

Тернопільський національний технічний
університет імені Івана Пулюя

Кафедра автоматизації
технологічних процесів
і виробництв

Лабораторна робота № 8
з курсу
Проектування мікропроцесорних
систем керування технологічними
процесами

Виконання операцій з ЦАП та АЦП в
схемі програмного симулятора
EdSim51

Тернопіль 2023

Методичні вказівки для виконання лабораторної роботи № 8 «Виконання операцій з ЦАП та АЦП в схемі програмного симулятора EdSim51» з курсу «Проектування мікропроцесорних систем керування технологічними процесами»/Укл.: Медвідь В.Р., Пісьціо В.П. - Тернопіль ТНТУ, 2023 - 15 с.

Розглянуто і затверджено на засіданні кафедри автоматизації технологічних процесів і виробництв (протокол № 1 від 30.08.2023 року)

Лабораторна робота № 8

Виконання операцій з використанням ЦАП та АЦП в схемі програмного симулятора EdSim51

1. Цифро-аналогові перетворювачі

Цифро-аналогові перетворювачі (ЦАП) здійснюють перетворення цифрових кодів в аналоговий сигнал.

ЦАП можна розглядати як потенціометр із цифровим керуванням, який задає вихідний струм чи напругу як суму вхідних їх частин, що формують повну шкалу. Базова схема ЦАП складається із джерела опорної напруги, матриці двійково-вагових прецизійних резисторів та набору електронних ключів – комутаторів.

При замиканні кожного ключа сумарний струм на вході суматора струмів, функцію якого виконує операційний підсилювач (DA1), перетворюється у вихідну напругу.

На рис. 1,а наведено схему ЦАП із ваговою резистивною матрицею, а на рис. 1,б ЦАП із ланцюговою $R - 2R$ матрицею. Обидві схеми мають виходи за напругою.

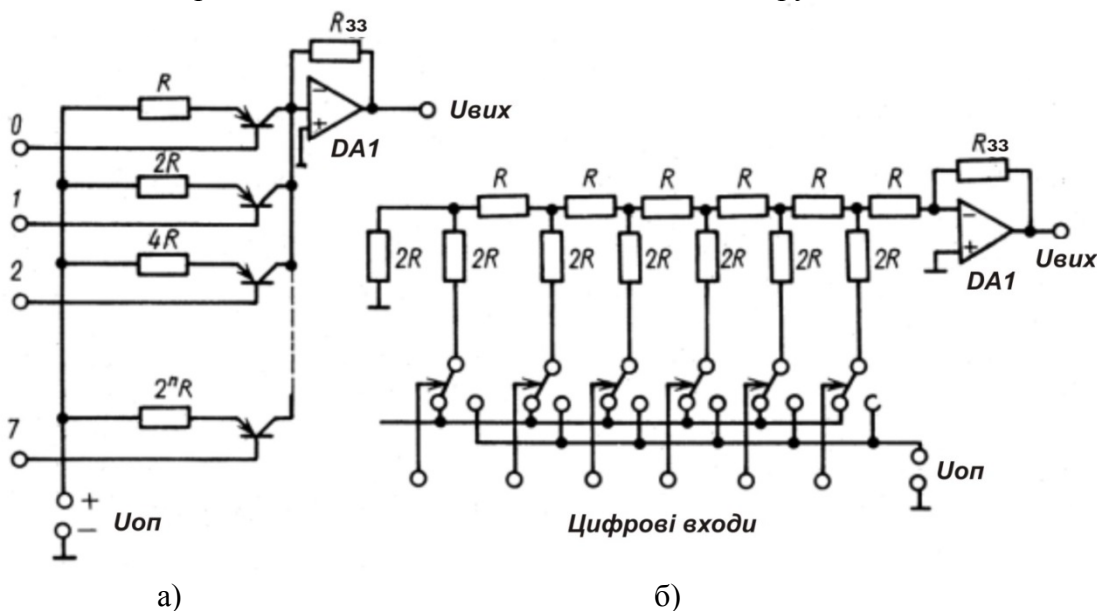


Рис. 1. Схеми ЦАП

Основною характеристикою ЦАП є роздільна здатність, яка визначається числом N розрядів ЦАП, що має N двійкові розряди. При цьому ЦАП повинен мати 2^N різних значень вихідного сигналу з роздільною здатністю $(2^N - 1) - 1$.

