифметических операций с целыми числами

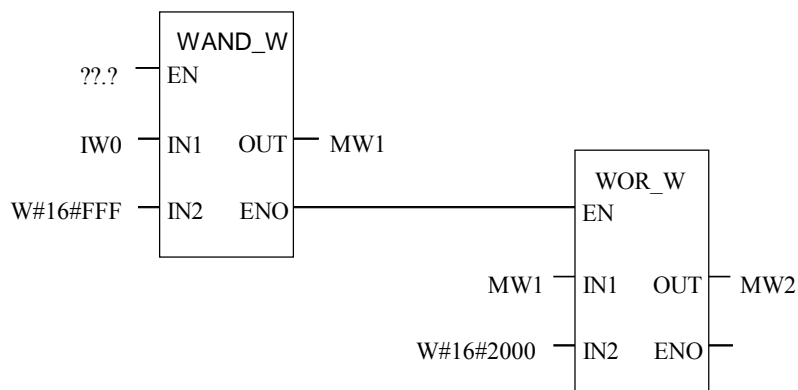
Сегмент 1: Если таймер работает, то включить нагрев. Если таймер работает, то команда “Возврат” заканчивает обработку здесь.



Сегмент 2: Если таймер работает, то команда “Возврат” заканчивает обработку здесь.



Сегмент 3: Замаскировать биты с I0.4 по I0.7 ? т.е. сбросить их в 0). Эти биты входов переключателя не используются. 16 битов входов переключателя комбинируются с W#16#0FFF в соответствии с командой “Поразрядное И со словами”. Результат загружается в меркерное слово MW1. Чтобы установить базу времени в секундах, предустановленное значение комбинируется с W#16#2000 в соответствии с командой “Поразрядное ИЛИ со словами”, устанавливая бит 13 в 1 и сбрасывая бит 12 в 0.



Сегмент 4: Запустить таймер T1 в режиме продленного импульса, если нажата кнопка “Пуск”, загружая в качестве предустановленного значения меркерное слово MW2 ? полученное из предыдущих логических операций).

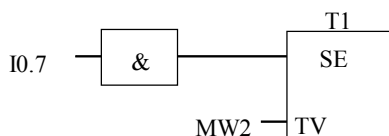


Рис. В-13. Функциональный план для нагрева печи

Система представления чисел

C

C.1	Система представления чисел	C-2
-----	-----------------------------	-----

С.1 Система представления чисел

Общая информация

Команды FUP работают с объектами данных определенных размеров (см. таблицу С-2). Например, битовые логические команды выполняют операции над двоичными цифрами (битами); команды передачи выполняют операции с байтами, словами и двойными словами.

Математические команды также выполняют операции с байтами, словами и двойными словами. В адресах этих байтов, слов и двойных слов Вы можете закодировать числа в различных форматах, таких как целые и вещественные.

Если Вы используете символическую адресацию, Вы определяете символы и указываете тип данных для каждого из этих символов (см. таблицу С-2). Различные типы данных имеют разные варианты выбора формата и записи чисел. Информация в следующих разделах поможет Вам понять форматы и систему записи чисел.

Эта глава руководства описывает только некоторое из возможных представлений чисел и констант.

Таблица С-1. Форматы чисел и констант, не описанные в этой главе

Формат	Размер в битах	Представление
Шестнадцатеричный	8, 16 и 32	B#16#, W#16# и DW#16#
Двоичный	8, 16 и 32	2#
Дата ИЕС	16	D#
Время ИЕС	32	T#
Время дня	32	TOD#
Символьный	8	'A'

Биты, байты, слова и двойные слова

Бит - двоичное число (0 или 1), байт - 8 битов, слово - 16 битов и двойное слово - 32 бита.

Типы данных

Каждый входной и выходной параметр блока FUP может иметь один из следующих типов:

- элементарные типы (см. таблицу С-2)
- составные типы (массив, структура, строка, дата и время)
- таймер, счетчик и блок
- указатель и массив

За более подробной информацией о структурах данных и массивах, которые Вы можете определить сами, и о других структурных типах данных, как, например, STRING (строка) и DATE_AND_TIME (дата и время), обратитесь к *Руководству по программированию /234/* и *Руководству пользователя /231/*.

