1. БЛОК «АЛГОРИТМ»

Обов'язкові байти: 4 байти операндів умови **IF_A** і **IF_B**, 1 байт операції **IF_Op**, 1 байт **Then/Else** вказує на дії, які виконуються у даному кроці. 4 байти **GOTO_Then** і **GOTO_Else** для переходу на наступний крок. Байт **Num_STEP** № крока 0-255, Разом 11 байт.

Структура байта Then / Else (9)	масив ALGORYTHM
Біт 0 - Включення \ Виключення виходів Output 1 - 32	8 байт (11-18)
Біт 1 - Включення \ Виключення прапорців Flags 1 – 32	8 байт (19-26)
Біт 2 – Управління виходами/задатчиками OUT1-32/SP1-128/FB1-96	4 байт (27-30)
Біт 3 – Управління виходами/задатчиками OUT1-32/SP1-128/FB1-96	4 байт (31-34)
Біт 4 - Включення \ Виключення виходів Output 1 - 32	8 байт (35-42)
Біт 5 - Включення \ Виключення прапорців Flags 1 – 32	8 байт (43-50)
Біт 6 – Управління виходами/задатчиками OUT1-32/SP1-128/FB1-96	4 байт (51-54)
Біт 7 – Управління виходами/задатчиками OUT1-32/SP1-128/FB1-96	4 байт (55-58)

Якщо біт встановлений в лог. 1 – дія виконується. Якщо біт дорівнює лог. 0 – Дія не виконується. У випадку коли байт **Then /Else** дорівнює 0, програма переходить на наступний крок, вказаний у байті переходу **GOTO_Then** або **GOTO_Else**, без виконання будь яких дій.

Разом у кожному кроці є 11 обов'язкових байт. Разом у 256 кроках **2816** байт. Всього для блока АЛГОРИТМ виділено **9184** байт пам'яті. Вільними залишається **6384** байти. При середньому використанню 24 байти на крок, програма може налічувати до **256** кроків.

Пакет КРОКА масив ALGORYTHM

2 Byte **IF_A**		2 Byte **IF_B**		Byte **IF_OP**	2 Byte **Goto_Then**		2 Byte **Goto_Else**		Byte **Then/Else**	Byte **#Block**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**

Таблиця 1 Операнд умови **IF_A:**

№ func	Ім'я	Адреса	****** Коментар ******
1	INP1-32	8	If INP1 = *SP1
2	OUT1-32	16	If OUT1 = *SP1
3	FLAG1-32	24	If FLAG1 = *SP1
4	TIMER1-32	32	If TIMER1 = *SP1
5	SP1-128	40	If SP1 = *SP1
6	INI(SP1-128)	48	If INI(SP10) = *SP1, якщо SP10 = 5 тоді If INP6 = *SP1
7	OUI(SP1-128)	56	If OUI(SP10) = *SP1, якщо SP10 = 5 тоді If OUT6 = *SP1
8	SI(SP1-128)	64	If SI(SP10) = *SP1, якщо SP10 = 5 тоді If SP6 = *SP1
9	INI(INP1-32)	72	If INI(INP1) = *SP1, якщо INP1 = 5 тоді If INP6 = *SP1
10	OUI(INP1-32)	80	If OUI(INP1) = *SP1, якщо INP1 = 5 тоді If OUT6 = *SP1
11	SI(INP1-32)	88	If SI(INP1) = *SP1, якщо INP1 = 5 тоді If SP6 = *SP1
12	FB1-96_OUT1-16	96	If FB10_OUT1 = *SP1
13	FB1-96_SP1-16	104	If FB10_SP1 = *SP1
14	FB1-96_OUI(SP1-16)	112	If FB10_OUI(SP10) = *SP1, якщо SP10=5 тоді If FB10_OUT6=*SP1
15	FB1-96_SI(SP1-16)	120	If FB10_SI(SP10) = *SP1, якщо SP10 = 5 тоді If FB10_SP6 = *SP1
16	RDY_PROG	128	0 – FRAM зайнята, 1 – FRAM вільна
17	CALL	136	Перехід у підпрограму
18	RET	144	Повернення з підпрограми
19	LOCK_SP1-128	152	If LOCK_SP10 = 1(0). Перевірка блокування задатчика SP10
20	LOCK_SI(SP1-128)	160	If LOCK_SI(SP10) = 1(0). Перевірка блокування задатчика SI(SP10)
21	D_INP1-32	168	If D_INP5 AND 0, якщо INP5 фронт (0 – фронт _/, 1 – зріз -\)
22	D_INI(SP1-128)	176	If D_INI(SP5) AND 0, якщо INI(SP5) фронт (0 – фронт _/, 1 – зріз -\)

2 байта - № функції старші 5 (0-31) біт, № операнда 11 біт (0-2047)

*SP1 – може бути будь-яким виразом **IF_B**

1 байт операції **IF_Op:** **=, <, >, <=, >=, AND, OR, XOR, CALL, RET**

Таблиця 2 Операнд умови **IF_B:**

№ func	Ім'я	Адреса	****** Коментар ******
0	No	0	
1	Const 0+-1024	8	*SP1 = Const (-1023 // 1023) SP1 = 255
2	INP1-32	16	*SP1 = INP10
3	OUT1-32	24	*SP1 = OUT10
4	SP1-128	32	*SP1 = SP10
5	SI(INP1-32)	40	*SP1 = SI(INP1), якщо INP1 = 5 тоді *SP1 = SP6
6	SI(SP1-128)	48	*SP1 = SI(SP10), якщо SP10 = 5 тоді *SP1 = SP6
7	INI(INP1-32)	56	*SP1 = INI(INP1), якщо INP1 = 5 тоді *SP1 = INP6
8	INI(SP1-128)	64	*SP1 = INI(SP10), якщо SP10 = 5 тоді *SP1 = INP6
9	INP1-32+SP1-64	72	*SP1 = INP1+SP10
10	INP1-32-SP1-64	80	*SP1 = INP1-SP10
11	SP1-32+SP1-64	88	*SP1 = SP5+SP10
12	SP1-32-SP1-64	96	*SP1 = SP5-SP10
13	SI(SP1-32)+SP1-64	104	*SP1 = SI(SP5)+SP10, якщо SP5 = 5 тоді *SP1 = SP6+SP10
14	SI(SP1-32)-SP1-64	112	*SP1 = SI(SP5)-SP10, якщо SP5 = 5 тоді *SP1 = SP6-SP10
15	FB1-96_OUT1-16	120	*SP1 = FB10_OUT5
16	FB1-96_SP1-16	128	*SP1 = FB10_SP5
17	FB1-96_OUI(SP1-16)	136	*SP1 = FB10_OUI(SP5), якщо SP5 = 5 тоді *SP1 = FB10_OUT6
18	FB1-96_SI(SP1-16)	144	*SP1 = FB10_SI(SP5), якщо SP5 = 5 тоді *SP1 = FB10_SP6
19	SP1-128_SP1-16	152	Запис FRAM: № початковий SP, кільк. SP у SP1-16
20	SP1-32*SP1-64	160	*SP1 = SP10*SP5
21	SP1-32/SP1-64	168	*SP1 = SP10/SP5
22	SI(SP1-32)*SP1-64	176	*SP1 = SI(SP10)*SP5, якщо SP10 = 5, тоді *SP1 = SP6*SP5
23	SI(SP1-32)/SP1-64	184	*SP1 = SI(SP10)/SP5, якщо SP10 = 5, тоді *SP1 = SP6/SP5
24	TIMER1-32_S-M-H	192	*SP1 = TIMER10_M, SP1 = TIMER10 у хвилинах (сек, хв, год)
25	ABS_SP1-128	200	*SP1 = ABS_SP50 Абсолютне значення SP
26	SQRT_SP1-128	208	*SP1 = SQRT_SP50 Квадратний корінь з SP
27	LOCK_0-2	216	*SP1 = LOCK_1 Блокування SP1(0-розблок,1- блок,2-розблок 1-128
28	MIN_SP1_128_const	224	SP5 = MIN_SP80_10 Сорт. масив SP80-89, результ SP5min, SP6num
29	MAX_SP1_128_const	232	SP5 = MAX_SP80_10 Сорт. масив SP80-89, результ SP5max, SP6num
30	SUM_SP1_128_const	240	SP5 = SUM_SP80_10 Сума задатчиків SP80-89
31	BYTE_SP0_3_const	248	SP5 |= BYTE_1_8 Встановити біт 3 у байті 1 SP5

