

**Методичні вказівки з програмування ПЛК ФК5001 в
інтегрованому середовищі Транскан/Скансот.**

Контрольні приклади

Зміст

1. ВСТУП
 - 1.1. Типи мов програмування ПЛК.5-рівнева модель ІЕС 1131-3
 - 1.2. Класифікація програмно-технічного забезпечення інформаційно-керуючих обчислювальних комплексів
 - 1.3. Класифікація мов програмування.
 2. ОСНОВНІ ВІДОМОСТІ ЩОДО ПРОГРАМУВАННЯ ПЛК РК 5100
 - 2.1. Технічні характеристики ПЛК РК 5100
 - 2.2. Типові рішення по розробці програмного забезпечення ПЛК
 - 2.2.1 Структура робочої програми
 - 2.2.2 Реалізація математичних функцій
 - 2.2.3 Реалізація логічних функцій
 - 2.2.3 Програмування АЦП і ЦАП
 - 2.3. Структура протоколу обміну даними ПК–ПЛК
 3. МОДЕЛЮВАННЯ ОДНОКОНТУРНОЇ САР
 - 3.1. Послідовність перетворювання диференційного рівняння об'єкта в різницеве рівняння
 - 3.2. Послідовність виконання лабораторної роботи з використанням ПЗ “Гранскан”
 - 3.3. Послідовність виконання лабораторної роботи з використанням ПЗ “Скансот”
- ДОДАТОК 1. Текст драйвера ПК для ПЛК РК 5100
- ДОДАТОК 2. Тексти типових програм для ПЛК РК 5100.
- ДОДАТОК 3. Зміст байту атрибутів при програмуванні АЦП.
- ДОДАТОК 4. Системні відмови.
- ДОДАТОК 5. Початкові адреси пам'яті модулів та регістрів. Адреси звернення до регістрів модуля послідовного вводу–виводу при різних місцях встановлення на каркасі.
- ДОДАТОК 6. Текст програми ТЕСТ.

1. ВСТУП

1.1 Типи мов програмування ПЛК.5-рівнева модель IEC 1131-3

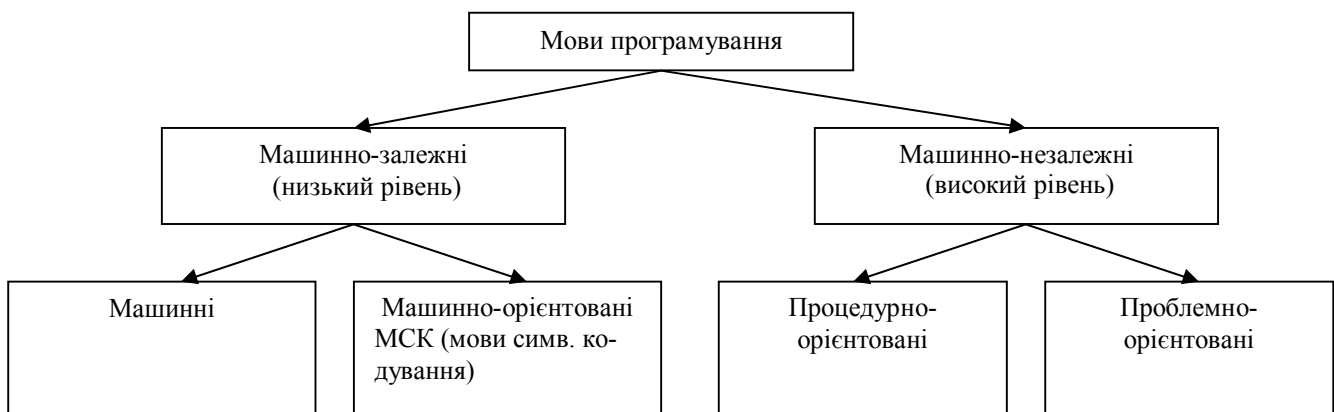
Для мов програмування контролерів був розроблений міжнародний стандарт IEC 1131-3, котрий передбачає 5 рівнів мов програмування (тому мови програмування контролерів називають мовами Step 5). Обов'язковою умовою для мов Step 5 є можливість програмування в термінах звичайних релейних схем (LD – сходинкова діаграма). Це дає можливість заводському інженеру переписати релейну схему на мову контролера, не вивчаючи спеціально мову програмування. Основною мовою програмування для кваліфікованого розробника є мова IL, тобто мова символного кодування (варіант асемблера) з елементами мов високого рівня (інструкції лічильників, таймерів та ін.). Але в Step 5 закладено й підтримку концепції процедурно-орієнтованої мови програмування високого рівня ST- структурований текст. Найбільш відомі промисловий BASIC (CAMBASIC, TechnoBASIC) та PASCAL. Термін “промисловий” в даному випадку означає, що діалект BASIC та PASCAL містить у собі спеціалізовані інструкції структурування, лічильників, таймерів, інструкції роботи з модулями ПЗО, що програмуються та ін. Крім цього, концепція програмування Step 5 підтримує і концепцію проблемно-орієнтованих мов високого рівня (мови SFD, SFC). Вони є мовами візуального програмування. Усі програми промислових контролерів працюють в real-time.

1.2 Класифікація програмно-технічного забезпечення інформаційно-керуючих обчислювальних комплексів

Інформаційно-керуючі обчислювальні комплекси є сукупністю різноманітних промислових локальних обчислювальних мереж. Підключення нових контролерів до мережі та їх відключення здійснюється за принципом Plug-n-Play без відключення живлення. В промислових мережах здійснюється тільки передача даних, передача коду для виконання на більш потужних ЗОТ не актуальна. Для промислових контролерів актуальний розподіл різних видів програм користувача між окремими станціями. Програмно-технічні засоби дворівневої АСУТП класифікують таким чином:

- SCADA/HMI – Супервізорний контроль і управління / Інтерфейс оператора (на робочих станціях)
- PLC/KTU – Програмовані логічні контролери / віддалені процесові станції (центральні і польові процесові станції).

1.3 Класифікація мов програмування



Машинні мови (послідовності цифр-символів) називають також автокодами. МСК називають мнемосодами, транслятори з цих мов – асемблерами. Для мнемосокодів характерним є запис “1:1”, тобто одна команда машинної мови представляється в виді одного символу. Відомі також асемблери, побудовані за принципом “1 команда асемблера – декілька машинних команд”, такий асемблер називається макроасемблером. Мова контролера IL є макроасемблером з елементами мови високого рівня. Мова IL контролера РК 5100 є процедурно-орієнтованою з елементами проблемної орієнтації (вбудовані інструкції таймерів, лічильників і т.п.).

2. ОСНОВНІ ВІДОМОСТІ ЩОДО ПРОГРАМУВАННЯ ПЛК РК 5100

2.1. Технічні характеристики ПЛК РК 5100

ПЛК РК-5100 є поширена сучасна розробка програмуємого логічного контролера серії MicroDAT (МБ,ФК-5001,РК-5100) розробленої на базі контролера Siemens SIMATIC S5. Розробник – фірма АО “Росток-Спарк” (м.Київ).

ПЛК РК-5100 має процесор, ПЗП, ОЗП, модулі вводу-виводу. Модулі вводу-виводу поділяються на модулі ПЗО (АЦП,ЦАП,ДЦП,ЦДП) та модулі послідовного вводу-виводу (МПВВ). Модулі ПЗО призначені для вводу-виводу від об'єкта та на об'єкт аналогових та дискретних сигналів. МПВВ призначені для цифрової передачі даних та мають стандартні інтерфейси: RS232C (до 15м) та CL (струмова петля; до 1км).

Процесор контролера зібраний на мікросхемах та є одноадресною машиною. Програмне забезпечення записується в ПЗП. Мова програмування за класифікацією ІЕС 1131-3: LD та IL. ОЗП контролера ємністю 16 КБ (тобто 8 кілослів інструкцій) поділяється на таблицю даних ТД та захищену область ЗО. ТД входить до складу процесорного модуля. ЗО є змінним, додається до слоту процесорного модуля у вигляді окремої плати. Після наладки програм користувача вона за допомогою пристрою запису програм ПЗП може бути записана в перепрограмуємий ПЗП (ППЗП). Модуль ППЗП додається в слот процесорного модуля замість ОЗП. ППЗП є енергонезалежним, тобто записана до нього програма користувача зберігається і при відключенні живлення (для збереження програми користувача в ОЗП при відключенні живлення передбачається батарейне живлення). При відключенні живлення програма тільки зберігається, але не виконується. ТД використовується як БД. В ЗО розміщують: область конфігурації модулів вводу-виводу, область констант (ці параметри в ході виконання програми користувача можуть тільки читатись, но не змінюватись), область програми користувача (ПрК).

В області конфігурації вказується кількість та тип модулів вводу-виводу, а також адреса початку ПрК. На панелі мікропроцесорного модулю є два тумблери: верхній тумблер запускає процесор, нижній механічно блокує доступ до ЗО. В режимі програмування обидва тумблери знаходяться в верхньому положенні. Користувач заносить ПрК в ЗО (а також область конфігурації та область констант) за допомогою пристрою програмування та налагодження (програма-тора). Програма-тор підключається до інтерфейсу “струмова петля” на мікропроцесорному модулі (роз'єм можна під'єднувати без відключення живлення).

Можливо програмування контролера від ПЕОМ з використанням транслятору ТранСКАН. Початковий код на мові контролеру компілюється ТранСКАН в об'єктний код, який записується в ОЗП по інтерфейсу “струмова петля”. В контролері об'єктний код виконується в режимі інтерпретації (аналог режиму мікропрограмування).

Для створення НМІ-програм використовується пакет СканСОТ, який створює ехе-файл, що виконується на робочій станції. Для підключення до робочої станції до 16 контролерів використовується програмуємий комутатор СК-16. Перемикання каналів комутатора та підтримка обміну даними виконується резидентними драйверами СканСЕТЬ та СканСЕРВЕР. Мова контролера містить інструкції таймерів, лічильників, читання та запису в слова, пересування файлів, арифметичні операції.

ОСОБЛИВОСТІ ПРОГРАМУВАННЯ: процесор не працює з числами, меншими за нуль; процесор працює тільки з цілими числами. АЦП є 12-разрядний, 13й біт –знаковий. Цифровий сигнал змінюється від 0 до $2^{12}-1=4095$ (тобто має 4096 значень, включаючи 0).

Структура програми користувача (інструкції структуровання):

НСТ XX – початок сегменту XX;

НБЛ YYY – початок блоку YYY;

НПП ZZZ – початок підпрограми ZZZ;

КОН – закінчення програми.

Інструкція ПБЛ YYY є інструкцією безумовного переходу на блок YYY.

Запуск контролера до роботи здійснюється переводом тумблеру в нижнє положення. При цьому починає працювати процесор та блокується доступ користувача к ЗО. Процесор перевіряє конфігурацію та готовність модулів ПЗО та тестує ПрК. При відмовах процесора, пам'яті, помилках загорається червоний індикатор на мікропроцесорном модулі, при відмовах модулів вводу-виводу чи помилкової конфігурації (невідповідності закладеної конфігурації модулів дійсно встановленим) загорається індикатор ВВ. При нормальній роботі загорається індикатор зеленого кольору ГОТ (готовність). Процесор, якщо немає відмов, розраховує контрольну суму програми та записує її до відповідного слова області конфігурації, після чого вмикає індикатор ГОТ, зчитує з області конфігурації адресу початку ПрК та переходить до її виконання. Після вмикання

ГОТ користувач переводить в нижнє положення нижній тумблер, механічно блокуючи доступ до ЗО.

При повторному перезапуску процесора він перевіряє співпадання записаної до області конфігурації контрольної суми розрахований для програми відразу після перезапуску. При співпаданні контрольних сум та при наявності сигналів від модулів ПЗО, вказаних в конфігурації, починається виконання програми, тобто контролер переходить у стан готовності (загорається зелений індикатор ГОТ). Якщо немає відповідних сигналів від модулів ПЗО, загорається червоний індикатор ВВ. При неспівпаданні контрольних сум та інших відмовах процесора чи запуску програми загорається червоний індикатор ПР. В обох варіантах зелений індикатор ГОТ не горить. При загоранні ПР та ВВ можливо визначити причину відмови, проглянувши програматором чи за допомогою ПЕОМ зарезервовані слова таблиці даних процесора. В цих словах одиничне значення того чи іншого біта виказує помилку.