Таблица С-2. Форматы констант для элементарных типов данных

Тип и описание	Размер в битах	Выбор формата	Диапазон и представление числа (от наименьшего до наибольшего значения)	Пример
BOOL (бит)	1	Булев текст	TRUE/FALSE	TRUE
BYTE (байт)	8	Шестнадцатиричный	от B#16#0 до B#16#FF	B#16#10 byte#16#10
WORD (слово)	16	Двоичный Шестнадцатиричный BCD Десятичный без знака	от 2#0 до 2#1111_1111_1111_1111 от W#16#0 до W#16#FFFF от C#0 до C#999 от B#(0,0) до B#(255,255)	2#0001_0000_0000_0000 W#16#1000 word16#1000 C#998 B#(10,20) byte#(10,20)
DWORD (двойное слово)	32	Двоичный Шестнадцатиричный Десятичный без знака	от 2#0 до 2#1111_1111_1111_1111_1111_1111_1111_1111 от DW#16#0000_0000 до DW#d16#FFFF_FFFF от B#(0,0,0,0) до B#(255,255,255,255)	2#1000_0001_0001_1000_1011_1011_0111_1111_1111_1111_1111_1111 DW#16#00A2_1234 dword#16#00A2_1234 B#(1, 14, 100, 120) byte#(1,14,100,120)
INT (целое число)	16	Десятичный со знаком	от -32768 до 32767	1
DINT (двойное целое число)	32	Десятичный со знаком	от L#-2147483648 до L#2147483647	L#1
REAL (число с плавающей точкой)	32	IEEE Плавающая точка	Верхний предел: ±3 402823e+38 Нижний предел: ±1 175 495e-38	1.234567e+13
S5TIME (время SIMATIC)	16	Время S5 с дискретностью 10 мс (как величина по умолчанию)	от S5T#0H_0M_0S_10MS до S5T#2H_46M_30S_0MS и S5T#0H_0M_0S_0MS	S5T#0H_1M_0S_0MS S5Time#0H_1H_1M_0S_0MS
TIME (время IEC)	32	Время IEC с дискретностью 1 мс, целое со знаком	от T#-24D_20H_31M_23S_648MS до T#24D_20H_31M_23S_647MS	T#0D_1H_1M_0S_0MS TIME#0D_1H_1M_0S_0MS
DATE (дата IEC)	16	Дата IEC с дискретностью 1 день	от D#1990-1-1 до D#2168-12-31	D#1994-3-15 DATE#1994-3-15
TIME_OF_DAY (время дня)	32	Время дня с дискретностью 1 мс	от TOD#0:0:0.0 до TOD #23:59:59.999	TOD#1:10:3.3 TIME_OF_DAY#1:10:3.3
CHAR (символ)	8	Символы ASCII	'A', 'B' и т.д.	'E'

Целые числа (Integer): 16 бит

Числа типа Integer - это целые числа, которые имеют знак, чтобы указать положительны они или отрицательны. В памяти 16-битное целое занимает одно слово. Таблица С-3 показывает диапазон значений 16-битного целого. Рисунок С-1 показывает целое +44 в двоичном формате.

Таблица С-3. Диапазон целых чисел (16 бит)

Формат	Диапазон
16-битное целое	от -32 768 до +32 767



Рис. С-1. 16-битовое целое в двоичном формате: +44

Двойные целые (Double Integers): 32 бита

Числа типа Integer - это целые числа, которые имеют знак, чтобы указать положительны они или отрицательны.. В памяти 32-битное целое число (двойное целое) занимает до двух слов. Таблица С-4 показывает диапазон значений двойного целого числа. Рисунок С-2 показывает целое - 500 000 в двоичном формате. В двоичном формате отрицательная форма целого представляется как дополнение до двух положительной формы этого целого. Вы получаете дополнение до двух целого числа, инвертируя состояния всех битов и затем добавляя + 1 к результату.

Таблица С-4. Диапазон значений двойного целого (32 бита)

Формат	Диапазон
32-битное целое	от -2 147 483 648 до +2 147 483 647

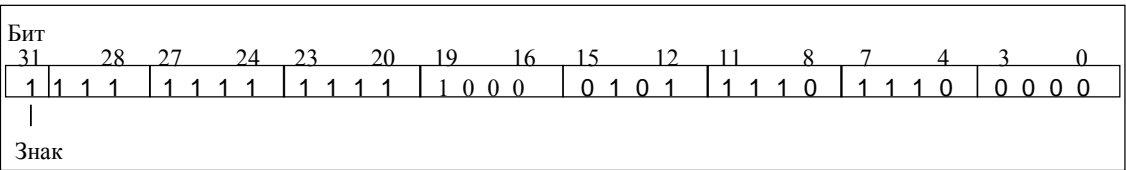


Рис. С-2. Двойное целое в двоичном формате: -500 000

Числа с плавающей точкой

Число с плавающей точкой (называемое также вещественным или действительным числом) - это положительное или отрицательное число, которое содержит десятичную величину, например, 0.339 или – 11.1. Вы можете также включить экспоненту для числа с плавающей точкой, чтобы указать целую степень числа 10, на которую действительное число умножается, чтобы получить величину, которую Вы хотите представить. Например, Вы можете представить 1 234 000 как 1.234E6 или 1.234e6 (т.е. 1.234×10^6). Таблица C-5 показывает диапазон значений числа с плавающей точкой.

В памяти число с плавающей точкой занимает два слова (32 бита, см. рисунок C–3). Самый старший бит указывает знак числа (бит 31, где 0 указывает на плюс, 1 указывает на минус). Другие биты представляют экспоненту и мантиссу.