2 байта - № функції старші 5 (0-31) біт, № операнда 11 біт (0-2047)

*SP1 – може бути будь-яким виразом дії Then /Else: **OUT/ SP/ FB/ TIMER/ FRAM,** або **IF_A**

Таблиця 3 Дія Then /Else **OUT/SP/FB/TIMER/FRAM**

№	Ім'я	****** Коментар ******
0	No	Пусто
1	OUT1-32 = **f(OUT/SP)**	OUT1 = *SP1
2	OUT1-32 += **f(OUT/SP)**	OUT1 = OUT1 + *SP1
3	OUT1-32 -= **f(OUT/SP)**	OUT1 = OUT1 - *SP1
4	OUT1-32 *= **f(OUT/SP)**	OUT1 = OUT1 * *SP1
5	OUT1-32 /= **f(OUT/SP)**	OUT1 = OUT1 / *SP1
6	OUI(INP1-32) = **f(OUT/SP)**	OUI(INP1) = *SP1, якщо INP1 = 5 тоді OUT6 = *SP1
7	OUI(SP1-128) = **f(OUT/SP)**	OUI(SP10) = *SP1, якщо SP10 = 5 тоді OUT6 = *SP1
8	SP1-128 = **f(OUT/SP)**	SP10 = *SP1
9	SP1-128 += **f(OUT/SP)**	SP10 = SP10 + *SP1
10	SP1-128 -= **f(OUT/SP)**	SP10 = SP10 - *SP1
11	SP1-128 *= **f(OUT/SP)**	SP10 = SP10 * *SP1
12	SP1-128 /= **f(OUT/SP)**	SP10 = SP10 / *SP1

13	SI(INP1-32) = **f(OUT/SP)**	SI(INP1) = *SP1, якщо INP1 = 5 тоді SP6 = *SP1
14	SI(SP1-128) = **f(OUT/SP)**	SI(SP10) = *SP1, якщо SP10 = 5 тоді SP6 = *SP1
15	SI(SP1-128) += **f(OUT/SP)**	SI(SP10) = SI(SP10) + *SP1, якщо SP10 = 5 тоді SP6 = SP6+*SP1
16	SI(SP1-128) -= **f(OUT/SP)**	SI(SP10) = SI(SP10) - *SP1, якщо SP10 = 5 тоді SP6 = SP6-*SP1
17	SI(SP1-128) *= **f(OUT/SP)**	SI(SP10) = SI(SP10) * *SP1, якщо SP10 = 5 тоді SP6 = SP6**SP1
18	SI(SP1-128) /= **f(OUT/SP)**	SI(SP10) = SI(SP10) / *SP1, якщо SP10 = 5 тоді SP6 = SP6/*SP1
19	FB1-96_SP1-16 = **f(OUT/SP)**	FB1_SP10 = *SP1
20	FB1-96_SI(SP1-16) = **f(OUT/SP)**	FB1_SI(SP10) = *SP1, якщо SP10 = 5 тоді FB1_SP6 = *SP1
21	TIMER1-32 = **f(OUT/SP)**	Встановлення таймера
22	WR_SP1-128 = **f(19)**	Запис FRAM: у SP початкова адреса у FRAM 0-8191
23	RD_SP1-128_SP1-16 = **f(4)**	Зчитування FRAM: № початковий SP, кільк. SP у SP1-16
24	CLR_SP1-128_SP1-16 = 0	Обнулення SP: № початковий SP, кільк. SP у SP1-16
25	ROUND_SP1-128 = const (0, 1, 3)	Округлення до const: 0 – цілих, 1 – десятих, 2 - сотих
26	LIM_MIN_SP1-128 = **f(OUT/SP)**	Обмеження SP до мінімуму. LIM_MIN_SP1 = -10
27	LIM_MAX_SP1-128 = **f(OUT/SP)**	Обмеження SP до максимуму. LIM_MAX_SP1 = 10

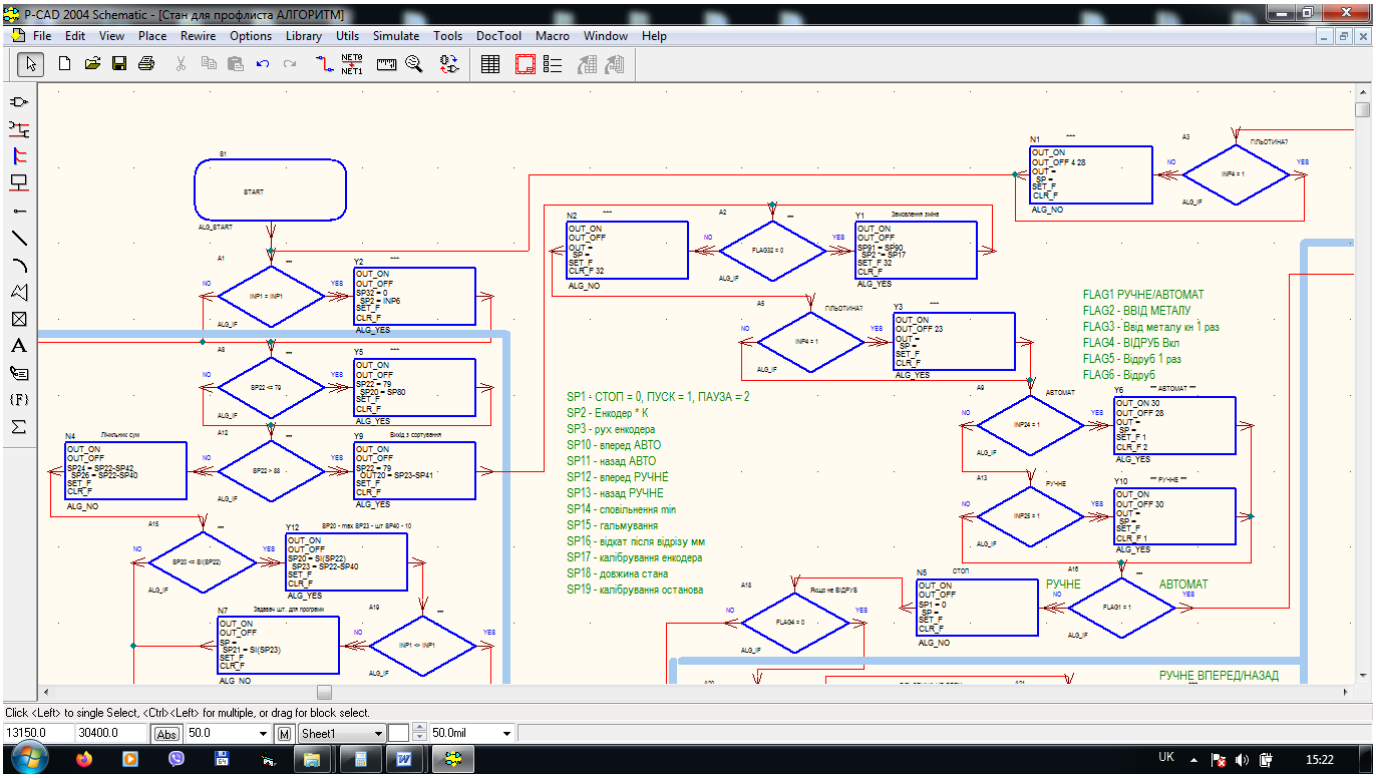
***f(OUT/SP) = IF_B** (Таблиця 2)