2.2 Типові рішення по розробці програмного забезпечення ПЛК

2.2.1 Структура робочої програми.

Робоча програма складається з інструкцій мови програмування.

Інструкції об'єднуються в програмні блоки та підпрограми, що являють собою основні структурні елементи програми. Блоки, в свою чергу, можуть бути об'єднані в сегменти.

Програма може містити:

-до 256 блоків з номерами 0...377

-до 256 підпрограм з номерами 0...377

-до 32 сегментів з номерами 0...37

Для задання структури програми використовуються інструкції структурування.

Порядок слідування блоків, сегментів та підпрограм може бути довільним.

Якщо програма не містить сегментів, то виконання програми у кожному скані розпочинається з блоку з номером 000. Для реалізації програм з більш складною структурою використовуються інструкції виклику блоків і підпрограм.

Якщо програма складається з одного сегменту, то в ньому може бути активним лише один блок. Він виконується повністю за один скан. Якщо у блоці є команда переходу на інший блок сегменту, то цей блок буде виконуватись у наступному скані програми. Отже, за один скан виконується лише один активний блок.

У разі, якщо програма складається з декількох сегментів, то і в цьому випадку кожен сегмент має лише один активний блок. Отже, за один скан виконується лише по одному блоку у кожному сегменті. Після вимикання живлення, або після скидання сегмента, активним стає блок, що знаходиться безпосередньо за інструкцією початку сегменту.

Приклад складно структурованої програми наведений у додатку 2. Введення програми в контроллер потрібно розпочинати з задання маски обміну. Приклад задання маски обміну приведений у додатку 2.

2.2.2 Реалізація математичних функцій.

Арифметичні операції виконуються шляхом дії над числами, одне з яких знаходиться в аккумуляторі, інше- за адресою слова. Результат заноситься в аккумулятор, а далі- в будь-яку іншу дільницю пам'яті, вказану в програмі.

В разі додавання чи множення, якщо в результаті отримується число більше, ніж 4095, то вмикається біт 14 в слові, де записується результат.

В разі віднімання, якщо отримуємо від'ємне число, також вмикається біт 14.

Якщо при діленні отримуємо число, менше за 1, то в результаті записується 0.

Приклади програм, що застосовують операції додавання, віднімання, множення та ділення, а також операцію переприсвоєння можна дивитись у додатку 2.

2.2.3 Реалізація логічних функцій.

Мова програмування контроллера зроблена на зразок мов програмування високого рівня. Якщо в тексті програми трапляється умова і вона є істинною, то виконуються команди, що знаходяться за цією умовою. У випадку, якщо умова є хибною, програма шукає іншу умову і, в разі її знаходження, перевіряє її. Якщо ж інша умова не буде знайдена, то програма звертається до першої команди структурування.

Винятком з цього правила є випадок, коли після інструкції перевірки умови стоїть інструкція структурування ПБЛ. Якщо умова є істинною, то виконується ця інструкція. В разі, якщо умова є хибною, то інструкція ПБЛ пропускається, а виконується наступна за ПБЛ інструкція. Лише цим винятком мова програмування контроллера схожа на мову Ассемблера.

Так що при використанні умов потрібно бути особливо уважним, тому що програма може зависнути.

Реалізацію логічних функцій виду I, АБО, НІ, І-НІ, АБО-НІ на прикладі опитуння контактів, а також релейні схеми можна бачити в додатку 2.

Ще одним прикладом логічних операцій є цикли. Так як у мові програмування контролера немає інструкцій циклу, то їх можна замінити інструкцією структурування ПБЛ, пересилаючи виконання програми на початок блоку, чи на блок, що звертається до цього блоку. Приклади циклів з передумовою з виходом по логічному 0 чи 1, з післямовою звиходом по логічному 0 чи 1, з лічильником циклів, а реалізацію умовного, безумовного переходів, а також переходу по вибору можемо бачити в додатку 2.

4.4. Програмування АЦП та ЦАП.

Модулі АЦП та ЦАП - окремі два модулі, яких додатково потрібно програмувати. Вони мають свої процесори, які потребують програмування, без якого вони не функціонують.

Програмування АЦП.

Суть програмування АЦП полягає в тому, щоб записати слово стану та сформувати масив атрибутів.

Для запису слова стану АЦП використовують дві команди: ЧТР та ЗПР. Ми маємо право змінювати лише старший байт слова стану. Молодший байт формується при тестуванні АЦП автоматично.

Формування масиву атрибутів залежить від виду оброблюваного сигналу, схем підключення каналів, які ми використовуємо.

Нижче буде приведений приклад програми для програмування АЦП в разі диференційного підключення джерел сигналів постійного струму величиною 0...20 мА. В прикладі використано чотири джерела сигналу, для приєднання яких попарно використані такі канали:

K001 та K002

K003 та K004

K005 та K006

K007 та K007

Для виконання програми в відповідні дільниці пам'яті були занесені константи, які приведені у додатку 14. Інші випадки при формуванні масиву атрибутів також приведені у додатку 14.

Текст програми:

НБЛ 001

УНС 000

СБС 177

ПФМ 177

УНС 020

ПРС 0030 -

ПРС 1022 | Перенос файла, що розпочинається з адреса 0030

ПРК 0004 | в АЦП за адресою 22 довжиною 4 слова за 1 скан

ПРК 0004 -

ЧТР 010 Прочитати реєстр АЦП

КБС 027 Загрузити константу в старший байт слова

ЗПР 010 Записати в реєстр АЦП

Програмування ЦАП.

Суть програмування ЦАП полягає в формуванні слова атрибутів, що полягає у встановленні каналів ЦАП на видачу сигналів потрібного виду та діапазону.

Нижче приведено приклад програми, за допомогою якої усі чотири канала ЦАП (тобто K000, K001, K002, K003) програмуються на видачу сигналів (0-20) мА постійного струму.

Для роботи цієї програми в певну дільницю пам'яті була занесена константа, що формує слово атрибутів.

10034 = 377377

Текст програми

НЛБ 002

УНС 020

ЧТС 034 Прочитати слово за адресою 034

ЗПР 021 Записати це слово в реєстр ЦАП

В додатку приведено текст програми з коментарями, яка тестує злагоджену спільну роботу АЦП та ЦАП.

2.3 Структура протоколу обміну даними ПК-ПЛК

Топологія - радіальна.Інтерфейс-RS232C.

Швидкість обміну – 9,6 Кбод. Відстань – до 500м.

Фізичне середовище – кручена пара. Схема підключення – нуль-модем. Тип протоколу – асинхронний, байт-орієнтований, напівдуплексний. Тип байта – октет. Код обміну інформацією – двоїчний формат. Менеджер обміну – ПК. Структура інформаційного кадру: службовий байт, дейтаграма, адреса. Види службових байтів: ЗАПИСЬ,ЧТЕНИЕ (ГОТОВ-сигнальний байт). Склад дейтаграми - два інформаційних байти. Структура дейтаграми: молодший байт, старший байт. Адреса: 00000-07777. Спосіб захисту від несанкціонованого доступу (запис) – завдання в ПК нижньої та верхньої адреси поля запису.

Спосіб захисту від помилок при запису та читанні – декілька повторів сеансів запису-читання. Відпрацювання режимів «запит-відповідь» – додання/вилучення слова. Готовність ПЛК – передача в ПК сигнального байта ГОТОВ.

Режим готовності:

ПЛК-ПЕОМ : ГОТОВ

Запит від ПЕОМ до ПЛК на читання з ПЛК та відповідь ПЛК:

ПЕОМ-ПЛК

ЧТН мол.байт адреси ст.байт адреси

ПЛК-ПЕОМ

ЧТН мол.інф байт ст.інф. байт мол.байт адреси ст.байт адреси

Запит від ПЕОМ до ПЛК на запис до ПЛК та відповідь ПЛК:

ПЕОМ-ПЛК

ЗАП мол.інф байт ст.інф. байт мол.байт адреси ст.байт адреси

ПЛК-ПЕОМ

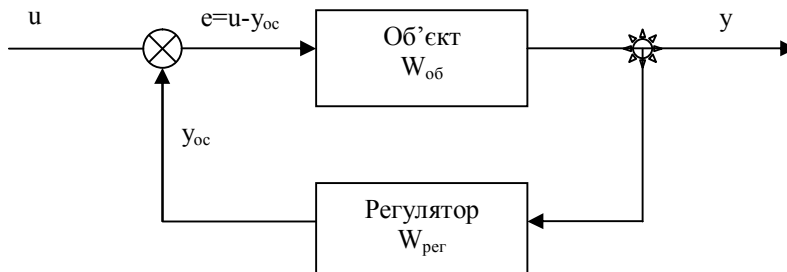
ЗАП мол.байт адреси ст.байт адреси

Текст драйвера ПК для ПЛК РК 5100 наведений в Додатку 1.

1. МОДЕЛЮВАННЯ ОДНОКОНТУРНОЇ САР

3.1 Послідовність перетворювання диференційного рівняння об'єкта в різницеве рівняння

Модель замкненої системи автоматичного регулювання:



$$Y(p) = W_{об}(p) * [U(p) - Y_{oc}(p)] = W_{об}(p) * [U(p) - W_{рег}(p) * Y(p)]$$

$$[1 + W_{об}(p) * W_{рег}(p)] * Y(p) = W_{об}(p) * U(p)$$

$$W(p) = \frac{W_{об}(p)}{1 + W_{об}(p) * W_{рег}(p)}$$

Об'єкт – аперіодична ланка з запізненням.

Регулятор – ІІІ закон регулювання.

$$W_{об}(p) = \frac{K_{об}}{T_{об} \cdot p + 1} \cdot \ell^{-p\tau_{об}} \quad W_{рег}(p) = K_p \cdot \left(1 + \frac{1}{T_u \cdot p}\right)$$

$$\begin{aligned} W(p) = \frac{y}{u} &= \frac{\frac{K_{об}}{T_{об} \cdot p + 1} \cdot \ell^{-p\tau_{об}}}{1 + \frac{K_{об}}{T_{об} \cdot p + 1} \cdot \ell^{-p\tau_{об}} \cdot K_p \cdot \left(1 + \frac{1}{T_u \cdot p}\right)} = \frac{K_{об} \cdot \ell^{-p\tau_{об}}}{T_{об} \cdot p + K_{об} \cdot \ell^{-p\tau_{об}} \cdot K_p \cdot \left(1 + \frac{1}{T_u \cdot p}\right)} = \\ &= \frac{K_{об} \cdot \ell^{-p\tau_{об}}}{T_{об} \cdot p + 1 + K_{об} \cdot \ell^{-p\tau_{об}} \cdot K_p \cdot \frac{T_u \cdot p + 1}{T_u \cdot p}} = \frac{K_{об} \cdot \ell^{-p\tau_{об}} \cdot T_u \cdot p}{T_{об} \cdot p \cdot T_u \cdot p + T_u \cdot p + K_{об} \cdot \ell^{-p\tau_{об}} \cdot K_p \cdot (T_u \cdot p + 1)} = \\ &= \frac{K_{об} \cdot \ell^{-p\tau_{об}} \cdot T_u \cdot p}{T_{об} \cdot T_u \cdot p^2 + T_u \cdot p + K_{об} \cdot K_p \cdot \ell^{-p\tau_{об}} \cdot T_u \cdot p + K_{об} \cdot K_p \cdot \ell^{-p\tau_{об}}} = \\ &= \frac{K_{об} \cdot \ell^{-p\tau_{об}} \cdot T_u \cdot p}{T_{об} \cdot T_u \cdot p^2 + T_u \cdot p \cdot (1 + K_{об} \cdot K_p \cdot \ell^{-p\tau_{об}}) + K_{об} \cdot K_p \cdot \ell^{-p\tau_{об}}} = \end{aligned}$$