Таблица C-5. Диапазоны значений чисел с плавающей точкой (вещественных)

Формат	Диапазон ¹⁾
Числа с плавающей точкой	от –3.402 823E+38 до –1.175 495E–38 и ± 0 и от +1.175 495E–38 до +3.402 823E+38

¹ Если результат операции с плавающей точкой попадет в диапазоны от – 1.175495E–38 до –1.401298E–45 или от +1.401298E–45 до +1.175495E–38, то произойдет потеря значимости (см. таблицу 12–2). Это - диапазон ненормализованных чисел.

Формат чисел с плавающей точкой

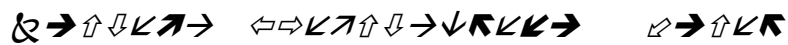
Числа с плавающей точкой в FUP соответствуют основному формату единичной ширины, описанному в стандарте ANSI/IEEE Std 754–1985, *Стандарт IEEE для двоичной арифметики с плавающей точкой*. В этом формате Вы можете представить только величины, которые определяются следующими тремя целыми параметрами:

- p = количество значимых битов (точность)
- E_{\max} = максимальная экспонента
- E_{\min} = минимальная экспонента

Таблица C–6 показывает параметры форматов.

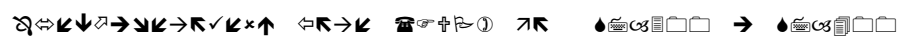
Таблица C-6. Параметры для формата чисел с плавающей точкой

Наименование параметра	Значение параметра
p	24
E_{\max}	+127
E_{\min}	-126
Смещение экспоненты	+127
Ширина экспоненты в битах	8
Ширина формата в битах	32



Формат включает следующие объекты:

- Числа в форме $(-1)^s 2^E (b_0 . b_1 b_2 \dots b_{p-1})$, где
 - $s = 0$ или 1
 - E = любое целое между E_{\min} и E_{\max} включительно
 - $b_i = 0$ или 1
- Две бесконечности, $+\infty$ и $-\infty$
- По крайней мере один сигнализирующий NaN (NaN означает "не число плавающей точкой")
- По крайней мере один не сигнализирующий NaN



Десятичное значение 10,0

Шестнадцатеричное значение 4

Биты 31 28 27 24 23 20 19 16 15 12 11 8 7 4 3 0

0 1 0 0 0 0 0 1 0 0 1 0

Знак
мантиссы:
s (1 бит)

Экспонента: e
(8 бит)
 $e = 2^7 + 2^1 = 130$
 $1.f * 2^{e-смещ} = 1,25 * 2^3 = 10,0$
 $[1,25 * 2^{(130-127)} = 1,25 * 2^3 = 10,0]$

Мантисса: f
(23 бита)
 $f = 2^{-2} = 0,25$

Десятичное значение 3,141593

Шестнадцатеричное значение 4

Биты 31 28 27 24 23 20 19 16 15 12 11 8 7 4 3 0

0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 0 0

Знак
мантиссы:
s (1 бит)

Экспонента: e
(8 бит)

Мантисса: f
(23 бита)

Десятичное значение 1,414214

Шестнадцатеричное значение 3

Биты 31 28 27 24 23 20 19 16 15 12 11 8 7 4 3 0

0 0 1 1 1 1 1 1 1 0 1 1 0 1 0 1 0 0 0 0 0 1 0 0 1 1 1 1 0 1 1 1

Знак
мантиссы:
s (1 бит)

Экспонента: e
(8 бит)

Мантисса: f
(23 бита)

Двоично-десятичные числа

Двоично-десятичный формат (BCD) представляет десятичное число с помощью групп двоичных цифр (битов). Одна группа из 4 битов представляет одну цифру десятичного числа со знаком или знак десятичного числа. Группы из 4 битов объединяются, образуя слово (16 битов) или двойное слово (32 бита). Четыре самых старших бита указывают знак числа (1111 указывают минус и 0000 указывают плюс). Команды с операндами в формате BCD оценивают только самый старший бит (15 в слове, 31 в двойном слове). Таблица С-7 показывает формат и диапазон для двух типов чисел BCD. Рисунки С-5 и С-6 дают пример двоично-десятичного числа в формате слова и двойного слова соответственно.

Таблица С-7. 16- и 32-битные двоично-десятичные числа

Формат	Диапазон
Слово (16 бит, трехразрядное BCD-число со знаком)	от -999 до +999
Двойное слово (32 бита, семи-разрядное BCD-число со знаком)	от -9 999 999 до +9 999 999