*SP1 – може бути будь-яким виразом **IF_B**

2 байта - № функції старші 5 (0-31) біт, № операнда 11 біт (0-2047)

2. ОПИС РОБОТИ БЛОКА “АЛГОРИТМ”

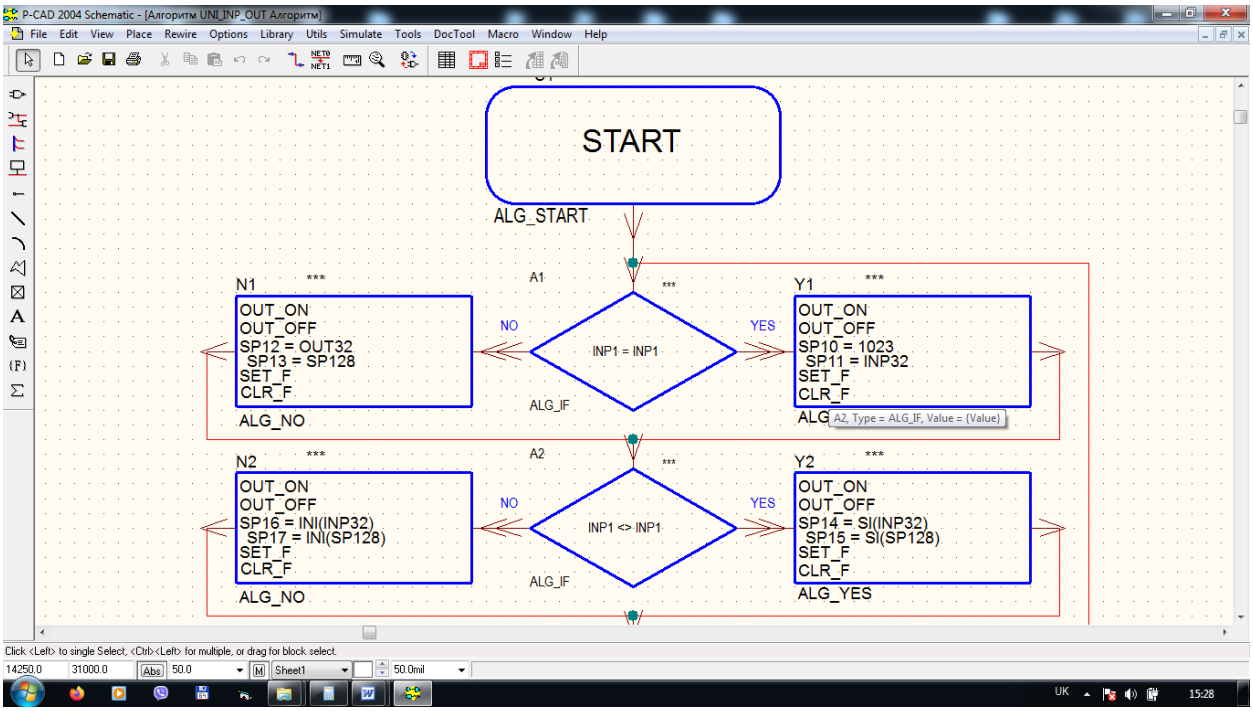
Блок “АЛГОРИТМ” - це вільно програмований блок, який має 32 входи (INP1-INP32) і 32 виходи (OUT1-OUT32). Входи і виходи блоку можуть бути як логічними (діапазон значень: 0 і 1), так і числовими (діапазон значень: дійсні числа). Блок обробляє покрокову блок-схему, яка може налічувати до 255 вільно програмованих кроків. Програмування блок-схеми відбувається в середовищі програмного пакету PCAD (додаток Schematic, бібліотека ALGORITHM_LIB_05.lib) і наступним завантаженням файлу алгоритму з розширенням *.net, в програмний пакет «Конфігуратор К1». Для генерування файлу *.net потрібно зайти в меню Utils – Generate Netlist. Вибрати файл схеми, встановити формат файлу P-CAD ASCII і натиснути Ок.



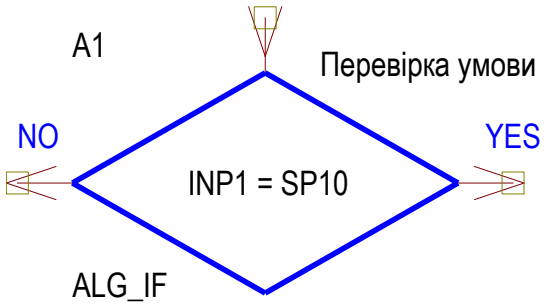
3. ПРИНЦИПИ ПОБУДОВИ БЛОК - СХЕМИ

Кожний крок алгоритму функціонує як оператор If - Then - GoTo
- Else - GoTo

Програма алгоритму завжди починається з оператора START. В роботі програми він не бере участі, але позначає початок циклу програми. Далі виконується оператор ALG_IF з порядковим номером A1 і, в залежності від результату порівняння чи логічної дії, виконуються команди блоків (якщо вони встановлені) ALG_N - N1 або ALG_Y - Y1. Потім програма переходить до наступного оператора ALG_IF. Напрямок руху виконання програми вказують стрілки на схемі блоків. Далі нумерація блоків у схемі не принципова і може відбуватися в автоматичному режимі. Важливо, щоб кількість блоків ALG_IF не перевищувала 255 (A1 – A255). Блоки типу ALG_N і ALG_Y в кожному кроці встановлювати не обов'язково. До одного входу блоку ALG_IF може бути підключена необмежена кількість виходів. На вхід блоків ALG_N і ALG_Y можна підключити тільки один вихід відповідного блоку ALG_IF. Входи блоків з'єднувати між собою заборонено.



4. БЛОК ALG_IF



Блок **ALG_IF** – це оператор управління виконання програми алгоритму в залежності від умови рівності, нерівності або результату логічної операції над операндами. Залежно від результату порівняння або логічної операції відбувається виконання певних дій і перехід до наступного кроку програми. Графічно напрямок переходу вказують ключові слова YES, NO і відповідні стрілки на схемі блоку.

Можливі типи операцій: **= , <=, >=, <>, AND, OR, XOR.**

Якщо результат порівняння є істинним, або результат логічної операції дорівнює лог. 1, то відбувається перехід за стрілкою вправо -> YES. В іншому разі вліво <- NO.
Можливі типи операндів у лівій частині умови: **таблиця 1**.

Операції з даними входів:

INPn – значення входу блока «АЛГОРИТМ» де n – ціле число 1 – 32. Діапазон значень входу INP - дійсні числа (float). Приклад: INP10.

INI(INPn) – значення входу блока "АЛГОРИТМ", на який вказує значення входу блока INPn, де n – ціле число 1 – 32. Приклад: INI(INP10).

INI(SPn) – значення входу блока "АЛГОРИТМ", на який вказує значення задатчика блока SPn, де n – ціле число 1 – 128. Приклад: INI(SP10).