Зворотнє перетворення Лапласа:

$$K_{o\bar{o}} \cdot T_u \cdot \dot{u} \cdot (t - \tau_{o\bar{o}}) = T_{o\bar{o}} \cdot T_u \cdot \ddot{y} + T_u \cdot \dot{y} + T_u \cdot K_{o\bar{o}} \cdot K_p \cdot \dot{y} \cdot (t - \tau_{o\bar{o}}) + K_{o\bar{o}} \cdot K_p \cdot y \cdot (t - \tau_{o\bar{o}}) =$$

$$y \cdot (t - \tau_{o\bar{o}}) = \bar{y}; u \cdot (t - \tau_{o\bar{o}}) = \bar{u}$$

$$T_{o\bar{o}} \cdot D^2 \cdot y + D \cdot y + K_{o\bar{o}} \cdot K_p \cdot D \cdot y + K_{o\bar{o}} \cdot \frac{K_p}{T_u} \cdot \bar{y} = K_{o\bar{o}} \cdot D \cdot \bar{u}$$

Метод Z-перетворення для отримання рекурентного співвідношення:

$$z(T_{o\bar{o}} + I)y + z \left(K_{o\bar{o}} \cdot K_p \cdot I + K_{o\bar{o}} \cdot \frac{K_p}{T_u} \cdot I^2 \right) \cdot \bar{y} = z(K_{o\bar{o}} \cdot I) \cdot \bar{u}$$

$$y(z) \left(T_{o\bar{o}} + \frac{\tau}{z-1} \right) + \bar{y}(z) \left(K_{o\bar{o}} \cdot K_p \cdot \frac{\tau}{z-1} + K_{o\bar{o}} \cdot \frac{K_p}{T_u} \cdot \frac{\tau^2}{(z-1)^2} \right) \cdot \bar{y} = \bar{u}(z) \left(K_{o\bar{o}} \cdot \frac{\tau}{z-1} \right)$$

$$y(z)(T_{o\bar{o}}z^2 - 2T_{o\bar{o}}z + T_{o\bar{o}} + \tau - \tau) + \bar{y}(z) \left(K_{o\bar{o}} \cdot K_p \cdot \tau - K_{o\bar{o}} \cdot K_p \cdot \tau + K_{o\bar{o}} \cdot \frac{K_p}{T_u} \cdot \tau^2 \right) \cdot \bar{y} = \bar{u}(z)(K_{o\bar{o}} \cdot \tau - K_{o\bar{o}} \cdot \tau)$$

Поділимо це на z^2 :

$$y(z)(T_{o\bar{o}} - 2T_{o\bar{o}}z^{-1} + T_{o\bar{o}}z^{-2} + \tau^{-1} - \tau^{-2}) = y(z)(T_{o\bar{o}} + (\tau - 2T_{o\bar{o}})z^{-1} + (T_{o\bar{o}} - \tau)z^{-2})$$

$$\bar{y}(z) \left(K_{o\bar{o}} \cdot K_p \cdot \tau^{-1} - K_{o\bar{o}} \cdot K_p \cdot \tau^{-2} + K_{o\bar{o}} \cdot \frac{K_p}{T_u} \cdot \tau^2 z^{-2} \right) = \bar{y}(z) \left(K_{o\bar{o}} \cdot K_p \cdot \tau^{-1} + \left(K_{o\bar{o}} \cdot \frac{K_p}{T_u} \cdot \tau^2 - K_{o\bar{o}} \cdot K_p \cdot \tau \right) z^{-2} \right)$$

$$\bar{u}(z)(K_{o\bar{o}} \cdot \tau^{-1} - K_{o\bar{o}} \cdot \tau^{-2})$$

Нехай параметри об'єкту такі: $T_{o\bar{o}}=246.4$ с, $K_{o\bar{o}}=2.5$ %РО/од.вх.вел. , $\tau_{o\bar{o}}=15.4$ с.

Враховуючи , що $y=y_n$; $yz^{-1}=y_{n-1}$; $yz^{-2}=y_{n-2}$ та обравши період дискретизації $\tau=3$ с ,отримаємо $\tau_{o\bar{o}}=15.4$ с= 3τ

$$\bar{y} = y \cdot (t - \tau_{o\bar{o}}) = y_{n-3}; \bar{u} = u \cdot (t - \tau_{o\bar{o}}) = u_{n-3}$$

Змінюючи період дискретизації, будемо отримувати рівняння з різною глибиною обліку попереднього стану системи. При зменшенні періода дискретизації зростає кількість аналізованих попередніх станів системи, підвищується точність регулювання (моделі САР), але збільшується програма і відповідно збільшується час розрахунку.

Рівняння отримує вигляд:

$$y_n = \frac{1}{246} \left[487y_{n-1} - 241y_{n-2} - 12.5K_p y_{n-4} + 12.5K_p y_{n-5} - 62.5 \frac{K_p y_{n-5}}{T_u} + 12.5U_{n-4} - 12.5U_{n-5} \right]$$

3.2 Послідовність виконання лабораторної роботи з використанням ПЗ “Транскан”

Отримане рівняння треба запрограмувати на мові програмування ПЛК РК 5100. Арифметичні, логічні операції, структурування програми та деякі алгоритми реалізації фрагментів програми приведені у підрозділі 2.3.

Нижче дано лістинг програми, що реалізує замкнену САР на МСК ПЛК РК 5100 для отриманого вище рівняння:

```
;LABA.lst
$ТИП : РК5100
$ОЗУ : 8
$ВВ1 : _____
$ВВ2 : _____
$ВВ3 : _____
$ВВ4 : _____
$НРП : 10020
;
;АДРЕС = КОНСТАНТА
/10000 = 000000
/10001 = 000000
/10002 = 000000
/10003 = 000000
/10004 = 000000
/10005 = 000000
/10006 = 000000
/10007 = 000000
/10010 = 020020
/10011 = 000000
/10012 = 000000
/10013 = 000000
/10014 = 000000
/10015 = 000000
/10016 = 000000
/10017 = 000000
; N INSTR ОПЕРАНД
 0 НСТ 00
 1 НВЛ 000
 2 СВС 200 ; Un
 3 СВС 201 ; Yn
 4 СВС 221 ; timer
 5 СВС 250 ; Yn-1
 6 СВС 251 ; Yn-2
 7 СВС 252 ; Yn-3
 8 СВС 253 ; Yn-4
 9 СВС 254 ; Yn-5
10 СВС 255 ; Un-1
11 СВС 256 ; Un-2
12 СВС 257 ; Un-3
13 СВС 260 ; Un-4
14 СВС 261 ; Un-5
15 СВС 262 ; 2
16 КВС 000
17 КВМ 002
18 ЗПС 262

19 ПВЛ 001
20 НВЛ 001
; ТАЙМЕР НА 5 С
21 ТВС 221
```

22	ПРК	0005
23	ПБЛ	002
24	НБЛ	002

; СДВИГ ЗНАЧЕНИЙ $Y_n...$, $Un...$

25	ЧТС	253
26	ЗПС	254
27	ЧТС	252
28	ЗПС	253
29	ЧТС	251
30	ЗПС	252
31	ЧТС	250
32	ЗПС	251
33	ЧТС	201
34	ЗПС	250
35	ЧТС	260
36	ЗПС	261
37	ЧТС	257
38	ЗПС	260
39	ЧТС	256
40	ЗПС	257
41	ЧТС	255
42	ЗПС	256
43	ЧТС	200
44	ЗПС	255
45	СВС	201

; +487* Y_{n-1}

46	КВС	000
47	КВМ	363
48	УМН	250
49	УМН	262
50	ЗПС	201

; -12,5* $Un-5$

51	КВС	000
52	КВМ	031
53	УМН	261
54	ДЕЛ	262
55	ЗПС	230
56	ЧТС	201
57	ВАС	230
58	ЗПС	201

; +12,5* $Un-4$

59	КВС	000
60	КВМ	031
61	УМН	260
62	ДЕЛ	262
63	САС	201
64	ЗПС	201

; -241* Y_{n-2}

65	КВС	000
66	КВМ	361
67	УМН	251
68	ЗПС	230
69	ЧТС	201
70	ВАС	230
71	ЗПС	201

; +12,5* K_p*Y_{n-5}

```

72 КВС 000
73 КБМ 031
74 УМН 202
75 ДЕЛ 262
76 УМН 254
77 САС 201
78 ЗПС 201

;-12,5*Кр*Yn-4
79 КВС 000
80 КБМ 031
81 УМН 202
82 ДЕЛ 262
83 УМН 253
84 ЗПС 230
85 ЧТС 201
86 ВАС 230
87 ЗПС 201

;-62,5*Кр*Yn-5/Ti
88 КВС 000
89 КБМ 175
90 УМН 202
91 ДЕЛ 262
92 УМН 254
93 ДЕЛ 203
94 ЗПС 230
95 ЧТС 201
96 ВАС 230
97 ЗПС 201

;[]/Тоб=Yn
98 КВС 000
99 КБМ 100
100 ЗПС 230
101 ЧТС 201
102 ДЕЛ 230 ;т.е на 25(24.6);
103 ЗПС 201

; ЦИКЛ НА timer
104 СВС 221
105 ПВЛ 001
106 КОН

```

Після написання програми у будь якому текстовому редакторі (laba.txt) потрібно запустити на виконання ПЗ ТранСКАН (ts.exe) та отримати об'єктний код програми (Перший пункт меню програми ТранСКАН: ИСХОДНЫЙ ФАЙЛ -> .LST, .RCS, *.OBJ) (laba.obj). Далі завантажити цей код до ПЛК (режим ОБМЕН С ПК -> ЗАПИСЬ В ПК С ДИСКА) та запустити на виконання (див. п.2.1).

Ввести початкові значення до пам'яті (Ti, Кр, U) та перевірити роботу програми можна в режимі індикації (режим ОБМЕН С ПК -> ИНДИКАЦИЯ).

При відсутності з'єднання з контролером перевірити роботу програми можна за допомогою програми-емулятора ПЛК.

На рисунках 1,2,3 наведено відеокадри вказаних пунктів меню ПЗ ТранСКАН.

Рис.1 Головне меню програми ТранСКАН

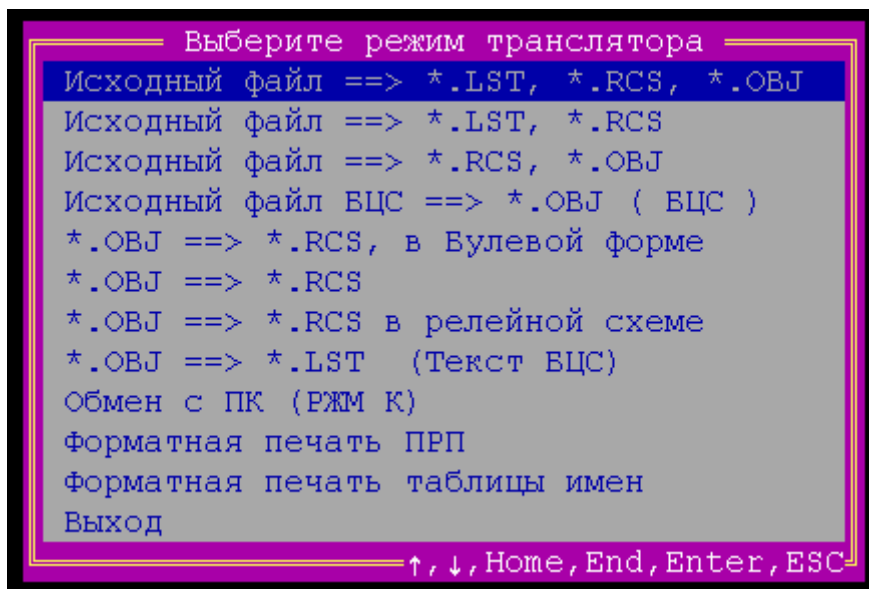


Рис.2 Режими обміну з ПК програми ТранСКАН

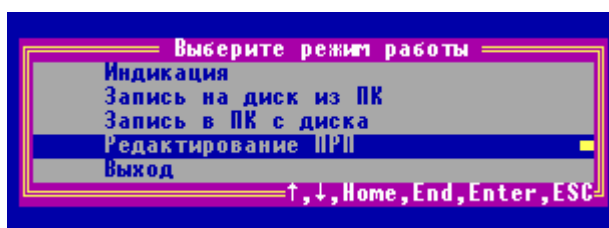


Рис.3 Эмулятор ПЛК із завантаженням об'єктним кодом програми замкненої САР



3.3 Послідовність виконання лабораторної роботи з використанням ПЗ “Скансот”

Проектування інтерфейсу програми моделювання замкненої САР виконується шляхом формування записів відповідного формату в БД пакету СканСОТ (рис.4).