Абсолютне значення мінімального вихідного кванту визначається максимальним значенням $(2^N - 1)$ вхідного коду та максимальною вихідною напругою ЦАП (напруга шкали). Наприклад, при $N = 12$, $U_{шк} = 10$ В, абсолютна роздільна здатність ЦАП складає $10 / (2^N - 1) \text{ В} \approx 2,45$ мВ.

Використовуються ЦАПи в схемах мікропроцесорних вимірювальних приладів та систем.

2. Аналого-цифрові перетворювачі

Аналого-цифрові перетворювачі (АЦП) забезпечують перетворення вхідної аналогової напруги у цифровий код на виході, що є зручним для реєстрації та вводу в цифрові вимірювальні прилади (ЦВП).

Призначення вузлів АЦП наступне (рис. 2). У вхідному перетворювачі $Pr1$ аналогова величина перетворюється із одного виду в інший, наприклад здійснюється масштабування вхідного сигналу, перетворення напруги в інтервал часу, або змінної напруги, опору, ємності в постійну напругу.

Тут також може здійснюватися попередня дискретизація в часі. Перетворення аналог-код

здійснюється у перетворювачі *Пр2*. Якщо отриманий код є незручним, то у перетворювачі *Пр3* здійснюють перехід до нового коду, який є вихідним кодом для АЦП. Із нього кодовий сигнал поступає на пристрій індикації *ПІ*.

Узгоджену роботу АЦП забезпечують сигнали пристроїв управління АЦП *ПУ* та пристрій управління усього приладу *ПУП*.

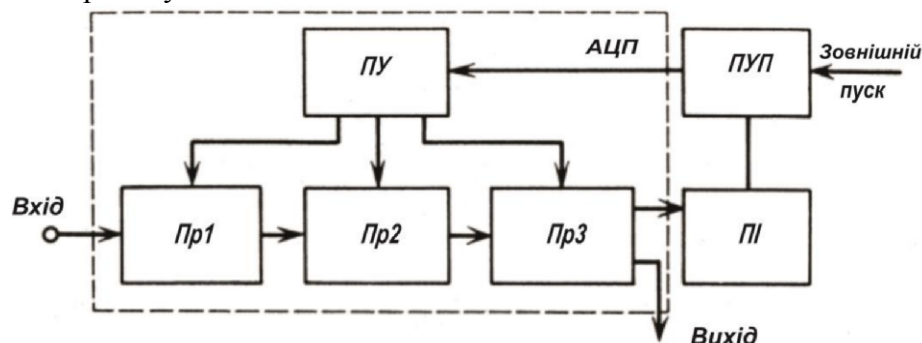


Рис. 2

За типом вимірюваних величин ЦВП діляться на вольтметри, частотоміри, фазометри, омметри і т. д. Крім того, їх ділять на групи за точністю, швидкістю, надійністю.

Залежно від способу організації процесу перетворення, усі ЦВП поділяються на прилади циклічні і слідкуючі.

У циклічних приладах весь процес перетворень протікає завжди незалежно від значення вимірюваної величини по заданій програмі від початку до кінця. Знімання показів допускаються лише під час частини повного такту вимірювання – часу індикації.

У слідкуючих ЦВП перехід до чергового перетворення здійснюється при зміні вхідного сигналу на певне прирощення. Прилади слідкуючої дії відносять до пристроїв адаптивної дискретизації, оскільки частоту перетворювань вибирають, виходячи із характеру сигналу.

Перетворення аналогової величини у цифровий код є вимірювальною процедурою і здійснюється шляхом виконання ряду операцій порівняння вимірюваної величини з набором дискретних еталонних величин, що мають однакову природу із перетворюваною. При цьому відбувається заміна неперервної величини на найближчу дискретну, утворену по певному алгоритму з допомогою міри, і зчитування коду, що відповідає цьому фіксованому значенню.

Одним з критеріїв, що лежать в основі класифікації методів аналого-цифрового (АЦ) – перетворення є характер АЦ–перетворення у часі. У відповідності із цим критерієм розрізняють методи послідовного підрахунку, метод порозрядного врівноваження, метод одночасного зчитування.

Метод послідовного підрахунку.