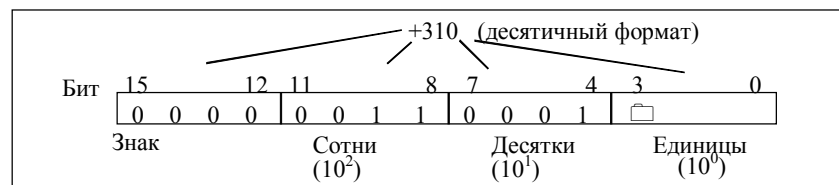


Рис. С-5 Двоично-десятичное число в формате слова

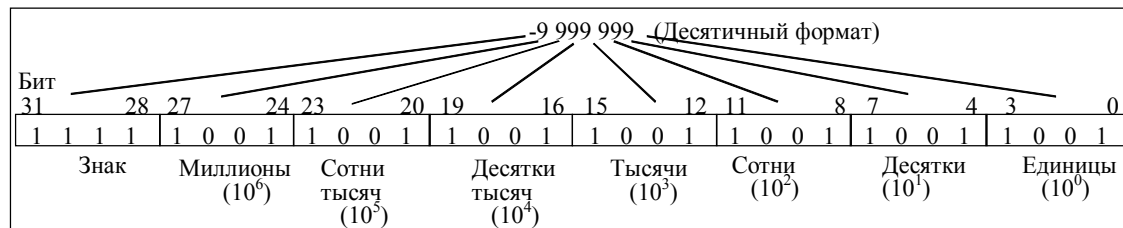


Рис. С-6 Двоично-десятичное число в формате двойного слова

Ввод временного интервала

Когда Вы вводите временной интервал, использующий тип данных S5TIME, Ваши данные хранятся в двоично-десятичном формате (BCD, см. рисунок С-7 и таблицу С-8).

Работая с S5TIME, Вы вводите величину времени в диапазоне от 0 до 999 и указываете базу времени (см. таблицу С-8). База времени показывает интервал, за который таймер уменьшает значение времени на одну единицу, пока оно не достигнет 0.

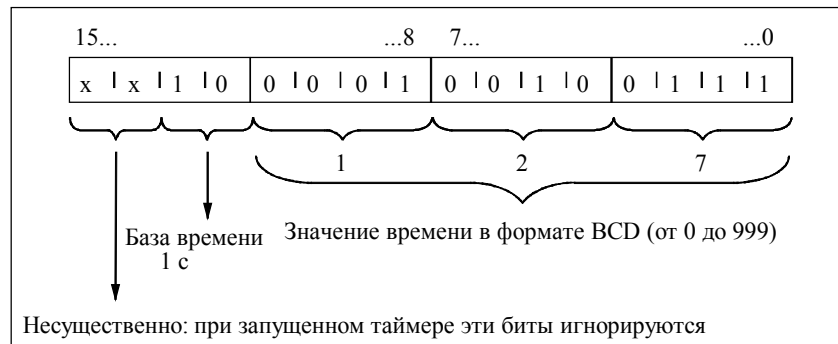


Рис. С-7. Содержимое операнда таймера: значение времени 127, база времени 1с

Таблица С-8. База времени для S5TIME

База времени	Двоичный код для базы времени
10 мс	00
100 мс	01
1 с	10
10 с	11

Вы можете загрузить предопределенное значение времени, используя любой из следующих форматов синтаксиса:

- W#16#wxyz
 - где w = база времени (т.е. интервал времени или разрешение)
 - где xyz = значение времени в двоично-десятичном формате
- S5T#hH_bbM_ccS_dddMS
 - где h = часы, bb = минуты, cc=секунды и dd = миллисекунды
 - база времени выбирается автоматически и величина округляется до ближайшего меньшего числа с этой базой времени.

Максимальная величина времени, которую Вы можете ввести, равна 9 990 секунд, или 2H_46M_30S.

Ввод даты и времени

Когда Вы вводите дату и время, используя тип данных DATE_AND_TIME, Ваши данные хранятся в двоично-десятичном формате (см. С-9). Тип данных DATE_AND_TIME имеет следующий диапазон:

от DT#1990-1-1-0:0:0.0 до DT#2089-12-31-23:59:59.999

Следующие примеры показывают синтаксис для даты и времени для четверга, 25 декабря 1993 года, 8:01 и 1.23 секунды утра. Возможны следующие два формата:

- DATE_AND_TIME#1993-12-25-8:01:1.23
- DT#1993-12-25-8:01:1.23

Следующие специальные стандартные функции IEC (Международная Электротехническая Комиссия) доступны для работы с типом данных DATE_AND_TIME (для получения дополнительной информации см. *Руководство по программированию* /234/:

- Преобразование даты и времени суток в формат DATE_AND_TIME (DT)

FC3: D_TOD_DT

- Извлечение даты из формата DATE_AND_TIME

FC6: DT_DATE

- Извлечение дня недели из формата DATE_AND_TIME

FC7: DT_DAY

- Извлечение времени суток из формата DATE_AND_TIME

FC8: DT_TOD

Таблица C-9 показывает содержимое байтов, которые содержат информацию о дате и времени для четверга, 25 Декабря, 1993, 8:01 и 1.23 секунды утра.