Рис.4. Відеокадр БД пакету СканСОТ

Метка	Объект	Канал	Слово	Маска	Период	Тип
Лабораторная работа						
UN		1	000200o	000011111111111b	1	Приемник
YN		1	000201o	000011111111111b	1	Приемник
KP		1	000202o	000011111111111b	1	Приемник
TI		1	000203o	000011111111111b	1	Приемник
	Un	1	000200o			Пересылка произ
	Kp	1	000202o			Пересылка произ
	Ti	1	000203o			Пересылка произ
	Щит управления					Пассивное изобр
	System Output				1	График

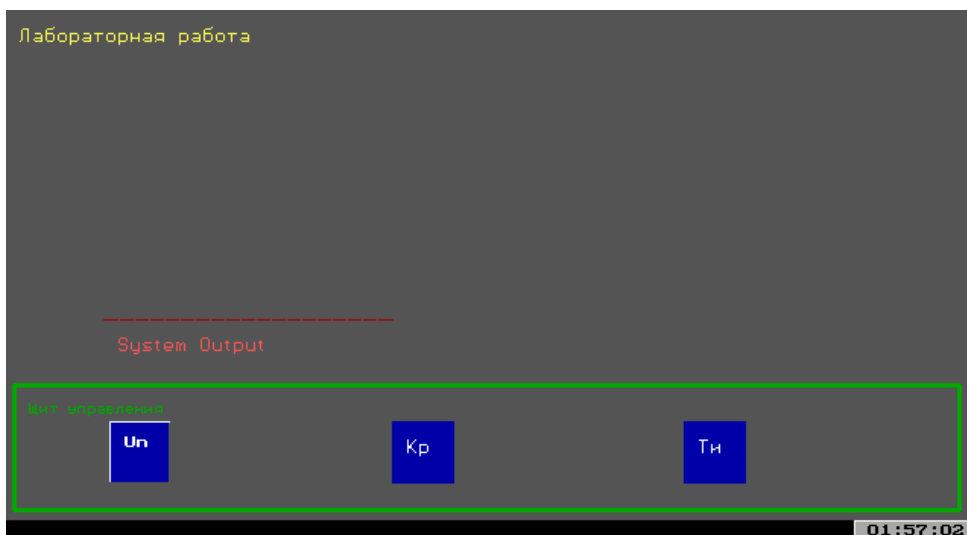
Всего записей: 10 Текущая запись: 1
 C:\ANDREY\INSTITUT\OATEP\SOТPRO\LABA.SSD
 F2 Запись F3 Удалить F4 Вставка F5 Копир. F6 Отобр. ← Редактор F10 Меню F9 Ошибки

В цьому прикладі створено такі об’єкти :

1. ОКНО “Лабораторная работа” – для рішення складних задач можливо створювати багатоконний інтерфейс.
2. ПРИЕМНИКИ UN,YN,KP,ТИ – потрібні для відображення клітин пам’яті ПЛК на екран програми.Маска визначає значущі біти даних.
3. YN-поточне значення виходного параметру САР.
4. Пересылка произвольного значения Un,Kp,Ti – ці елементи не відображаються на екрані, але до них прив’язуються функціональні клавіші (у прикладі F2,F3,F4) та заносяться значення параметрів регулятора та запускаючий вплив.
5. ПАССИВНОЕ ИЗОБРАЖЕНИЕ “Щит управления” – це статичний малюнок, потрібен для групування елементів на екрані.
6. ГРАФИК “SYSTEM OUTPUT” – прив’язаний до ПРИЕМНИКА YN та відображує у вигляді графіку зміни виходного параметру системи в залежності від часу (перехідний процес).
 Має такий параметр настройки як ПЕРИОД – інтервал відновлення інформації.

Отриманий результат моделювання представлено на рис.5.

Рис 5. Відеокадр перехідного процесу в одноконтурній САР



Текст драйверу ПК для ПЛК РК 5100

```

; Конфігурація
/10000 = 000000      ; Виходи
/10001 = 000000
/10002 = 000000
/10003 = 000000
/10004 = 000000      ; Входи
/10005 = 000000
/10006 = 000000
/10007 = 000000
/10010 = 020024      ; НРП
/10011 = 000000      ; УДАЛ
/10012 = 000000      ; ПРЕР
/10013 = 000000      ; ПРИОР
/10014 = 000000      ; Резерв
/10015 = 000000      ; Резерв
/10016 = 000062      ; Сигн. 100 мс
/10017 = 000000      ; КС
; Константи
/10020 = 000377      ; Байт ЗАПИСЬ
/10021 = 000376      ; Байт ЧТЕНИЕ
/10022 = 000317      ; Мах адреса запису до ТД
/10023 = 000260      ; Min адреса запису до ТД
; Службові слова ТД
; 00000      Слово стану МПВВ
; 00001      Контрольне слово МПВВ
; 00002      Лічильник файлу
; 00010      Байт ЗАПИСЬ
; 00011      Байт ЧТЕНИЕ
; 00012      Таймер
; 00100      Мах адреса запису до ТД
; 00101      Min адреса запису до ТД
; 00102      Проміжне слово
;             лічильника файлу
; Примітка. Сигнальний байт ГОТОВ=375
; формується в тілі програми

0000  НСТ  00      ; Текст ПрП-початок програми
      НБЛ  000     ; Блок запуску

      СБС  000     ; Обнулення службових слів в ТД
      СБС  001
      СБС  002
      СБС  010
      СБС  011
      СБС  012
      СБС  100
      СБС  101
      СБС  102

```

ПФФ 002 ; запис до ТД байтів ЗАПИСЬ, ЧТЕНИЕ
 УНС 020
 ПРС 0020
 ПРС 0010
 ПРК 0002
 ПРК 0002
 СБС 002
 ПФФ 002 ; запис до ТД max и min адрес запису до ТД
 УНС 020
 ПРС 0022
 ПРС 0100
 ПРК 0002
 ПРК 0002
 СБС 002
 ЧТР 000 ; Програмування МПВВ
 ЗПС 000
 -||- 00002 ; Якщо результат тестування
 КБМ 376 ; більш ніж нуль , тоді запис керуючого
 КБС 200 ; слова до МПВВ
 ЗПР 000

 0031 ПБЛ 001

 0032 НДЛ 001 ; Блок ситуаційного аналізу

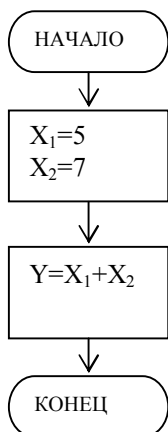
 ЧТР 000 ; Отримання байту
 ЗПС 000
 -||- 00000
 ЧТР 004
 ЗБМ 001
 -||- 00000 ; Ідентифікація байту ЗАПИСЬ
 РВН 001
 ПРС 0010
 ПБЛ 002 ; Перехід на блок запису в ТД
 -||- 00000 ; Ідентифікація байту ЧТЕНИЕ
 РВН 001
 ПРС 0011
 ПБЛ 003 ; Перехід на блок читання з ТД
 -||- 00000 ; Немає байтів ЗАПИСЬ
 ⊥ ; чи
 -| / |- 00000 ; ЧТЕНИЕ
 0049 ПБЛ 004 ; Перехід на блок режиму готовності
 0050 НБЛ 002 ; Блок запису слів до ТД
 ЧТР 000 ; Отримання байтів інформаційних та
 ЗПС 000 ; байтів адреси
 -||- 00000
 ЧТР 004
 ЗБМ 001
 ЧТР 000
 ЗПС 000
 -||- 00000

	ЧТР	004	
	ЗПС	001	
	ЧТР	000	
	ЗПС	000	
	- -	00000	
	ЧТР	004	
	ЗБМ	102	
	ЧТР	000	
	ЗПС	000	
	- -	00000	
	ЧТР	004	
	ЗБС	102	
	ЧТС	102	
	ЗПС	002	
	- -	00000	; Контроль отриманої адреси по діапазону
	ПРН	002	
	ПРС	0100	
	ПРС	0101	
	ЗСФ	002	; Запис слова до ТД
	ПРС	0000	
	ПРС	0001	
	ПРК	4095	
	ЧБМ	010	;запис байту ЗАПИСЬ до МПВВ
	ЗПР	004	
	ЧБМ	102	;запис адреси до МПВВ
	ЗПР	004	
	ЧБС	102	
	ЗПР	004	
	ПБЛ	005	; Перехід на блок обнулення службових слів ТД
	- -	00000	; Адреса не в діапазоні
	└┐		; чи
	- / -	00000	; невірний прийом байту
0091	ПБЛ	004	; Перехід на блок режиму готовності
0092	НБЛ	003	; Блок читання з ТД
	ЧТР	000	; Прийом байтів адреси
	ЗПС	000	
	- -	00000	
	ЧТР	004	
	ЗБМ	102	
	ЧТР	000	
	ЗПС	000	
	- -	00000	
	ЧТР	004	
	ЗПС	102	
	ЧТС	102	
	ЗПС	002	
	ЧСФ	002	; Читання слова з ТД
	ПРС	0000	
	ПРС	0001	
	ПРК	4095	
	ЧБМ	011	; Запис байту ЧТЕНИЕ до МПВВ
	ЗПР	004	

	ЧБМ	001	; Запис інф-них байтів до МПВВ
	ЗПР	004	
	ЧБС	001	
	ЗПР	004	
	ЧБМ	102	; Запис адреси до МПВВ
	ЗПР	004	
	ЧБС	102	
	ЗПР	004	
	ПБЛ	005	; Перехід на блок обнулення службових слів ТД
	- / -	00000	; Не прийнят байт адреси
0121	ПБЛ	004	; Перехід на блок режиму готовності
0122	НБЛ	004	; Блок режиму готовності
	КБМ	375	; Запис байту ГОТОВ до МПВВ
	КБС	000	
	ЗПР	004	
0126	ПБЛ	005	; Перехід на блок обнулення службових слів ТД
0127	НБЛ	005	; Блок обнулення сл.слів ТД
	СБС	001	
	СБС	002	
	СБС	102	
0131	ПБЛ	001	
0132	КОН		; Возврат на початок ПрП

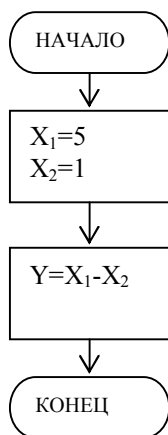
Текст типових програм для ПЛК РК 5100

ДОДАВАННЯ



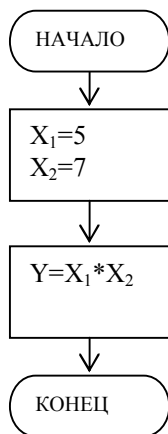
0	НСТ	00
1	НБЛ	000
2	КБС	000
3	КБМ	005
4	ЗПС	200
5	КБС	000
6	КБМ	007
7	САС	200
8	ЗПС	201
9	КОН	

ВІДНІМАННЯ



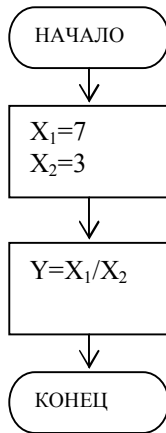
0	НСТ	00
1	НБЛ	000
2	КБС	000
3	КБМ	001
4	ЗПС	200
5	КБС	000
6	КБМ	005
7	ВАС	200
8	ЗПС	201
9	КОН	

МНОЖЕННЯ



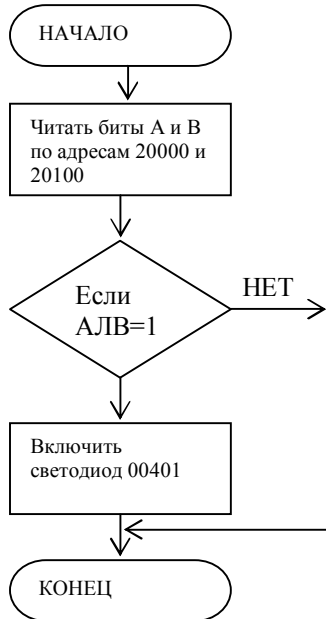
0	НСТ	00
1	НБЛ	000
2	КБС	000
3	КБМ	005
4	ЗПС	200
5	КБС	000
6	КБМ	007
7	УМН	200
8	ЗПС	201
9	КОН	