Його суть полягає у послідовному у часі порівнянні вимірюваної величини з відомою однорідною дискретною мірою x_0 .

Нехай вимірювана величина $x \in [x_{\min}, x_{\max}]$. На рис. 3,а зображено перший варіант методу послідовного підрахунку. При деякому числі n квантів x_0 спостерігається строга рівність $nx_0 = x^*$ або ж із деякою похибкою квантування ξ : $nx_0 + \xi = x^*$, де x^* – значення вимірюваної величини.

Якщо вибрати x_0 рівним одиниці вимірювання x , то число n буде одиничним кодом вимірювальної величини.

Похибка квантування $0 \leq \xi \leq x_0$.

Час аналого-цифрового перетворення обернено пропорційний точності перетворення.

Переваги: порівняна простота, мала статистична похибка, недолік – мала швидкодія.

Області застосування – цифрові вольтметри постійного струму.

Метод порозрядного врівноваження (порівняння).

Метод дозволяє отримати суттєвий вигравш у швидкодії (до 10^5 – 10^6 перетворень в секунду).

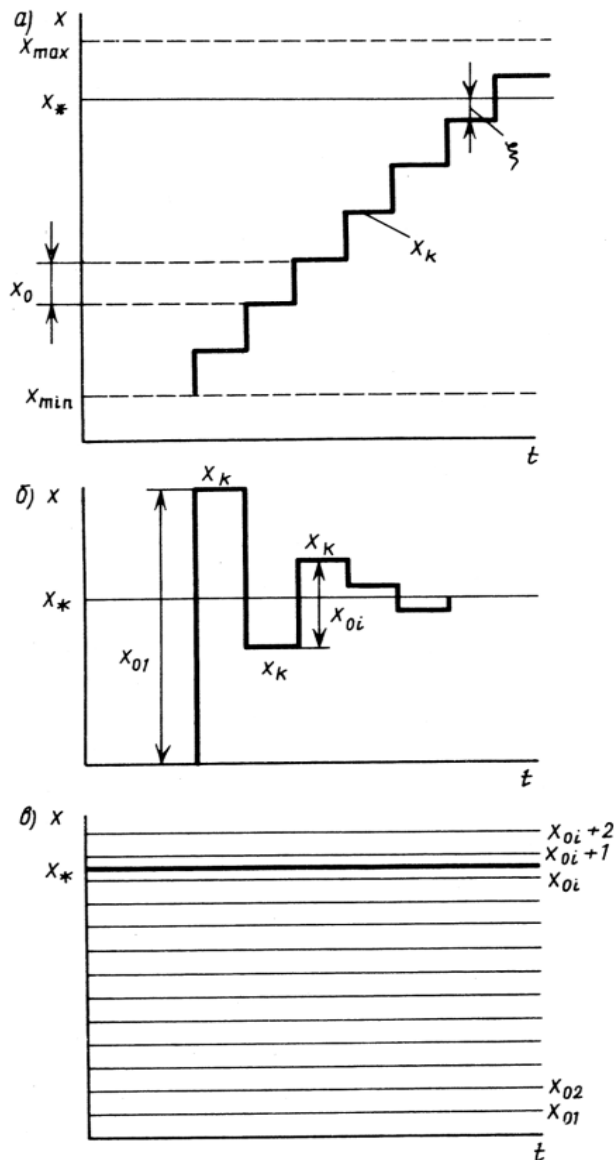


Рис. 3

Метод одночасного зчитування (порівняння).

При цьому методі реалізується взаємо-однозначна відповідність між множиною $\{x_{0i}\}$ квантів порівняння і очікуваною множиною дискретних значень вхідної неперервної величини x^*i . Вихідний код утворюється за номером найближчого значення x_{0i} . Можна значно прискорити вимірювання, якщо користуватися набором різновеликих квантів x_{0i} (кодоімпальсний метод).

Процес порівняння x^* та x_k (рис. 3,б), як і в попередньому методі, проходить послідовно у часі. Тут підключаються чи відключаються від процесу порівняння кванти x_{0i} , що по ефективності рівносильні деякому набору із k елементарних квантів x_0 . Це дозволяє кваліфікувати метод порозрядного врівноваження як паралельно-послідовний.