Операції з даними виходів:

OUTn – значення виходу блока «АЛГОРИТМ» де n – ціле число 1 – 32. Діапазон значень виходу OUT - дійсні числа (float). Приклад: OUT10.

OUI(INPn) – значення виходу блока "АЛГОРИТМ", на який вказує значення входу блока INPn, де n – ціле число 1 – 32. Приклад: OUI(INP10).

OUI(SPn) – значення виходу блока "АЛГОРИТМ", на який вказує значення задатчика блока SPn, де n – ціле число 1 – 128. Приклад: OUI(SP10).

Операції з задатчиками:

SPn – значення задатчик блока "АЛГОРИТМ", де n – ціле число 1 – 128. Діапазон значень задатчика SPn - дійсні числа (float). Приклад: SP10.

SI(INPn) – значення задатчика блока "АЛГОРИТМ", на який вказує значення входу блока INPn, де n – ціле число 1 – 32. Приклад: SI(INP10).

SI(SPn) – значення задатчика блока "АЛГОРИТМ", на який вказує значення задатчика блока SPn, де n – ціле число 1 – 128. Приклад: SI(SP10).

Операції з прапорцями:

FLAGn – стан прапорця, де n – ціле число 1 – 32 . Прапорець може приймати значення тільки лог. 0 аба лог. 1. Приклад: FLAG10.

Операції з таймерами:

TIMERn - значення таймера блока "АЛГОРИТМ", де n – ціле число 1 – 32. Приклад: TIMER10. Таймер перевіряється і інкрементується кожного циклу. Якщо значення таймера менше завдання, програма алгоритму переходить у напрямку "NO" блоку **ALG_IF**. Після досягнення таймером завдання програма алгоритму переходить у напрямку "YES" блоку **ALG_IF** і відлік часу припиняється. Для того, щоб обнулити таймер, достатньо не перевіряти його значення один чи більше циклів або обнулити за допомогою команди TIMERn = 0. Запис завдання таймера у задатчик SPn відбувається безпосередньо у кроці перевірки.

Операції з даними функціональних блоків:

FBn_OUTm – значення виходу OUTm функціонального блока FBn, де n – ціле число 1 – 96, m – ціле число 1 – 16. Приклад: FB50_OUT5.
задатчика блока SPm, де n – ціле число 1 – 96, m – ціле число 1 – 128. Приклад: FB50_OUI(SP5).

FBn_SPm – значення задатчика SPm функціонального блока FBn, де n – ціле число 1 – 96, m – ціле число 1 – 16. Приклад: FB50_SP5.

FBn_OUI(SPm) – значення виходу OUT функціонального блока FBn, на який вказує значення

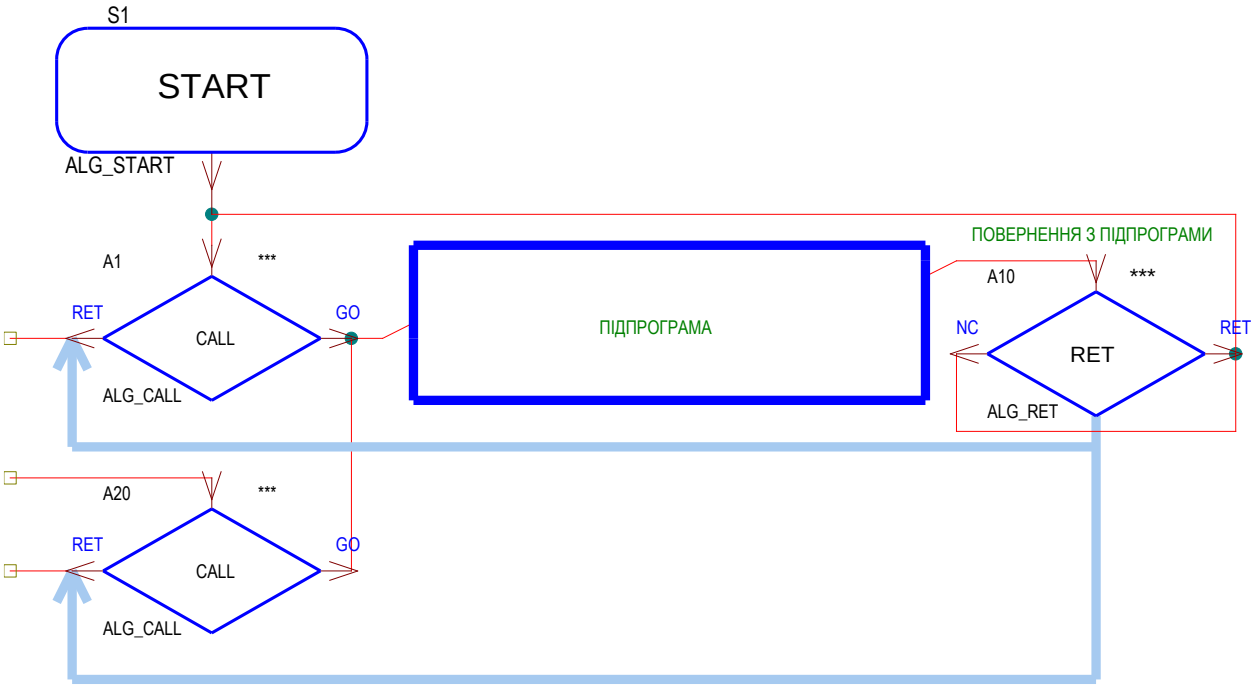
FBn_SI(SPm) – значення задатчика SP функціонального блока FBn, на який вказує значення задатчика блока SPm, де n – ціле число 1 – 96, m – ціле число 1 – 128. Приклад: FB50_SI(SP5).

Інші операції:

RDY_PROG – перевірка стану готовності пам'яті FRAM, при зчитуванні або запису даних. Якщо значення **RDY_PROG =** 0, то пам'ять FRAM зайнята, якщо 1 – готова.

CALL – перехід у підпрограму. Рівень вкладення – 1.

RET – повернення з підпрограми.



Підпрограма виконується до моменту виконання оператора RET. Після чого, програма переходить до виконання кроку вказаного виходом RET відповідного оператора CALL. Виходи оператора RET (NC і RET) з'єднуються між собою і входом першого оператора A1 (точка START).

LOCK_SPn – перевірка блокування задатчика блока "АЛГОРИТМ", де n – ціле число 1 – 128. Приклад: LOCK_SP50 = 0 (0 – задатчик вільний, 1 – заблокований)

LOCK_SI(SPn) – перевірка блокування задатчика блока "АЛГОРИТМ", на який вказує значення задатчика блока SPn, де n – ціле число 1 – 128. Приклад: LOCK_SI(SP50) = 0 (0 – задатчик вільний, 1 – заблокований).

D_INPn – перевірка входу INPn, де n – ціле число 1 – 32, на подію фронт сигналу. Діапазон значень входу INPn - біт (0,1). Для визначення типу фронту використовується операція AND. Приклад: D_INP10 AND 0 – передній фронт по входу INP10, D_INP10 AND 1 – задній фронт по входу INP10.

D_INI(SPn) – перевірка входу INP на який вказує значення задатчика блока SPn, де n – ціле число 1 – 128, на подію фронт сигналу. Діапазон значень входу INI(SPn) - біт (0,1). Для визначення типу фронту використовується операція AND. Приклад: D_INI(SP10) AND 0 – передній фронт по входу INP10, D_INI(SP10) AND 1 – задній фронт по входу INP10.

Можливі типи операндів у правій частині умови: **таблиця 2**.

Операції з константами:

Const – константа. Діапазон цілих чисел: -1023 / +1023.