ДІЛЕННЯ



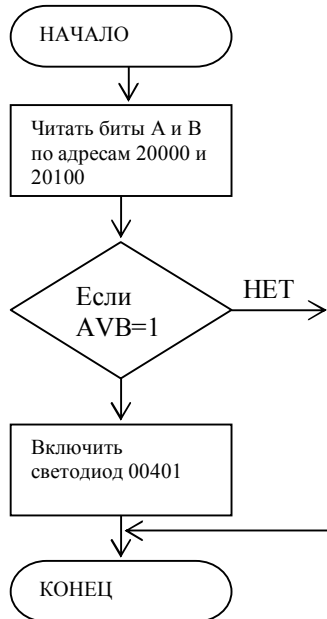
0	НСТ	00
1	НБЛ	000
2	КБС	000
3	КБМ	007
4	ЗПС	200
5	КБС	000
6	КБМ	003
7	ДЕЛ	200
8	ЗПС	201
9	КОН	

ЛОГІЧНА ФУНКЦІЯ AND



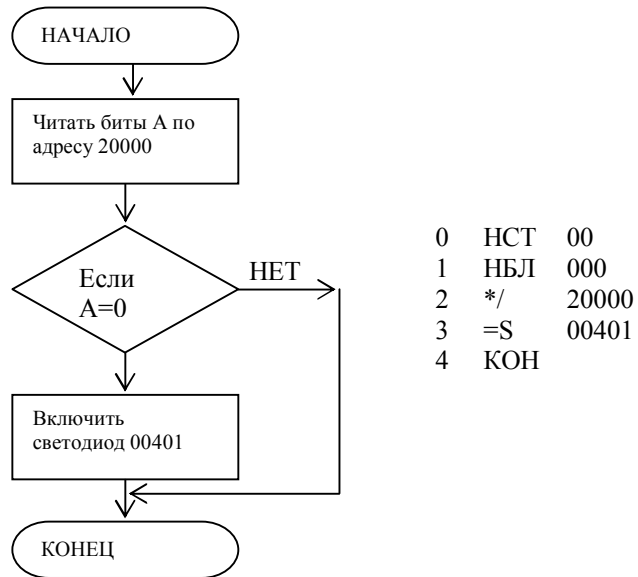
0	НСТ	00
1	НБЛ	000
2	*	20000
3	*	20100
4	=S00401	
5	КОН	

ЛОГІЧНА ФУНКЦІЯ OR

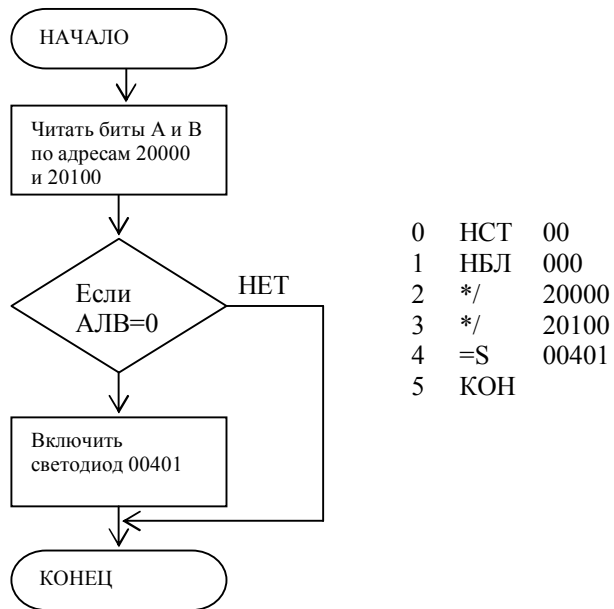


0	НСТ	00
1	НБЛ	000
2	(
3	*	20000
4	+	
5	*	20100
6)	
7	=S00401	
8	КОН	

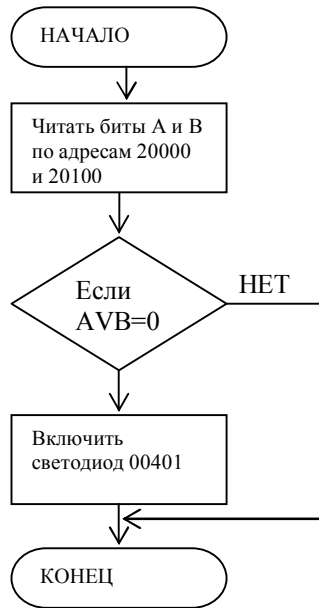
ЛОГІЧНА ФУНКЦІЯ NOT



ЛОГІЧНА ФУНКЦІЯ AND-NOT



ЛОГІЧНА ФУНКЦІЯ OR-NOT



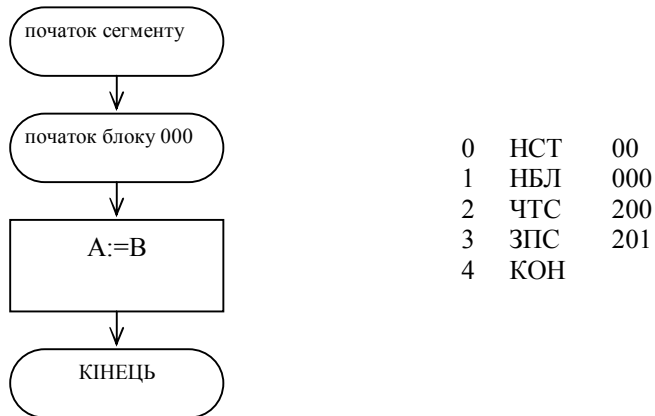
0	НСТ	00
1	НБЛ	000
2	(
3	*/	20000
4	+	
5	*/	20100
6)	
7	=S	00401
8	КОН	

ТАЙМЕР

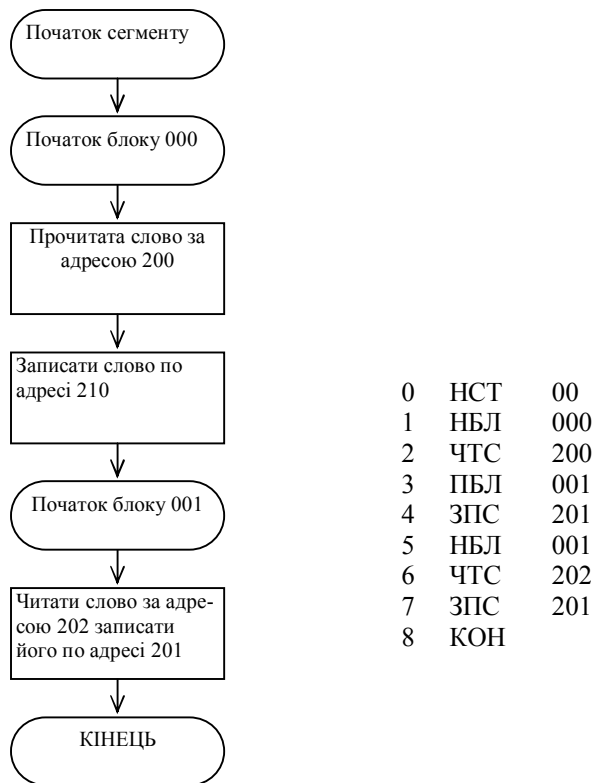


0	НСТ	00
1	НБЛ	000
2	СБС	200
3	-(R)-	00400
4	-(R)-	00401
5	=S	00400
6	ТВД	200
7	ПРК	0020
8	*	20016
9	=S	00401
10	КОН	

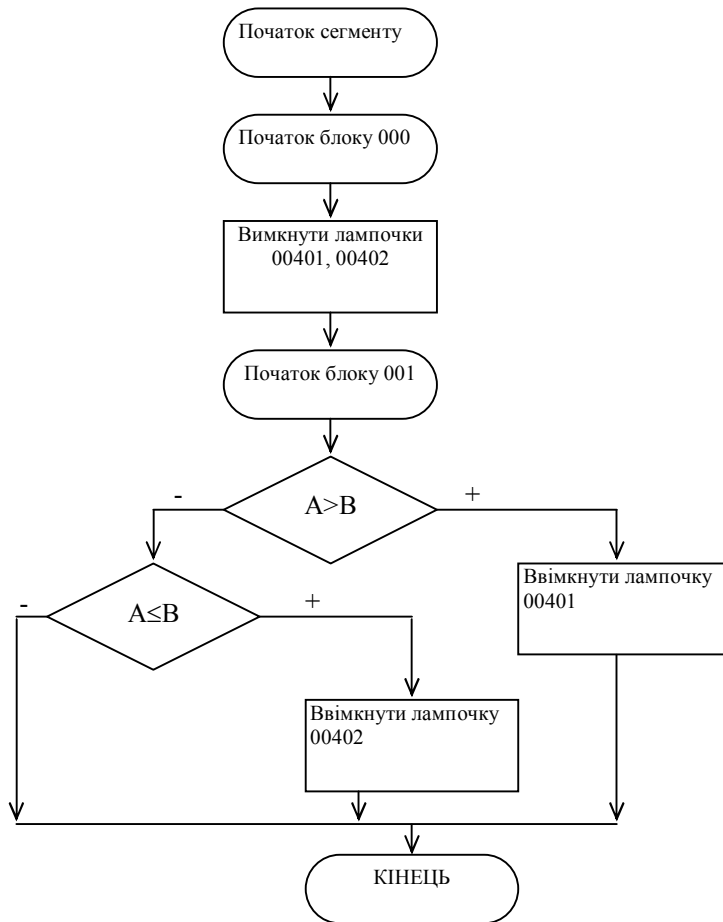
Програма, що реалізує конструкцію присвоєння засобами ЯСК ПЛК ФК5001



Програма, що реалізує конструкцію безумовного переходу з записом слова, яке знаходиться за однією адресою за другою адресою

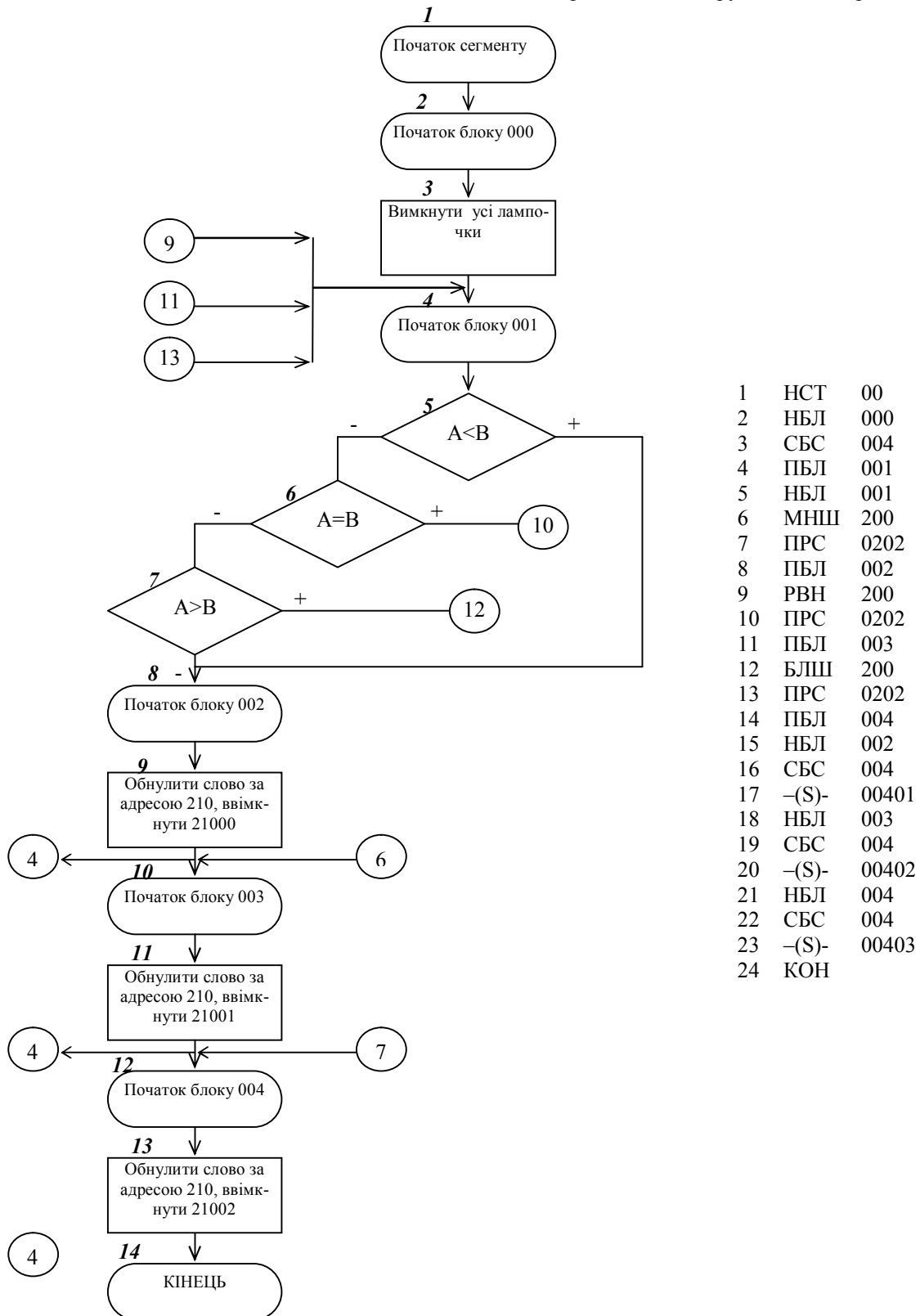


Програма, що реалізує конструкцію безумовного переходу. Умовою переходу взято порівняння двох слів А і В. Якщо $A > B$ то включити один індикатор, якщо $A \leq B$ – інший