Таблица C-9. Содержимое байтов даты и времени

Байт	Содержимое	Пример
0	Год	B#16#93
1	Месяц	B#16#12
2	День	B#16#25
3	Час	B#16#08
4	Минута	B#16#01
5	Секунда	B#16#01
6	Две наиболее значимых цифры MSEC	B#16#23
7 (4MSB)	Наименее значимая цифра MSEC	B#16#6
7 (4LSB)	День недели 1 = Воскресенье 2 = Понедельник ... 7 = Суббота	B#16#5

Список литературы

D

- /30/ Primer: *S7-300 Programmable Controller, Quick Start*
- /30/ Букварь: Программируемый контроллер *S7-300*, Быстрый старт
- /70/ Manual: *S7-300 Programmable Controller, Hardware and Installation*
- /70/ Руководство: Программируемый контроллер *S7-300*, Аппаратное обеспечение и установка
- /71/ Reference Manual: *S7-300 and M7-300 Programmable Controllers, Module Specifications*
- /71/ Справочное руководство: *Программируемые контроллеры S7-300 и M7-300*, Технические данные модулей
- /72/ Instruction List: *S7-300 Programmable Controller*
- /72/ Список команд: *Программируемые контроллеры S7-300*
- /100/ Manual: *S7-400, M7-400 Programmable Controllers, Hardware and Installation*
- /100/ Руководство: *Программируемые контроллеры S7-400, M7-400*, Аппаратное обеспечение и установка
- /101/ Reference Manual: *S7-400, M7-400 Programmable Controllers, Module Specifications*
- /101/ Справочное руководство: *Программируемые контроллеры S7-400, M7-400*, Технические данные модулей
- /102/ Reference Guide: *S7-400 Instruction List, CPU 412, 413, 414, 416*
- /102/ Справочник: *Список команд S7-400, CPU 412, 413, 414, 416*
- /231/ User Manual: *Standard Software for S7 and M7, STEP 7*
- /231/ Руководство пользователя: базовое программное обеспечение для *S7* и *M7*, STEP 7
- /232/ Manual: *Statement List (STL) for S7-300 and S7-400, Programming*
- /232/ Руководство: *AWL для S7-300/400, Программирование блоков*
- /233/ Manual: *Ladder Logic (LAD) for S7-300 and S7-400, Programming*
- 233/ Руководство: *KOP для S7-300 и S7-400, Программирование блоков*
- /234/ Programming Manual: *System Software for S7-300 and S7-400, Program Design*

- /234/Руководство по программированию: системное программное обеспечение для S7-300/400
Разработка программы
- /235/ Reference Manual: *System Software for S7-300 and S7-400*,
System and Standard Functions
- /235/Справочное руководство: системное программное обеспечение для S7-300/400
Системные и стандартные функции
- /237/ *Master Index*, STEP 7
- /237/Главный индекс, STEP 7
- /250/ Manual: *Structured Control Language (SCL) for S7-300 and S7-400*,
Programming
- /250/Руководство: *Структурный язык управления (SCL) для S7-300 и S7-400*,
Программирование
- /251/Manual: *GRAPH for S7-300 and S7-400*,
Programming Sequential Control Systems
- /251/ Руководство: *GRAPH для S7-300 и S7-400*,
Программирование систем последовательного управления
- /252/Manual: *HiGraph for S7-300 and S7-400*,
Programming State Graphs
- /252/ Руководство: *HiGraph для S7-300 и S7-400*,
Программирование графов состояния
- /253/ Manual: *C Programming for S7-300 and S7-400*,
Writing C Programs
- /253/ Руководство: *Программирование на C для S7-300 и S7-400*,
Написание программ на языке C
- /254/Manual: *CFC Programming Continuous Function Charts*
- /254/ Руководство: *CFC - Программирование непрерывных функциональных карт*
- /270/ Manual: *S7-PDIAG for S7-300 and S7-400*,
Configuring Process Diagnostics for LAD, STL, and FBD
- /270/ Руководство: *S7-PDIAG для S7-300 и S7-400*,
Конфигурирование диагностики процесс для KOP, AWL и FUP
- /271/ Manual: *NETPRO*,
Configuring Networks
- /271/ Руководство: *NETPRO*,
Конфигурирование сетей
- /800/ *DOCPRO*
Creating Documentation (CD only)
- /800/ *DOCPRO*
Создание документации (только CD)

- /801/ TeleService for S7, C7, and M7*
Remote Maintenance for Automation Systems (CD only)
- /801/ Телесервис для S7, C7 и M7*
Дистанционная поддержка систем автоматизации (только CD)
- /802/ PLC Simulation for S7-300 and S7-400 (CD only)*
- /802/Имитация ПЛК для S7-300 и S7-400 (только CD)*
- /803/ Reference Manual: Standard Software for S7-300 and S7-400,*
STEP 7 Standard Functions, Part 2 (CD only)
- /803/ Справочное руководство: Базовое программное обеспечение для S7-300 и S7-400, Стандартные функции STEP 7, ч. 2 (только CD)*

Глоссарий

А

Абсолютная адресация

При абсолютной адресации указывается расположение в памяти адреса, подлежащего обработке. Пример: адрес Q4.0 описывает бит 0 в байте 4 области отображения процесса на входах.