Операції з даними входів:

INPn – значення входу блока «АЛГОРИТМ» де n – ціле число 1 – 32. Діапазон значень входу INP - дійсні числа (float). Приклад: INP10.

INI(INPn) – значення входу блока "АЛГОРИТМ", на який вказує значення входу блока INPn, де n – ціле число 1 – 32. Приклад: INI(INP10).

INI(SPn) – значення входу блока "АЛГОРИТМ, на який вказує значення задатчика блока SPn, де n – ціле число 1 – 128. Приклад: INI(SP10).

INPn+SPm – значення суми входу INPn і задатчика SPm блока "АЛГОРИТМ", де n – ціле число 1 – 32, m – ціле число 1 – 64. Приклад: INP5+SP10.

INPn-SPm – значення різниці входу INPn і задатчика SPm блока "АЛГОРИТМ", де n – ціле число 1 – 32, m – ціле число 1 – 64. Приклад: INP5-SP10.

Операції з даними виходів:

OUTn – значення виходу блока «АЛГОРИТМ» де n – ціле число 1 – 32. Діапазон значень виходу OUT - дійсні числа (float). Приклад: OUT10.

Операції з задатчиками:

SPn – значення задатчик блока “АЛГОРИТМ”, де n – ціле число 1 – 128. Діапазон значень задатчика SPn - дійсні числа (float). Приклад: SP10.

SI(INPn) – значення задатчика блока “АЛГОРИТМ”, на який вказує значення входу блока INPn, де n – ціле число 1 – 32. Приклад: SI(INP10).

SI(SPn) – значення задатчика блока “АЛГОРИТМ”, на який вказує значення задатчика блока SPn, де n – ціле число 1 – 128. Приклад: SI(SP10).

SPn+SPm – значення суми задатчика SPn і задатчика SPm блока “АЛГОРИТМ”, де n – ціле число 1 – 32, m – ціле число 1 – 64. Приклад: SP5+SP10.

SPn-SPm – значення різниці задатчика SPn і задатчика SPm блока “АЛГОРИТМ”, де n – ціле число 1 – 32, m – ціле число 1 – 64. Приклад: SP5-SP10.

SI(SPn)+SPm – значення суми задатчика SP, на який вказує значення задатчика блока SPn, і задатчика SPm блока “АЛГОРИТМ”, де n – ціле число 1 – 32, m – ціле число 1 – 64. Приклад: SI(SP5)+SP10.

SI(SPn)-SPm – значення різниці задатчика SP, на який вказує значення задатчика блока SPn, і задатчика SPm блока “АЛГОРИТМ”, де n – ціле число 1 – 32, m – ціле число 1 – 64. Приклад: SI(SP5)-SP10.

SPn_SPm – масив задатчиків SP блока “АЛГОРИТМ”, для зчитування/запису пам'яті FRAM. SPn – початковий номер задатчика, SPm – кількість за датчиків, де n – ціле число 1 – 128, m – ціле число 1 – 16. Приклад: SP64_SP10.

SPn*SPm – значення добутку задатчика SPn і задатчика SPm блока “АЛГОРИТМ”, де n – ціле число 1 – 32, m – ціле число 1 – 64. Приклад: SP5*SP10.

SPn/SPm – значення частки задатчика SPn і задатчика SPm блока “АЛГОРИТМ”, де n – ціле число 1 – 32, m – ціле число 1 – 64. Приклад: SP5/SP10.

SI(SPn)*SPm – значення добутку задатчика SP, на який вказує значення задатчика блока SPn, і задатчика SPm блока “АЛГОРИТМ”, де n – ціле число 1 – 32, m – ціле число 1 – 64. Приклад: SI(SP5)*SP10.

SI(SPn)/SPm – значення частки задатчика SP, на який вказує значення задатчика блока SPn, і задатчика SPm, де n – ціле число 1 – 32, m – ціле число 1 – 64. Приклад: SI(SP5)/SP10.

Операції з таймерами:

TIMERn_S_M_H – режим роботи таймера TIMER блока “АЛГОРИТМ” у секундах, хвилинах або годинах, де n - ціле число 1 – 32, S_M_H – ціле число 0 – 2. Приклад: TIMER10_1 – таймер 10 у хвилинах.

Операції з даними функціональних блоків:

FBn_OUTm – значення виходу OUTm функціонального блока FBn, де n – ціле число 1 – 96, m – ціле число 1 – 16. Приклад: FB50_OUT5.

FBn_SPm – значення задатчика SPm функціонального блока FBn, де n – ціле число 1 – 96, m – ціле число 1 – 16. Приклад: FB50_SP5.

FBn_OUI(SPm) – значення виходу OUT функціонального блока FBn, на який вказує значення задатчика блока SPm, де n – ціле число 1 – 96, m – ціле число 1 – 128. Приклад: FB50_OUI(SP5).

FBn_SI(SPm) – значення задатчика SP функціонального блока FBn, на який вказує значення задатчика блока SPm, де n – ціле число 1 – 96, m – ціле число 1 – 128. Приклад: FB50_SI(SP5).

Інші операції:

ABS_SPn - абсолютне значення задатчика SPn блока “АЛГОРИТМ”, де n – ціле число 1 – 128. Приклад: ABS_SP10.

SQRT_SPn – квадратний корінь задатчика SPn блока “АЛГОРИТМ”, де n – ціле число 1 – 128. Приклад: SQRT_SP10.

LOCK_n – блокування задатчика SPn блока “АЛГОРИТМ”, де n – ціле число 0 – 2. 0 – розблокування SP, 1 – блокування SP, 2 – розблокування усіх SP. Можливі дії: SPn = , SI(SPn) =. Приклад: SP10 = LOCK_1 – блокує доступ до за датчика SP10.

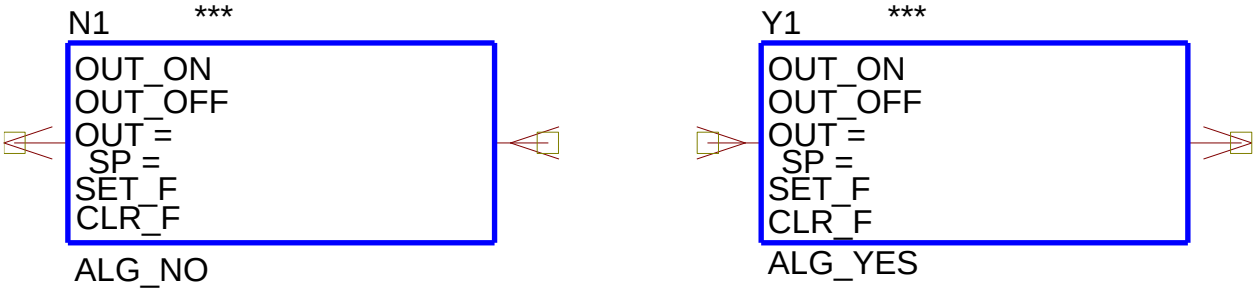
MIN_SPn_const – пошук найменшого значення задатчика у масиві, де n – номер першого за датчика, const – кількість елементів у масиві. n - ціле число 1 – 128, const - ціле число 1 – 16. Результат пошуку (мінімальне значення) розміщується у задатчику вказаному у лівій частині виразу, номер задатчика у масиві – у наступному задатчику. Приклад: SP5 = MIN_SP10_12. SP5 – найменше значення масиву, SP6 – номер мінімального SP.

MAX_SPn_const – пошук найбільшого значення задатчика у масиві, де n – номер першого задатчика, const – кількість елементів у масиві. n - ціле число 1 – 128, const - ціле число 1 – 16. Результат пошуку (максимальне значення) розміщується у задатчику вказаному у лівій частині виразу, номер задатчика у масиві – у наступному задатчику. Приклад: SP5 = MAX_SP10_12. SP5 – найбільше значення масиву, SP6 – номер мінімального SP.