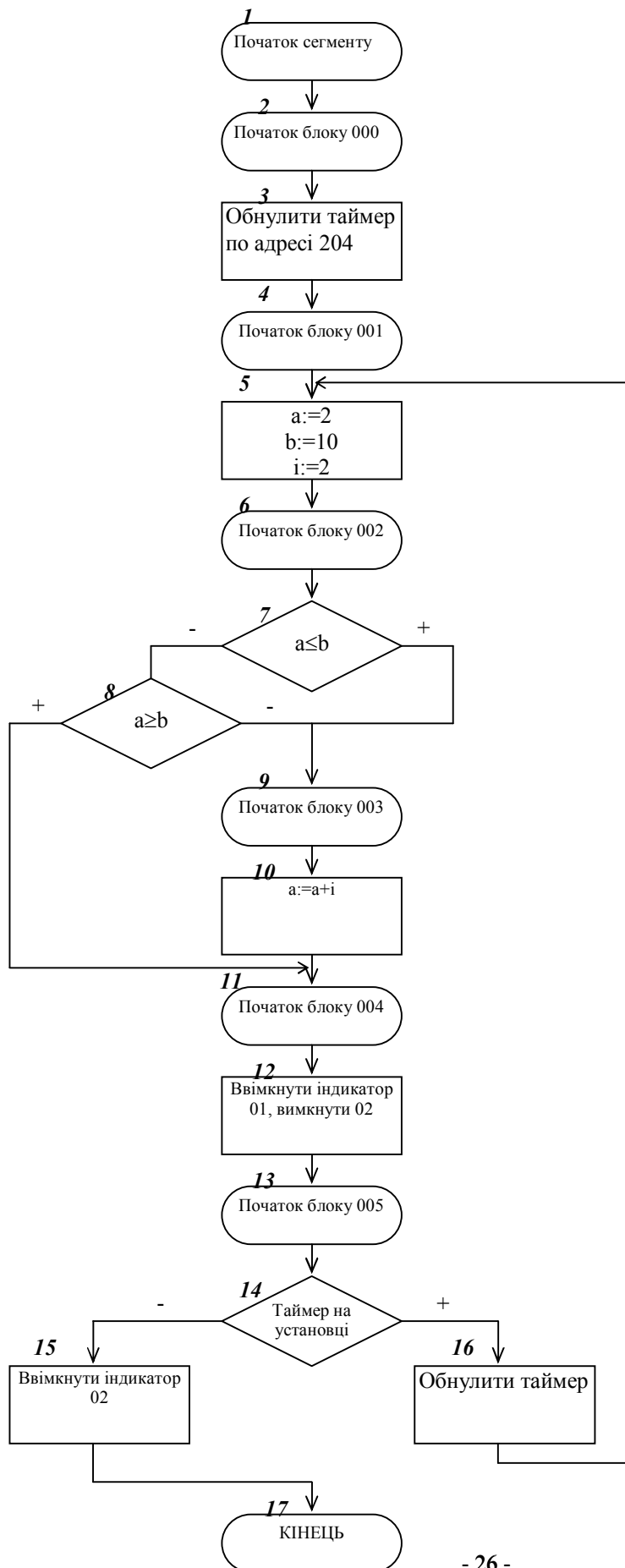


0	НСТ	00
1	НБЛ	000
2	-(R)-	00401
3	-(R)-	00402
4	ПБЛ	001
5	НБЛ	001
6	БЛШ	202
7	ПРС	0200
8	-(S)-	00401
9	МНШ	202
10	ПРС	0200
11		
12	РВН	202
13	ПРС	0200
14	-(S)-	00402
15	КОН	

Програма, що реалізує конструкцію безумовного переходу. Умовою переходу взято порівняння двох слів А і В. Якщо $A > B$ то включити один індикатор, якщо $A = B$ – другий, $A < B$ – третій

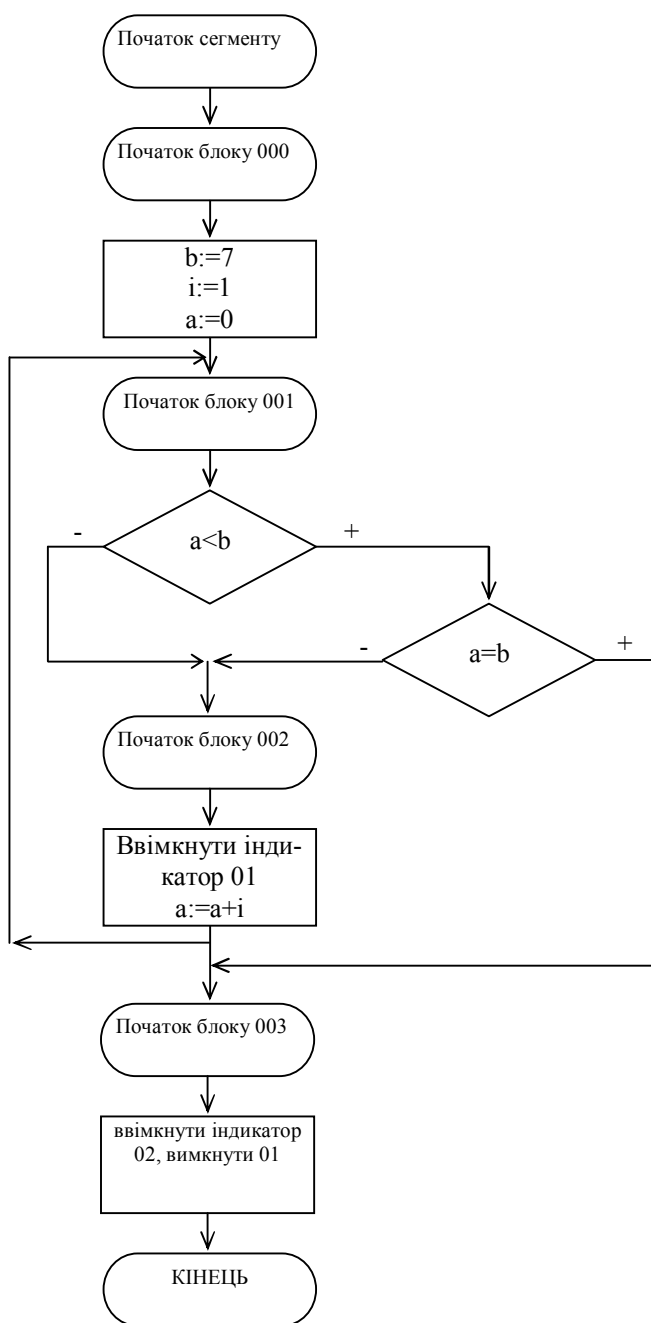


Програма, що реалізує конструкцію циклу. Збільшувати число А на якусь постійну і до заданих пір (число В), ввімкнувши індикатор. По досягненні А числа В погасити цей індикатор і ввімкнути інший. Поставити таймер на виконання циклу.



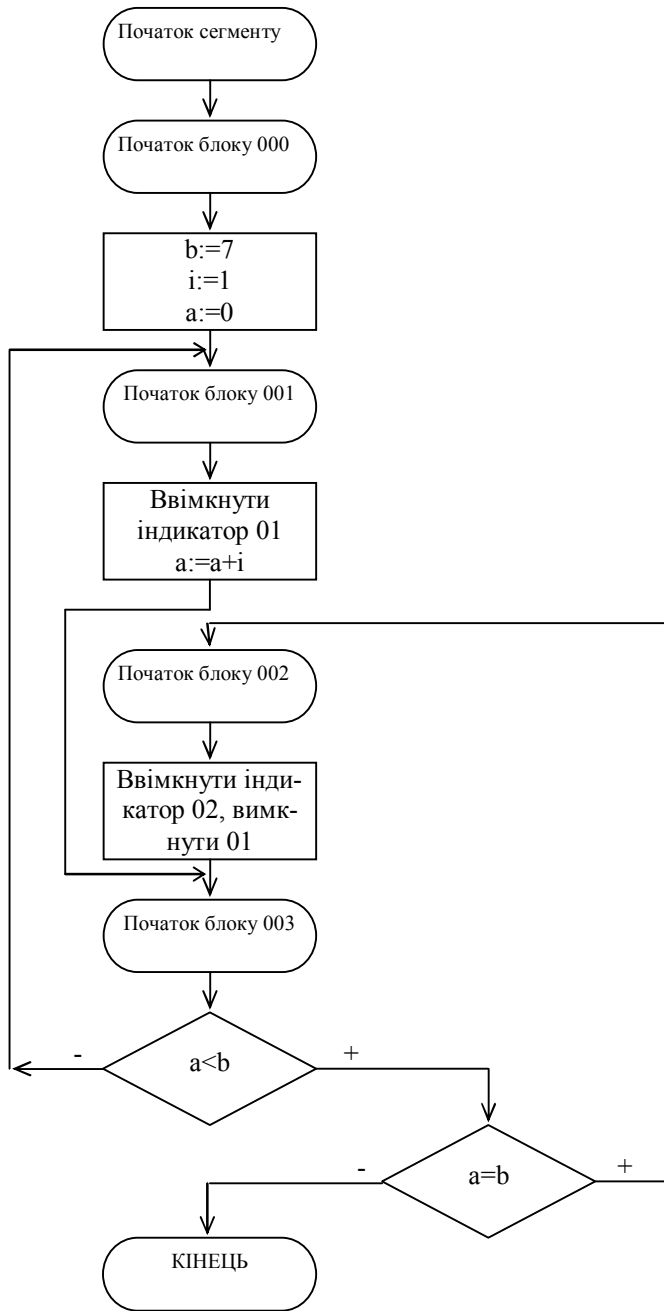
0	НСТ	00
1	НБЛ	000
2	СБС	004
3	СБС	204
4	ПБЛ	001
5	НБЛ	001
6	КБС	000
7	КБМ	002
8	ЗСП	200
9	КБС	000
10	КБМ	010
11	ЗПС	202
12	КБС	000
13	КБМ	002
14	ЗПС	203
15	ПБЛ	002
16	НБЛ	002
17	МНШ	200
18	ПРС	0202
19	ПБЛ	003
20	РВН	200
21	ПРС	0202
22	І	
23	БЛШ	200
24	ПРС	0202
25	ПБЛ	004
26	НБЛ	003
27	ЧТС	200
28	САС	203
29	ЗПС	200
30	ПБЛ	005
31	НБЛ	004
32	-(S)-	00401
33	-(R)-	00401
34	НБЛ	005
35	ТВД	204
36	ПРК	0050
37	- / -	20416
38	СБС	204
39	ПБЛ	002
40	- / -	20416
41	-(S)-	00402
42	КОН	

Програма, що реалізує цикл з передумовою з виходом по логічній «1». При виконанні циклу засвітити індикатор, після виконання – інший.