Абсолютный операнд

Абсолютный операнд включает в себя идентификатор операнда и физическое место в памяти, где этот операнд находится. Примеры: I12.1, меркерное слово MW25; блок данных DB3.

Аккумулятор

Аккумуляторы - это регистры в CPU, которые действуют как промежуточные буферы для операций загрузки, передачи, сравнения, арифметических операций и операций преобразования.

Б

Бит переполнения

Бит состояния OS указывает на переполнение. Переполнение может возникнуть, например, после арифметической операции.

Бит переполнения с запоминанием

Бит состояния OS означает “бит переполнения с запоминанием”. Переполнение может иметь место, например, после арифметической операции.

Бит результата (BR)

Бит результата - это связующее звено между обработкой битов и слов. Это эффективный метод, позволяющий осуществить двоичную интерпретацию результата выполнения команды над словом и включить его в последовательность логических операций.

Бит состояния

Бит состояния хранит значение бита, к которому производится обращение. Состояние битовой команды, которая имеет доступ для чтения к памяти (A, AN, O, ON, X, XN), всегда такое же, как и значение бита, который эта команда контролирует (бита, с которым она выполняет свою логическую операцию). Состояние битовой команды, которая имеет доступ к памяти на запись (S, R, =), такое же, как и значение бита, в который команда записывает, или, если запись не производится, такое же, как и значение бита, к которому команда обращается. Бит состояния не имеет значения для битовых команд, которые не имеют доступа к памяти. Такие команды устанавливают бит состояния в 1 (STA=1). Бит состояния не контролируется командой. Он только интерпретируется во время тестирования программы (статус программы).

Бит OR

Бит OR необходим при выполнении операции логическое И перед ИЛИ. Бит OR показывает этим командам, что ранее выполненная операция И дала значение 1, так что исход операции ИЛИ уже предрешен. Любая другая команда, обрабатывающая биты, сбрасывает бит OR.

Блок данных (DB)

Блоки данных - это области в программе пользователя, которые хранят данные пользователя. Имеются глобальные (разделяемые) блоки данных, к которым имеют доступ все логические блоки, и экземпляры блоков данных, ассоциированных с вызовами определенного функционального блока (FB). В отличие от других блоков блоки данных не содержат команд.

Блоки (программные объекты)

Контейнер для блоков, загружаемых в программируемый модуль S7 (напр., CPU, FM), где они могут исполняться для управления установкой или процессом.

В

Ввод, инкрементный

Когда блок вводится инкрементно, то каждая строка или элемент немедленно проверяется на наличие ошибок (например, синтаксических ошибок). Если обнаружена ошибка, она отмечается и должна быть исправлена, прежде чем будет выполняться компиляция. Инкрементный ввод возможен в AWL (список команд), KOP (контактный план) и FUP (функциональный план).

Г

Глобальный блок данных (DB)

Глобальный (разделяемый) блок данных - это DB, адрес которого загружается в адресный регистр DB при его открытии. Он предоставляет память и данные для всех исполняемых в данный момент логических блоков (FC, FB или OB).

Напротив, экземпляр DB спроектирован для использования в качестве специфической памяти и данных для FB, с которым он связан.

Д

Данные, статические

Статические данные - это локальные данные функционального блока, которые хранятся в экземпляре блока данных и вследствие этого сохраняются, пока функциональный блок не будет обрабатываться снова.

И

Идентификатор операнда

Идентификатор операнда - это часть операнда, содержащая различные данные. Эти данные могут включать такие элементы, как объект данных или размер объекта данных, с которым команда может, например, выполнить логическую операцию. В команде "L IB10" IB - это идентификатор операнда ("I" указывает на область входов в памяти, а "B" указывает байт в этой области).

Иерархия вызовов

Все блоки перед обработкой должны быть вызваны. Последовательность и вложенность этих вызовов в организационном блоке называется иерархией вызовов.

Исходный файл

Исходный файл (текстовый файл) - это часть программы, создаваемая с помощью графического или текстового редактора, которая компилируется в исполняемую пользовательскую программу S7 или в машинный код M7.

Исходный файл S7 хранится в папке "Sources" ("Исходные тексты") в программе S7.

К

Карта

Специальный графический исходный файл, создаваемый с использованием языка программирования CFC (Continuous Function Chart - Непрерывная функциональная карта).

Ключевое слово

Ключевые слова используются при программировании в исходных файлах для идентификации начала и конца блока и для выбора подразделов в разделе описаний блока, начала комментариев к блоку и начала заголовка.

Коды условий CC 1 и CC 0

Биты CC 1 и CC 0 (коды условий) предоставляют информацию о следующих результатах или битах:

- результат арифметической операции
- результат сравнения
- результат цифровой операции
- биты, выдвинутые командой сдвига или циклического сдвига

Команда

Команда - это часть оператора; она определяет, что должен делать процессор.