SUM_SPn_const - сума елементів масива, де n – номер першого задатчика, const – кількість елементів у масиві. n - ціле число 1 – 128, const - ціле число 1 – 16. Приклад: SP5 = SUM_SP10_12.

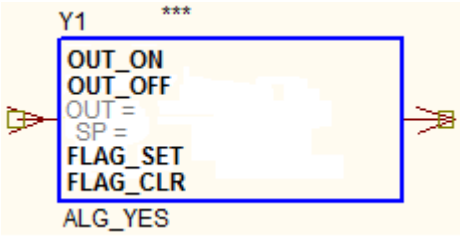
BYTE_SP0_3_const - Бітовий доступ до окремих байт задатчика SP1-128. Const – ціле число 0-255

5. БЛОКИ ALG_YES і ALG_NO



Блоки **ALG_YES і ALG_NO** функціонально тотожні. Вони виконують певні дії з виходами, задатчиками, прапорцями і таймерами блока залежно від результату операції блоку **ALG_IF**. Блоки ALG_YES і ALG_NO виконують різні дії: логічні, математичні, пересилки і збереження даних. Вони можуть оперувати як з даними, що знаходяться в середині блока АЛГОРИТМ (входи, виходи, таймери, прапорці, задатчики…), так і з даними зовнішніх функціональних блоків.

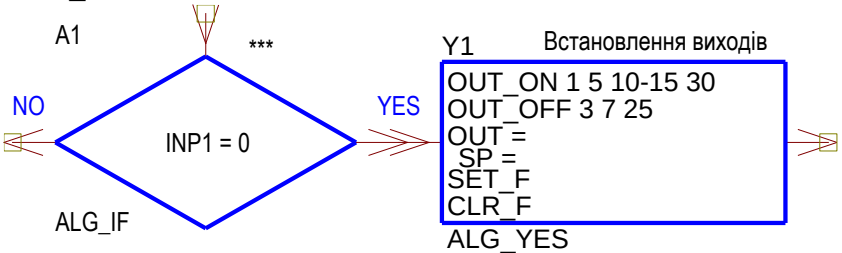
6. ЛОГІЧНІ ОПЕРАЦІЇ БЛОКА ALG_YES і ALG_NO



Логічні операції з виходами:

OUT_ON n – встановлює виходи OUTn блока "АЛГОРИТМ" рівними логічній "1". Вираз n перераховує потрібні номери виходів, або діапазон номерів виходів, які встановлюються рівними лог. "1". Номери n лежать в діапазоні цілих чисел 1- 32. Роздільним знаком між номерами слугує пробіл " ". Діапазон виходів вказується за допомогою знака дефіс(мінус) "-", без пробілів. Приклад: OUT_ON 1 5 10-20.

OUT_OFF n – встановлює виходи OUTn блока "АЛГОРИТМ" рівними логічному "0". Вираз n перераховує потрібні номери виходів, або діапазон номерів виходів, які встановлюються рівними лог. "0". Номери n лежать в діапазоні цілих чисел 1- 32. Роздільним знаком між номерами слугує пробіл " ". Діапазон виходів вказується за допомогою знака дефіс(мінус) "-", без пробілів. Приклад: OUT_OFF 1 5 10-20.

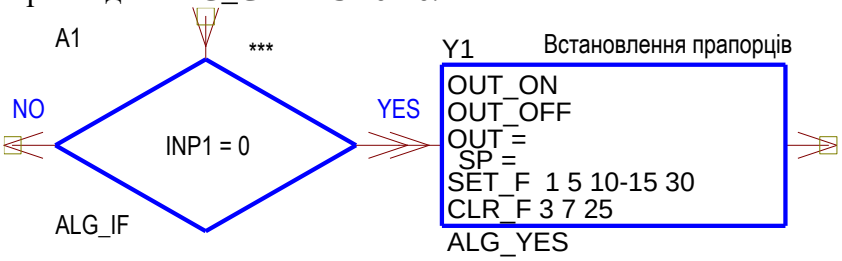


Логічні операції з прапорцями:

FLAG_SET n – встановлює прапорці FLAGn блока "АЛГОРИТМ" рівними логічній "1". Вираз n перераховує потрібні номери прапорців, або діапазон номерів прапорців, які встановлюються рівними лог. "1". Номери n лежать в діапазоні цілих чисел 1- 32. Роздільним знаком між номерами

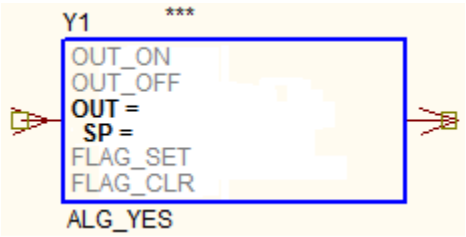
слугує пробіл ” “. Діапазон прапорців вказується за допомогою знака дефіс(мінус) “-”, без пробілів. Приклад: FLAG_SET 1 5 10-20.

FLAG_CLR n – встановлює прапорці FLAGn блока “АЛГОРИТМ” рівними логічному “0”. Вираз n перераховує потрібні номери прапорців, або діапазон номерів прапорців, які встановлюються рівними лог. “0”. Номери n лежать в діапазоні цілих чисел 1- 32. Роздільним знаком між номерами слугує пробіл ” “. Діапазон прапорців вказується за допомогою знака дефіс(мінус) ”-“, без пробілів. Приклад: FLAG_CLR 1 5 10-20.



Якщо одночасно встановити виходи (прапорці) в логічну “1” і “0”, тоді виконується інверсія виходу (прапорця). Приклад: OUT_ON 5 і OUT_OFF 5.

7. ДІЯ БЛОКА ALG_YES і ALG_NO



В одному блоку можуть виконуватися дві будь-які дії, що визначені у таблиці 3. На схемі блока вони умовно позначені надписами OUT =, SP =.

Можливі типи дій: **таблиця 3**.

OUT =, SP = - немає дії.

Встановлення виходів:

OUTn = IF_B - вихід OUTn блока “АЛГОРИТМ” встановлюється рівним значенню результату дії IF_B (Таблиця 2), де n – ціле число 1 – 32. Приклад: OUT5 = SP10.

OUTn += IF_B - вихід OUTn блока “АЛГОРИТМ” встановлюється рівним сумі виходу OUTn і результату дії IF_B (Таблиця 2), де n – ціле число 1 – 32. Приклад: OUT5 += SP10 (OUT5 = OUT5+SP10).

OUTn -= IF_B - вихід OUTn блока “АЛГОРИТМ” встановлюється рівним різниці виходу OUTn і результату дії IF_B (Таблиця 2), де n – ціле число 1 – 32. Приклад: OUT5 -= SP10 (OUT5 = OUT5-SP10).

OUTn *= IF_B - вихід OUTn блока “АЛГОРИТМ” встановлюється рівним добутку виходу OUTn і результату дії IF_B (Таблиця 2), де n – ціле число 1 – 32. Приклад: OUT5 *= SP10 (OUT5 = OUT5*SP10).

OUTn /= IF_B - вихід OUTn блока “АЛГОРИТМ” встановлюється рівним частці виходу OUTn і результату дії IF_B (Таблиця 2), де n – ціле число 1 – 32. Приклад: OUT5 /= SP10 (OUT5 = OUT5/SP10).

OUI(INPn) – значення виходу блока “АЛГОРИТМ”, на який вказує значення входу INPn, встановлюється рівним результату дії IF_B (Таблиця 2), де n – ціле число 1 – 32. Приклад: OUI(INP10) = SP10.