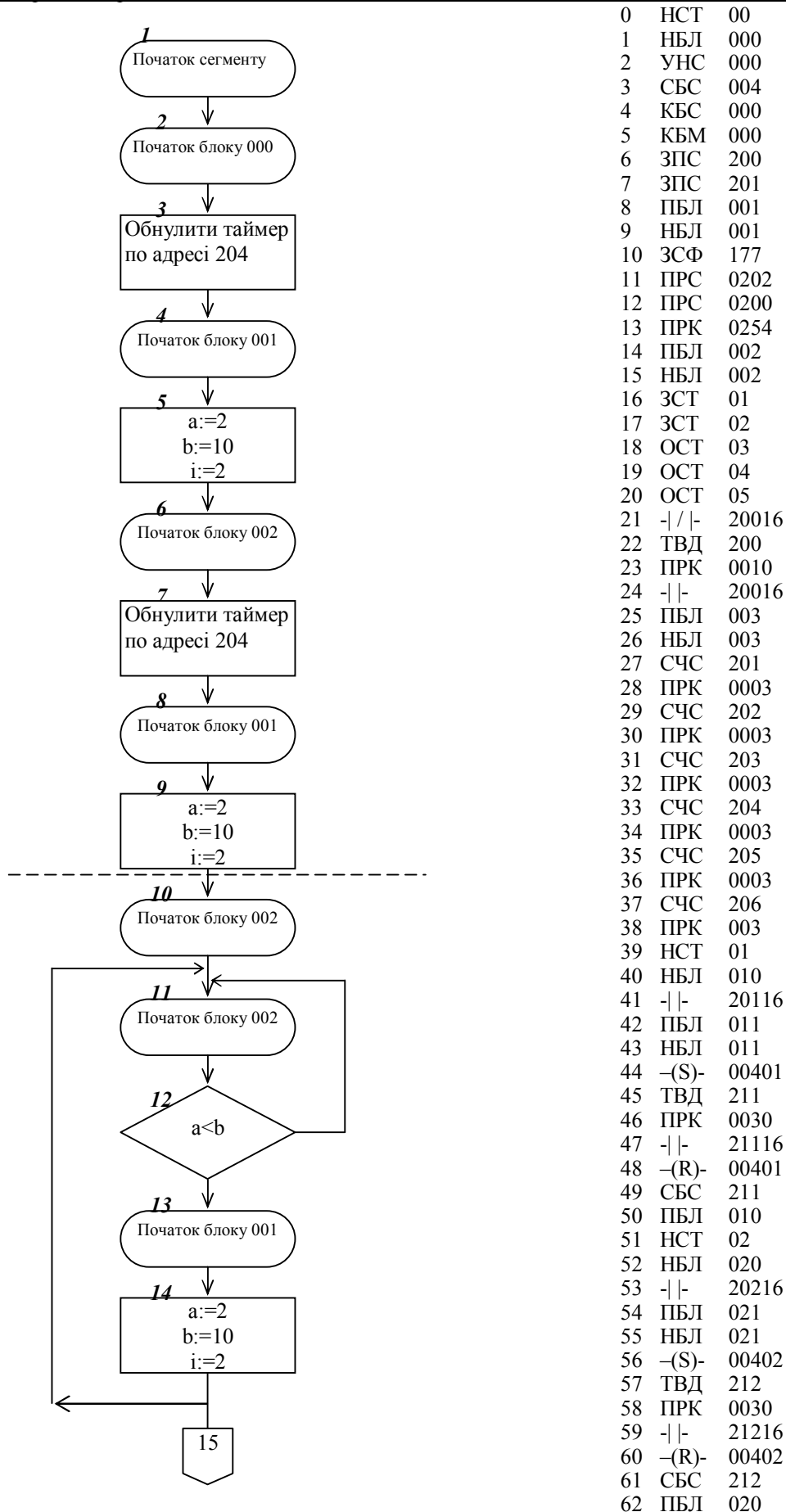
0	НСТ	00
1	НБЛ	000
2	СБС	213
3	СБС	210
4	КБС	000
5	КБМ	007
6	ЗПС	212
7	КБС	000
8	КБМ	001
9	ЗПС	211
10	ПБЛ	001
11	НБЛ	001
12	ТВС	213
13	ПРК	0003
14	- -	21316
15	МНШ	210
16	ПРС	0212
17	СБС	213
18	ПБЛ	002
19	- -	21316
20	РВН	210
21	ПРС	0212
22	СБС	213
23	ПБЛ	003
24	НБЛ	002
25	-(S)-	00401
26	ЧТС	210
27	САС	211
28	ЗПС	210
29	ПБЛ	001
30	НБЛ	003
31	-(S)-	00402
32	-(R)-	00401
33	КОН	

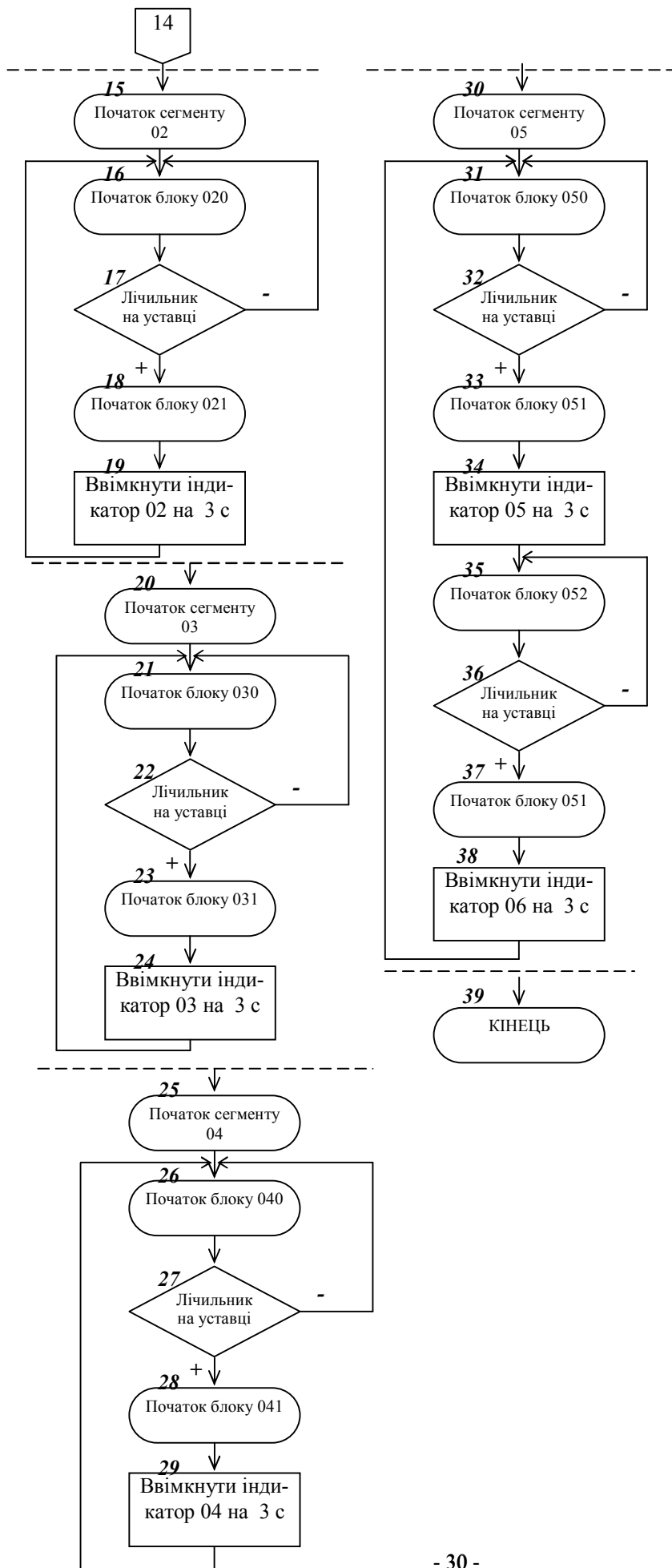
Програма, яка реалізує цикл з післямовою з виходом по логічній "1". Про закінчення циклу сповіщає засвітлюванням іншого індикатора.



0	НСТ	00
1	НБЛ	000
2	СБС	004
3	СБС	210
4	КБС	000
5	КБМ	001
6	ЗПС	211
7	КБС	000
8	КБМ	007
9	ЗПС	212
10	СБС	213
11	ПБЛ	001
12	НБЛ	001
13	-(S)-	00401
14	ЧТС	210
15	САС	211
16	ЗПС	210
17	ПБЛ	003
18	НБЛ	002
19	-(S)-	00402
20	-(R)-	00401
21	ПБЛ	003
22	НБЛ	003
23	ТВС	213
24	ПРК	0003
25	- -	21316
26	МНШ	210
27	ПРС	0212
28	СБС	213
29	ПБЛ	001
30	- -	21316
31	РВН	210
32	ПРС	0212
33	СБС	213
34	ПБЛ	002
35	КОН	

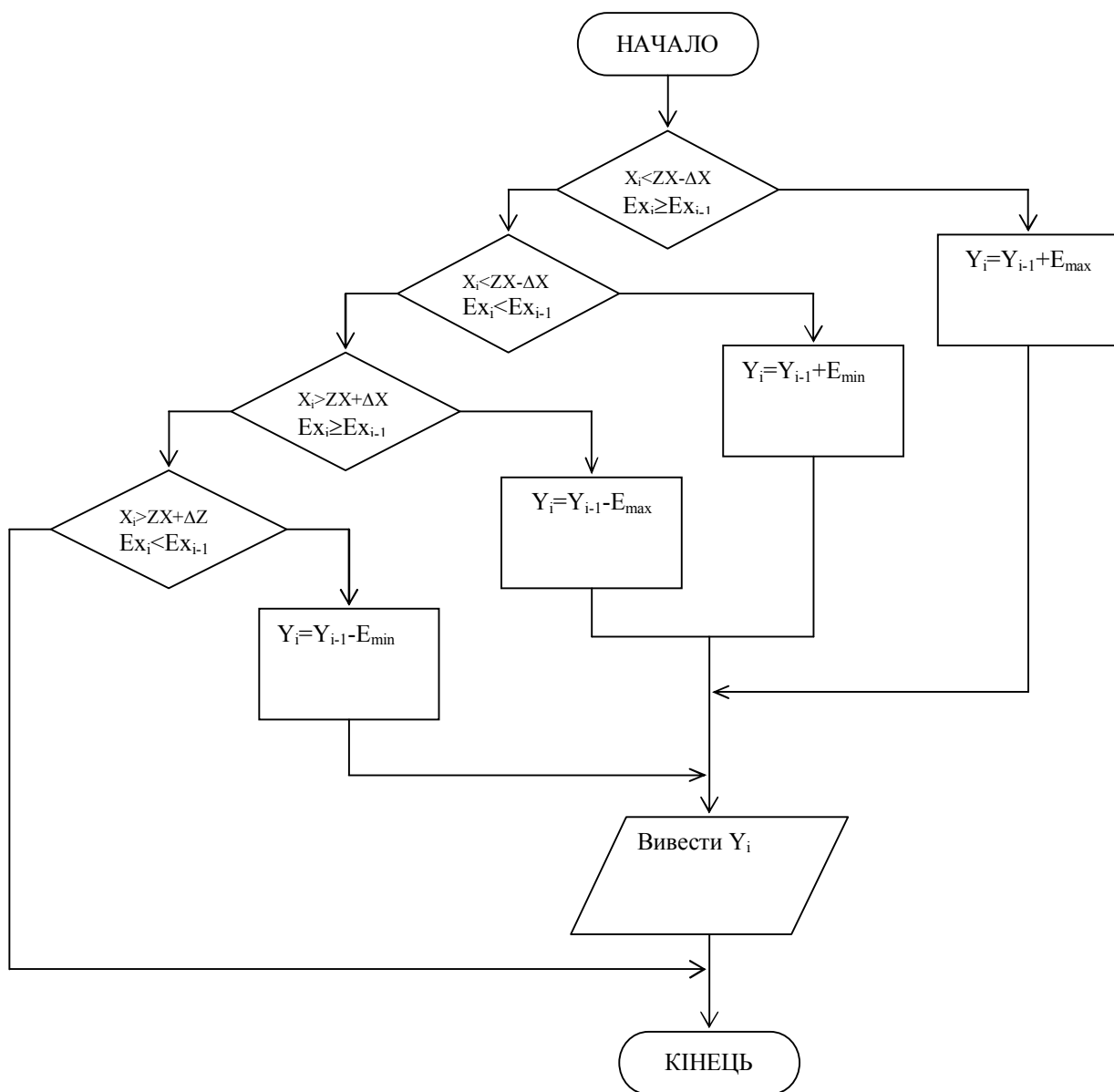
Програма, яка реалізує приклад структурування програм користувача ПЛК ФК5001. В ній 2 закритих сегменти, 3 відкритих, серед яких 4 сегменти з «завжди» активним блоком, один з двома активними блоками.