Контактный план (KOP)

Контактный план - это графический язык программирования в STEP 5 и STEP 7. Его представление стандартизировано в соответствии с DIN (международный стандарт ИЕС 1131-1). Представление в виде контактного плана соответствует представлению в виде релейных контактных схем. В отличие от списка команд (AWL) KOP имеет более ограниченный набор команд.

Л

Логическая цепь

Логическая цепь - это часть программы пользователя, которая начинается битом FC, находящимся в состоянии 0, и заканчивается, когда команда или событие сбрасывает бит FC в 0. Когда CPU исполняет первую команду в логической цепи, бит FC устанавливается в 1. Некоторые команды, такие, как команды вывода (например, установка, сброс, присваивание), сбрасывают бит FC в 0. См. ниже *Первичный опрос*.

Логический блок

Логические блоки - это блоки внутри STEP 7, содержащие программу для управляющей логики. В отличие от них, блоки данных (DB) содержат только данные. Имеются следующие типы логических блоков:

- организационный блок (OB)
- функциональный блок (FB)
- функция (FC)
- системный функциональный блок (SFB)
- системная функция (SFC)

М

Массив

Массив - это составной тип данных, который состоит из элементов данных одного и того же типа. Эти элементы могут быть элементарными или составными.

Мнемоническое представление

Мнемоническое представление - это сокращенная форма отображения наименований операндов и команд в программе (например, "I" используется вместо "input" ("вход")). STEP 7 поддерживает международное представление (основанное на английском языке) и представление SIMATIC (основанное на немецких сокращениях набора команд и соглашениях об адресации, принятых в SIMATIC).

Н

Непосредственная адресация

При непосредственной адресации операнд содержит значение, с которым работает команда.

Пример: L 27 означает загрузку в аккумулятор числа 27.

О

Область памяти

CPU в SIMATIC Manager имеет три области памяти:

- загрузочную память
- рабочую память
- системную память

Операнд

Операнд - это часть команды STEP 7, который указывает, над чем процессор должен выполнить команду. Операнды бывают абсолютные и символические.

Оператор

Оператор - это наименьшая часть программы пользователя, создаваемой на текстовом языке. Оператор представляет собой команду для процессора.

Описание

Раздел описаний используется для описания локальных данных логического блока при программировании в текстовом редакторе.

Описание переменной

Описание переменной включает в себя символическое имя, тип данных и, факультативно, начальное значение, адрес и комментарий.

П**Первичный опрос**

Первый опрос результата логической операции.

Программа пользователя

Программа пользователя содержит все операторы и описания и все данные для обработки сигналов, которые могут быть использованы для управления установкой или процессом. Она является частью программируемого модуля (CPU, FM) и может быть структурирована разбиением на более мелкие единицы (блоки).

Программа S7

Это контейнер для блоков, исходных файлов и карт для программируемых контроллеров S7. Программа S7 включает также таблицу символов.

Проект

Проект - это контейнер для всех объектов в задаче автоматизации независимо от количества станций, модулей и способа их соединения в сети.

Прямая адресация

При прямой адресации адрес содержит информацию о местоположении в памяти значения, которое должно быть использовано командой.

Пример:

Адрес Q4.0 определяет бит 0 в байте 4 таблиц отображения процесса на выходах.

Р**Результат логической операции (RLO)**

Результат логической операции (RLO) - это текущее состояние сигнала в процессоре, которое используется для обработки других двоичных сигналов. Исполнение определенных команд полностью зависит от предшествующего им значения RLO.

С

Сегмент

Сегменты подразделяют блоки FUP на разделы.

Символ

Символ - это имя, которое может быть определено пользователем при условии соблюдения синтаксических правил. После того, как он описан (например, как переменная, тип данных, метка перехода, блок и т.д.), символ может использоваться для программирования и для функций связи с оператором. Пример: Операнд I 5.0, тип данных: Bool, Символ: momentary contact switch / emergency stop (выключатель мгновенного действия/ аварийный останов).

Символическая адресация

При символической адресации обрабатываемый адрес обозначается символом (в отличие от абсолютного адреса). Назначение символа адресу производится в таблице символов.

Системная функция (SFC)

Системная функция встроена в CPU и может быть вызвана, если необходимо, из программы пользователя STEP 7.

Системный функциональный блок (SFB)

Системный функциональный блок - это функциональный блок, который встроен в операционную систему CPU, который Вы можете при необходимости вызвать из своей пользовательской программы.

Слово состояния

Слово состояния - это часть регистра CPU. Оно содержит информацию о состоянии и информацию об ошибках, которая отображается при исполнении определенных команд STEP 7. Биты состояния могут как читаться, так и записываться пользователем, биты ошибок могут быть только считаны.

Список команд (AWL)

Список команд - это текстовое представление языка программирования STEP 7, подобного машинному коду. AWL - это язык ассемблера STEP 5 и STEP 7. Если Вы программируете на AWL, то отдельные команды представляют собой фактические шаги, которыми CPU исполняет программу.