Таким чином “одночасність” методу означає паралельність включення усіх квантів x_{0i} у процесі порівняння. Отже, цей метод повністю паралельний. Він є найбільш швидкодіючим метод (100 – 200 МГц).

Гібридні методи – на основі сполучення методів зчитування і порозрядного врівноваження

Схема підключення АЦП та ЦАП програмного симулятора EdSim51 показана на рис. 4.

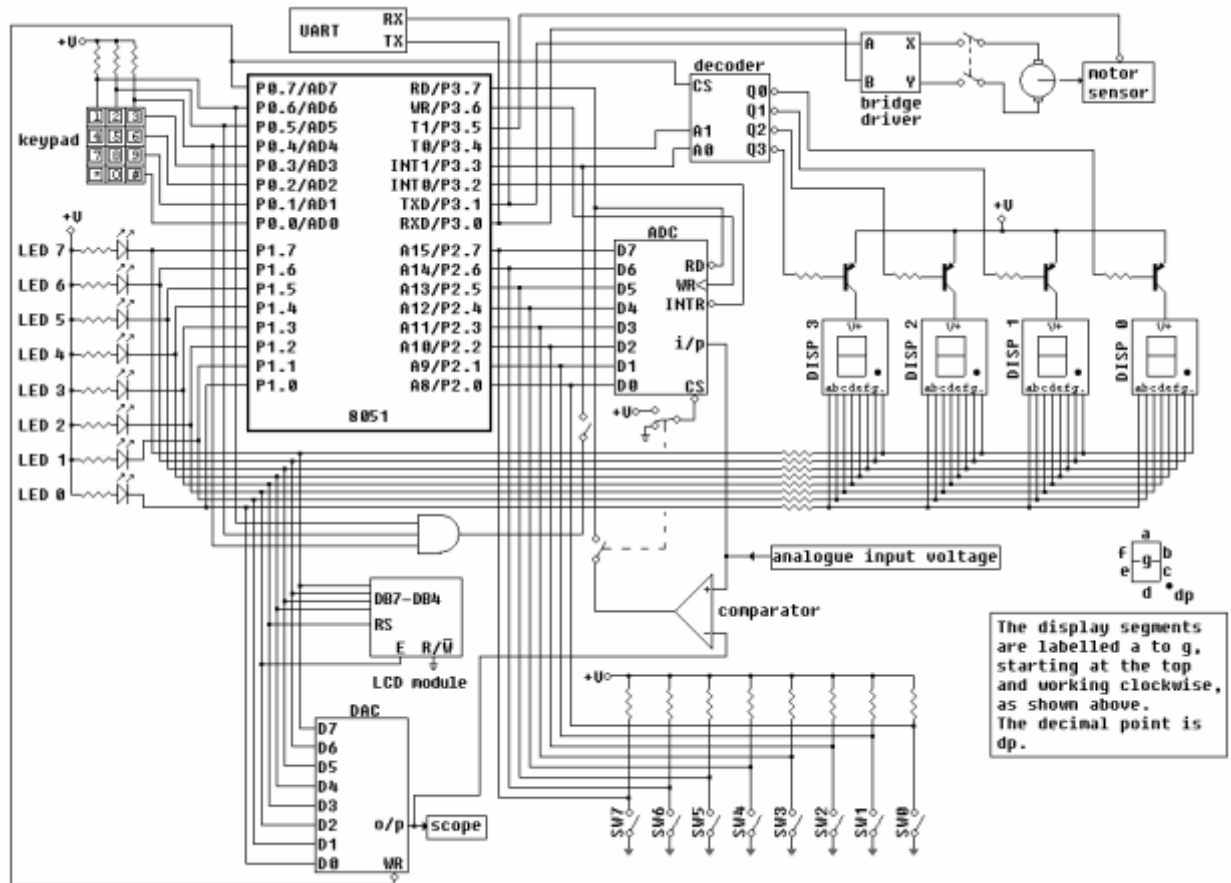


Рис. 4. Підключення АЦП (ADC) та ЦАП (DAC) до MCS51 в схемі симулятора EdSim51

3. Утворення пилкоподібного сигналу на виході ЦАП

Програма 1 генерує пилкоподібну напругу на виході ЦАП.