63	НСТ	03
64	НБЛ	030
65	-(S)-	00403
66	ТВД	213
67	ПРК	0030
68	- -	21316
69	-(R)-	00403
70	СБС	213
71	ПБЛ	030
72	НСТ	04
73	НБЛ	040
74	- -	20416
75	ПБЛ	041
76	НБЛ	041
77	-(S)-	00404
78	ТВД	214
79	ПРК	0030
80	- -	21416
81	-(R)-	00404
82	СБС	214
83	ПБЛ	040
84	НСТ	05
85	НБЛ	050
86	- -	20516
87	ПБЛ	051
88	НБЛ	051
89	-(S)-	00405
90	ТВД	215
91	ПРК	0030
92	- -	21516
93	-(R)-	00405
94	СБС	215
95	ПБЛ	052
96	НБЛ	052
97	- -	20616
98	ПБЛ	063
99	НБЛ	053
100	-(S)-	00406
101	ТВД	216
102	ПРК	0030
103	- -	21616
104	-(R)-	00406
105	СБС	216
106	ПБЛ	050
107	КОН	

Блок-схема нелінійного закону регулювання



Програма реалізації нелінійного закону регулювання

;N	ИНСТР	ОПЕРАНД	
	НСТ	00	
0		НБЛ	000
1		КБС	0000
2		КБМ	0012
3		ЗПС	265 E _{max} =10
4		КБС	0000
5		КБМ	0005
6		ЗПС	266 E _{min} =5
7		КБС	0000
8		КБМ	0000
9		ЗПС	275 Y _{min} =0
10		КБС	0000
11		КБМ	0120
12		ЗПС	276 Y _{max} =100
13		ПБЛ	001
14		НБЛ	001
15		ТВС	220
16		ПРК	0010
17		* 22016	
18		ПБЛ	002
19		НБЛ	002
20		ВПП	004
21		УНС	003
22		ЧТС	267
23		УНС	000 запис результатів роботи за
24		ЗПС	217 адресою 217
25		СБС	220
26		ПБЛ	001
27		НПП	004
28		УНС	003 Встановлюється сектор 3
29		*/	26710
30		КБМ	000
31		КБС	000 Y(i)=0
32		ЗПС	263
33		*	26710
34		ЧТС	263
35		ЗПС	263
36		*	27700 Якщо за адресою 27700 – 1, то вийти з ПП
37		ЧТС	261
38		ВАС	264
39		ЗПС	270 Записати за адресою 270 (zx-z)
40		ЧТС	261
41		САС	264
42		ЗПС	271 Записати за адресою 271 (zx+z)
43		МНШ	261 Якщо zx<z, то zx-z=0
44		ПРС	1664
45		КБМ	000
46		КБС	000
47		ЗПС	270
48		БЛШ	260 Якщо x(i)>zx, то E _x =x(i)-zx
49		ПРС	1661
50		ЧТС	260
51		ВАС	261
52		ЗПС	272
53		КБМ	000
54		КВС	000
55		ЗПС	273 E _x =zx-x(i)=0
56		ЧТС	272
57		САС	273
58		ЗПС	274 Записуємо E _x за адресою 274

59	МНШ	260	Якщо $x(i) < z_x$, то записуємо
60	ПРС	1661	за адресою 273 $E_x = z_x - x(i)$
61	ЧТС	261	
62	ВАС	260	
63	ЗРС	273	
64	КБМ	000	
65	КБС	000	
66	ЗПС	272	$E_x = 0$ за адресою 272
67	ЧТС	273	
68	САС	272	
69	ЗПС	274	Записати E_x за адресою 274
70	*	26707	
71	ПРН	260	
72	ПРС	1671	
73	ПРС	1670	
74	ЧТС	263	
75	ЗПС	263	
76	*/	26707	
77	ПРН	260	
78	ПРС	1671	
79	ПРС	1670	
80	ЧТС	263	
81	ЗПС	263	
82	=R	26700	Обнулити біти 00, 01, 03, 04
83	=R	26701	за адресою 267
84	=R	26703	
85	=R	26704	

Додаток 3

Зміст байту атрибутів при програмуванні АЦП.

16 однопровідних каналів (0-5)мА та (0-20)мА.

027 007
067 047
127 107
167 147
227 207
267 247
327 307
367 347

16 однопровідних каналів (4-20)мА.

025 005
065 045
125 105
165 145
225 205
265 245
325 305
365 345

16 однопровідних каналів (0-72)мВ та (0-10)В.

026 006	024 004
066 046	064 044
126 106	124 104
166 146	164 144
226 206	224 204
266 246	264 244
326 306	324 304
366 346	364 344

8 каналів з диференціальним включенням (0-5)мА та (0-20)мА

037 017
077 057
137 117
177 157

8 каналів з диференціальним включенням (4-20)мА

035 015
075 055
135 115
175 155

8 каналів з диференціальним включенням (0-72)мВ та (0-10)В

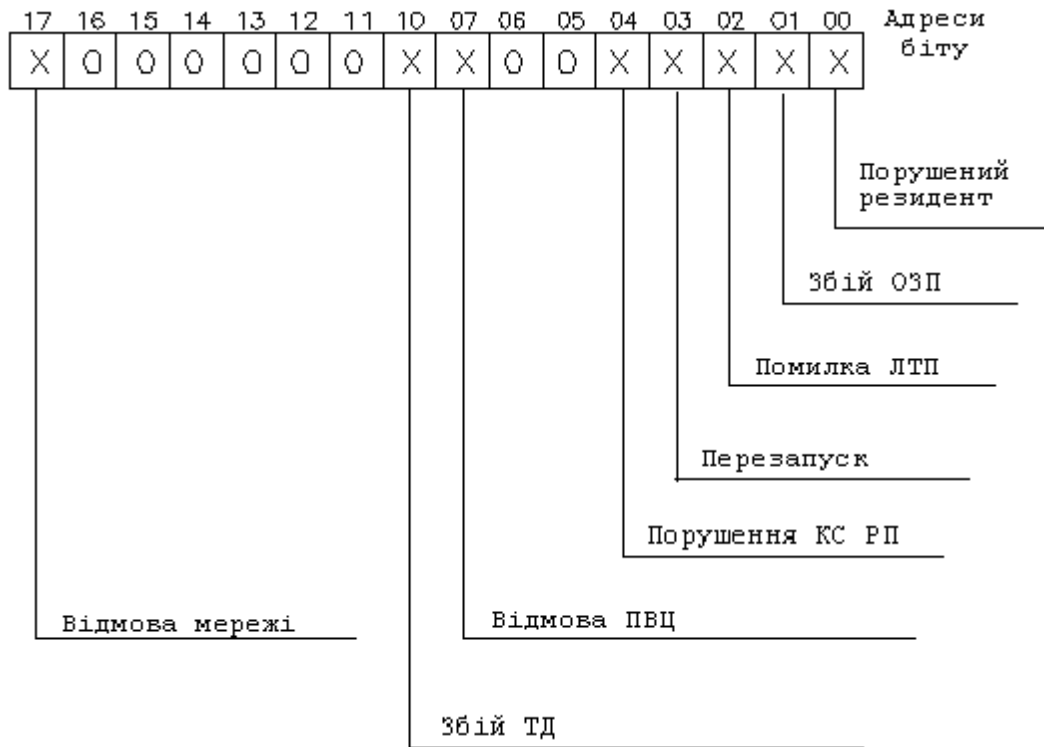
036 016	034 014
076 056	074 054
136 116	134 114
176 156	174 154

Зміст старшого байту АЦП

у випадку 16 каналів 037
у випадку 8 каналів 027

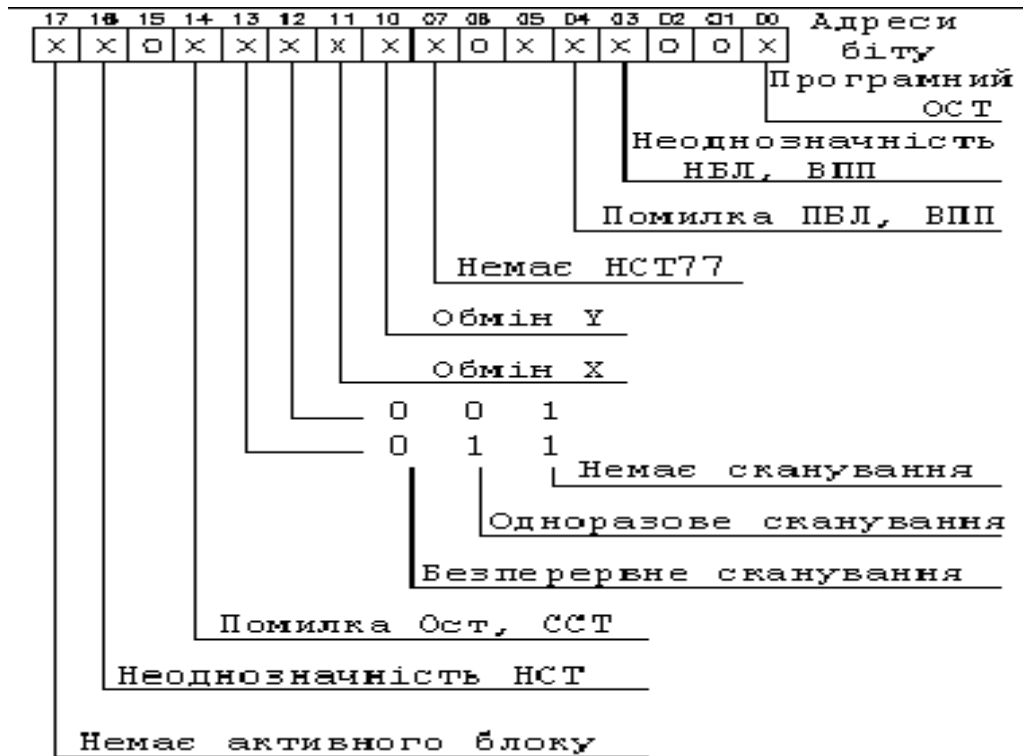
Системні відмови.

Системні відмови при "Початковому запуску".
Адреса слова 01772



Додаток 16 (продовження)

Системні відмови в "Інтерпритаторі".
Адреса слова 01773



Додаток 5

Початкові адреси пам'яті модулів та реєстрів.

Посадочне місце	Команда		
	ПФМ	ПМФ	ЧТР та ЗПР
A1	ПРС 1022	ПРС 1002	010
A2	ПРС 2022	ПРС 2002	020
A3	ПРС 3022	ПРС 3002	030
A4	ПРС 4022	ПРС 4002	040
A5	ПРС 5022	ПРС 5002	050
A6	ПРС 6022	ПРС 6002	060
A7	ПРС 7022	ПРС 7002	070

Примітка: кількість реєстрів - 7

Адреси звернення до реєстра модуля послідовного вводу-виводу при різних місцях встановлення на каркасі.

Посадочне Місце	Команди
	ЧТР та ЗПР
A0	000...004
A1	010...014
A2	020...024
A3	030...034
A4	040...044
A5	050...054
A6	060...064
A7	070...074

Примітка: кількість реєстрів - 4.