OUI(SPn) – значення виходу блока “АЛГОРИТМ”, на який вказує значення задатчика SPn, встановлюється рівним результату дії IF_B (Таблиця 2), де n – ціле число 1 – 128. Приклад: OUI(SP5) = SP10.

Встановлення задатчиків:

SPn = IF_B - задатчик SPn блока “АЛГОРИТМ” встановлюється рівним значенню результату дії IF_B (Таблиця 2) Де n – ціле число 1 – 128. Приклад: SP5 = SP10.

SPn += IF_B - задатчик SPn блока “АЛГОРИТМ” встановлюється рівним сумі задатчика SPn і результату дії IF_B (Таблиця 2) Де n – ціле число 1 – 128. Приклад: SP5 += SP10 (SP5 = SP5+SP10).

SPn -= IF_B - задатчик SPn блока “АЛГОРИТМ” встановлюється рівним різниці задатчика SPn і результату дії IF_B (Таблиця 2) Де n – ціле число 1 – 128. Приклад: SP5 -= SP10 (SP5 = SP5-SP10).

SPn *= IF_B - задатчик SPn блока “АЛГОРИТМ” встановлюється рівним добутку задатчика SPn і результату дії IF_B (Таблиця 2) Де n – ціле число 1 – 128. Приклад: SP5 *= SP10 (SP5 = SP5*SP10).

SPn /= IF_B - задатчик SPn блока “АЛГОРИТМ” встановлюється рівним частці задатчика SPn і результату дії IF_B (Таблиця 2) Де n – ціле число 1 – 128. Приклад: SP5 /= SP10 (SP5 = SP5/SP10).

SI(INPn) = IF_B – задатчик блока “АЛГОРИТМ”, на який вказує значення входу INPn, встановлюється рівним результату дії IF_B (Таблиця 2), де n – ціле число 1 – 32. Приклад: SI(INP10) = SP10.

SI(SPn) = IF_B – задатчик блока “АЛГОРИТМ”, на який вказує значення задатчика SPn, встановлюється рівним результату дії IF_B (Таблиця 2), де n – ціле число 1 – 128. Приклад: SI(SP10) = SP20.

SI(SPn) += IF_B – задатчик блока “АЛГОРИТМ”, на який вказує значення задатчика SPn, встановлюється рівним сумі цього задатчика і результату дії IF_B (Таблиця 2), де n – ціле число 1 – 128. Приклад: SI(SP10) += SP20 (SI(SP10) = SI(SP10) + SP20).

SI(SPn) -= IF_B – задатчик блока “АЛГОРИТМ”, на який вказує значення задатчика SPn, встановлюється рівним різниці цього задатчика і результату дії IF_B (Таблиця 2), де n – ціле число 1 – 128. Приклад: SI(SP10) -= SP20 (SI(SP10) = SI(SP10) – SP20).

SI(SPn) *= IF_B – задатчик блока “АЛГОРИТМ”, на який вказує значення задатчика SPn, встановлюється рівним добутку цього задатчика і результату дії IF_B (Таблиця 2), де n – ціле число 1 – 128. Приклад: SI(SP10) *= SP20 (SI(SP10) = SI(SP10) * SP20).

SI(SPn) /= IF_B – задатчик блока “АЛГОРИТМ”, на який вказує значення задатчика SPn, встановлюється рівним частці цього задатчика і результату дії IF_B (Таблиця 2), де n – ціле число 1 – 128. Приклад: SI(SP10) /= SP20 (SI(SP10) = SI(SP10) / SP20).

Обнуління таймерів:

TIMERn = 0 – обнуління таймера TIMERn), де n – ціле число 1 – 32. Приклад: TIMER5 = 0.

Встановлення виходів виходів і задатчиків функціональних блоків:

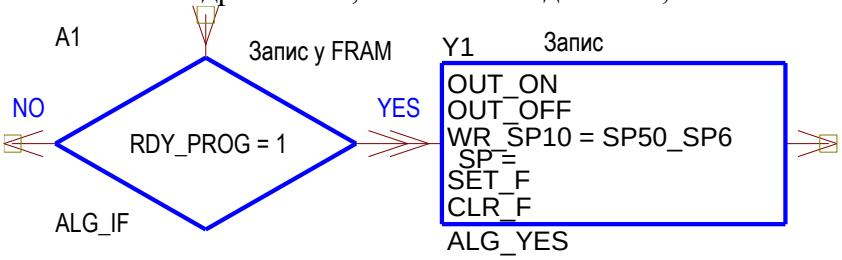
FBn_SPm = IF_B – значення задатчика SPm функціонального блока FBn, встановлюється рівним результату дії IF_B (Таблиця 2), де n – ціле число 1 – 96, m – ціле число 1 – 16. Приклад: FB50_SP5 = SP10.

FBn_SI(SPm) = IF_B – значення задатчика SP функціонального блока FBn, на який вказує значення задатчика блока SPm, встановлюється рівним результату дії IF_B (Таблиця 2), де n – ціле число 1 – 96, m – ціле число 1 – 16. Приклад: FB50_SI(SP5) = SP10.

Інші дії:

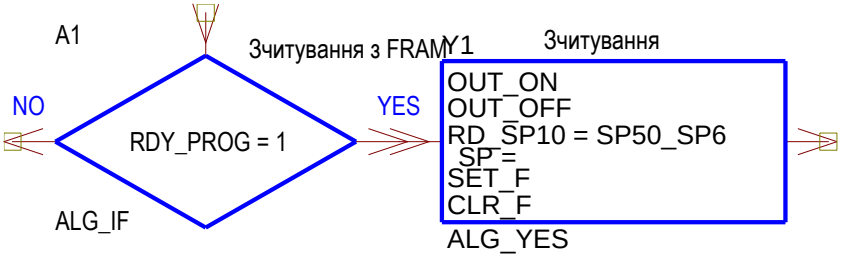
WR_SPn = SPm_SPk – запис у FRAM задатчиків блока “АЛГОРИТМ”, де n – ціле число 1 – 128, m – ціле число 1 – 128, k– ціле число 1 – 16. Об’єм пам'яті FRAM складає 8192 дійсних числа (float). Значення задатчика SPn вказує на початкову адресу у FRAM (0 - 8191), куди записуються дані. Задатчик SPm (1 – 128) вказує на початковий задатчик SP. Значення задатчика SPk (1 – 128) вказує кількість задатчиків SP для запису. Готовність до запису даних перевіряється за допомогою значення **RDY_PROG –** стан готовності пам’яті FRAM. Якщо значення **RDY_PROG =** 0, то пам'ять FRAM зайнята, якщо 1 – готова.

Приклад: якщо SP10 = 1000, SP6 = 10 - команда WR_SP10 = SP50_SP6, запише у пам'ять FRAM починаючи з адреси 1000, масив з 10 задатчиків, починаючи з задатчика SP50.



RD_SPn_SPm = SPk – зчитування з FRAM даних в задатчики блока “АЛГОРИТМ”, де n – ціле число 1 – 128, m – ціле число 1 – 16, k– ціле число 1 – 128. Об’єм пам'яті FRAM складає 8192 дійсних числа (float). Значення задатчика SPk вказує на початкову адресу у FRAM (0 - 8191), для зчитування даних. Задатчик SPn (1 – 128) вказує на початковий задатчик SP. Значення задатчика SPm (1 – 16) вказує кількість задатчиків SP для зчитування. Готовність до зчитування даних

перевіряється за допомогою значення **RDY_PROG –** стан готовності пам’яті FRAM. Якщо значення **RDY_PROG =** 0, то пам'ять FRAM зайнята, якщо 1 – готова.
Приклад: якщо SP5 = 10, SP6 = 1000 - команда RD_SP50_SP5 = SP6, зчитає з пам'яті FRAM починаючи з адреси 1000, у задатчики блока “АЛГОРИТМ”, масив з 10 задатчиків, починаючи з задатчика SP50.