Справочные данные

Справочные данные используются для проверки программы пользователя и содержат список перекрестных ссылок, список назначений, структуру программы, список неиспользованных адресов и список адресов, на имеющих символических обозначений.

Станция

Станция - это устройство, которое может быть соединено с одной или более подсетей, например, программируемый контроллер, устройство программирования, операторная станция.

Структура программы пользователя

Структура программы пользователя описывает иерархию вызовов блоков внутри программы пользователя и дает обзор используемых блоков и их зависимости.

Т

Таблица описания переменных

Таблица описания переменных используется для описания локальных данных логического блока, когда программирование выполняется в инкрементном редакторе.

Таблица переменных (VAT)

Таблица переменных используется для того, чтобы собрать вместе переменные, которые Вы хотите наблюдать и которыми хотите управлять, и установить для них подходящие форматы.

Таблица символов

Таблица, в которой размещены символы адресов для глобальных данных и блоков. Примеры: Emergency Stop (символ) – I 1.7 (адрес) или closed-loop control (символ) – SFB24 (блок).

Тип данных

Тип данных определяет, как значение переменной или константы должно использоваться в программе пользователя.

В SIMATIC STEP 7 в распоряжении пользователя имеется две группы типов данных (IEC 1131-3):

- Элементарные типы данных
- Составные типы данных

Тип данных, составной

Составные типы данных создаются пользователем с помощью описания типа данных. Они не имеют собственного имени и поэтому не могут быть использованы снова. Типы данных STRING (строка) и DATE AND TIME (дата и время) относятся к составным типам данных.

Тип данных, элементарный

Элементарные типы данных являются заранее установленными типами данных в соответствии с IEC 1131-3.

Примеры:

- "BOOL" определяет двоичную переменную ("бит")
- тип данных "INT" определяет 16-битовую переменную с фиксированной точкой.

Типы данных, определенные пользователем (UDT)

Типы данных, определенные пользователем, - это специальные структуры, которые Вы можете создавать сами и применять после их определения во всей пользовательской программе. Их можно использовать подобно элементарным или составным типам данных в разделе описания переменных логических блоков (FC, FB, OB) или как шаблон для создания блоков данных с такой же структурой данных.

У

Указатель

Вы можете использовать указатель для идентификации адреса переменной. Указатель содержит идентификатор вместо значения. Если Вы назначаете тип фактического параметра, то Вы предоставляете адрес в памяти. В STEP 7 Вы можете ввести указатель или в формате указателя, или просто как идентификатор (напр., M 50.0). В следующем примере показан формат указателя, с помощью которого можно получить доступ к данным из M 50.0:

R#M50.0

Ф

Фактический параметр

Фактические параметры заменяют формальные параметры при вызове функциональных блоков (FB) и функций (FC).

Пример: Формальный параметр “Start” заменяется фактическим параметром “I3.6”.

Формальный параметр

Формальный параметр - это держатель места для фактического параметра в логических блоках. В функциональных блоках (FB) и функциях (FC) формальные параметры описываются пользователем, в системных функциональных блоках (SFB) и системных функциях (SFC) они уже имеются в распоряжении. При вызове блока формальные параметры ставятся в соответствие фактическим параметрам; блок работает с фактическими параметрами.

Формальные параметры относятся к локальным данным. Они могут быть входными, выходными и проходными (вход/выход).

Функциональный блок

В соответствии со Стандартом Международной электротехнической комиссии IEC 1131-3 функциональные блоки - это логические блоки, связанные с экземпляром блока данных, т.е. они имеют статические данные. Функциональные блоки позволяют передавать параметры в программу пользователя, поэтому они пригодны для программирования часто повторяющихся сложных функций, например, управления по замкнутому контуру или выбора режима работы.

Функциональный план (FUP)

Функциональный план - это один из языков программирования в STEP 7. FUP представляет собой логику в блоках, известных из булевой алгебры.

Функция (FC)

В соответствии со Стандартом Международной электротехнической комиссии IEC 1131-3 функции - это логические блоки, которые не связаны с экземпляром блока данных, т.е. они не имеют “памяти”. Функции позволяют передавать параметры в программу пользователя, поэтому они пригодны для программирования часто повторяющихся сложных функций, например, расчетов.

Э

Экземпляр

“Экземпляр” - это вызов функционального блока. Если, например, функциональный блок вызывается в STEP 7 пять раз, то имеется пять экземпляров. Каждый вызов ставится в соответствие экземпляру блока данных.

Экземпляр блока данных (DB)

Экземпляр блока данных хранит формальные параметры и статические локальные данные функционального блока. Экземпляр блока данных может быть поставлен в соответствие одному или нескольким функциональным блокам.

М

Master Control Relay

Master Control Relay (Главное управляющее реле) - это переключатель, пропускающий или запирающий поток сигнала (путь тока). Деактивированный путь тока соответствует последовательности команд, которая записывает нулевое значение вместо расчетного значения, или последовательности команд, которая оставляет неизменным существующее состояние памяти.