CLR_SPn_SPm = 0 – обнулення масиву задатчиків SP блока “АЛГОРИТМ”, де n – ціле число 1 – 128, m – ціле число 1 – 16. Команда записує число “0” у масив задатчиків починаючи з номера SPn у кількості вказаній у задатчику SPm. Приклад: CLR_SP50_SP10. Якщо SP10 = 20, буде обнулено 20 задатчиків SP, починаючи з SP50.

ROUND_SPn = const – округлення значення задатчика SP, де n – ціле число 1 – 128, const – ціле число 0 – 2. Якщо const = 0 - округлення до цілих, 1 – десятих, 2 – сотих. Приклад: ROUND_SP10 = 1, якщо SP10 = 35.123 – результат округлення: SP10 = 35.1.

LIM_MIN_SPn = IF_B – обмеження до мінімуму значення задатчика SP блока “АЛГОРИТМ”), де n – ціле число 1 – 128. Величина мінімуму є результатом дії IF_B (Таблиця 2. Приклад: LIM_MIN_SPn = -10.

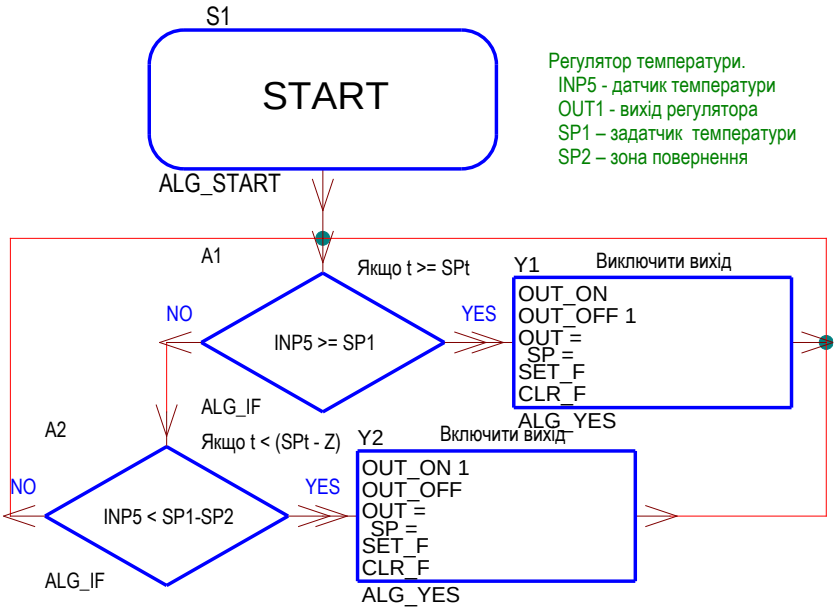
LIM_MAX_SPn = IF_B – обмеження до максимуму значення задатчика SP блока “АЛГОРИТМ”, де n – ціле число 1 – 128. Величина максимуму є результатом дії IF_B (Таблиця 2). Приклад: LIM_MAX_SPn = 10.

SPn &= IF_B – логічна бітова операція AND над окремими байтами за датчика SP”), де n – ціле число 1 – 128. Наприклад SP5 &= 16. Скинути біт 4 у SP5 в «0».

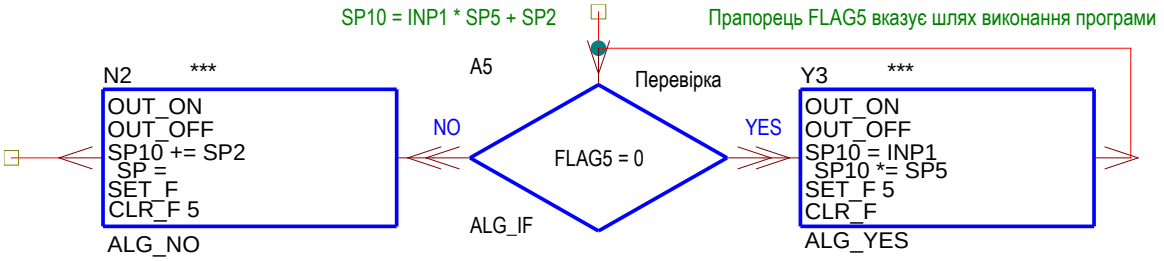
SPn |= IF_B – логічна бітова операція OR над окремими байтами за датчика SP”), де n – ціле число 1 – 128. Наприклад SP5 |= 16. Встановити біт 4 у SP5 в «1».

8. ПРИКЛАДИ ПРОГРАМУВАННЯ БЛОК_СХЕМИ

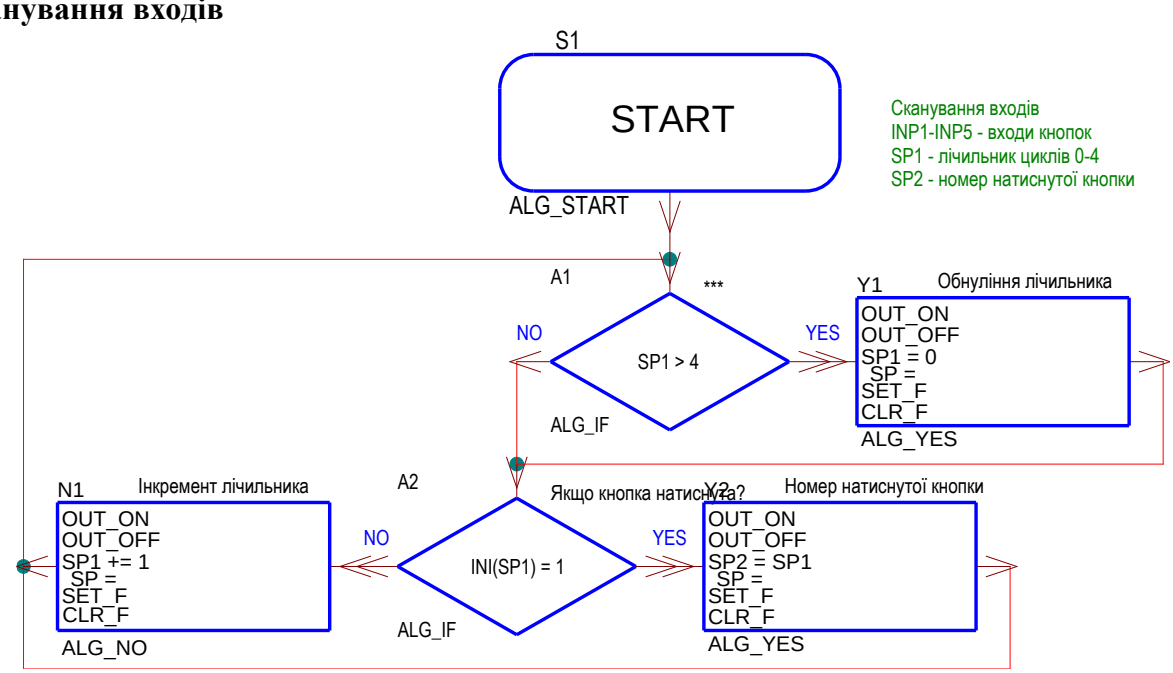
Регулятор температури.



Виконання у одному кроці дій ALG_YES і ALG_NO



Сканування входів



.