Програма 1

Програма забезпечує виведення пилкоподібного сигналу на «*DAC output on scope*» (рис. 5). Амплітуду та частоту пилкоподібного вихідного сигналу можна змінювати, задаючи в програмі величину константи const та крок квантування n. При значенні const=#0FFH на виході ЦАП отримуємо вихідний сигнал з максимальною амплітудою 5 В.

Час розгортки сигналу на «*DAC output on scope*» визначається частотою «*Update Freq.*». Мінімальний час розгортки забезпечується при частоті 1 Гц (рис. 5). Для роботи з програмою встановити значення «*Update Freq.*» рівним 1000 Гц.

```

const EQU 0EАН      ; встановити максимальне значення напруги на виході ЦАП
n EQU 04H           ; встановити значення кроку квантування
start: MOV R2,#00    ; завантажити початкове значення напруги в регістр R2
      CLR P0.7       ; включити лінію WR ЦАП
loop:  MOV A,R2       ; переслати вміст регістра R2 в акумулятор
      MOV P1, A      ; переслати дані з акумулятора на входи ЦАП (на P1)
      ADD A, #n      ; збільшити вміст акумулятора на крок квантування
      MOV R2,A       ; зберегти вміст акумулятора в регістрі R2
      SUBB A,#const  ; відняти від вмісту A значення максимальної амплітуди const
      JC loop        ; перейти за міткою loop, якщо вміст A менший за const (флажок
                    ; C=«1»)
      JMP start      ; якщо вміст A більший за const (флажок C=«0»)
                    ; перехід на початок програми за міткою start

```

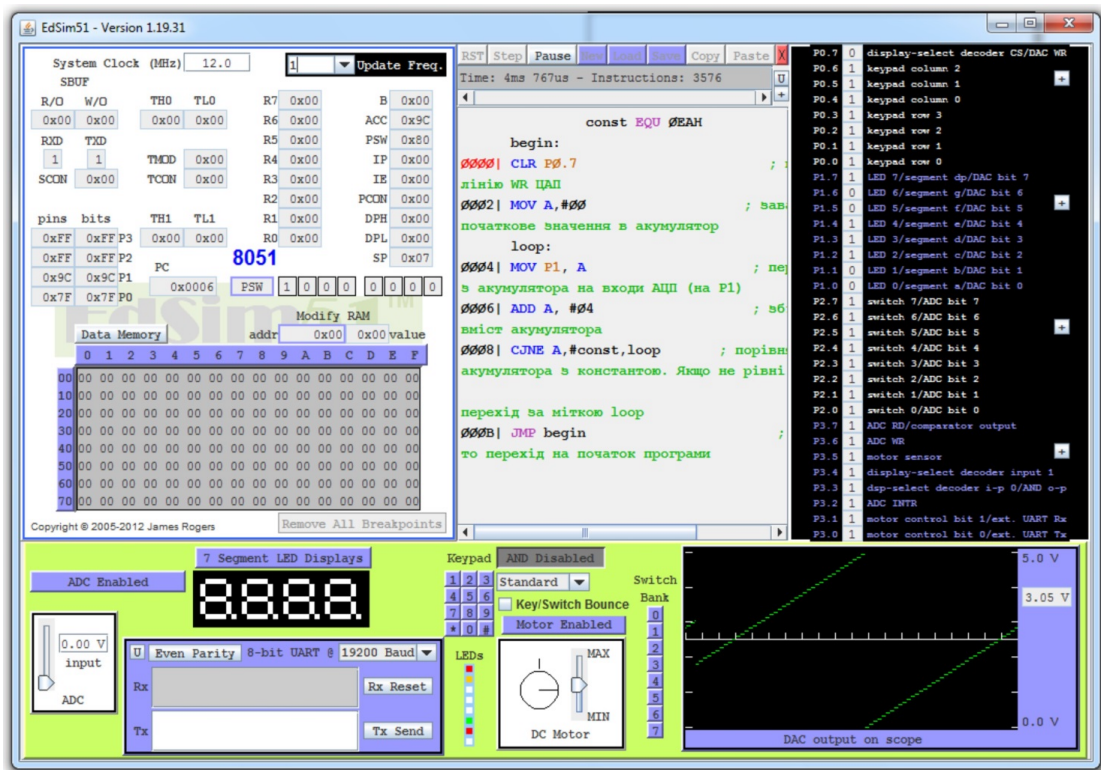


Рис. 5. Відображення вихідної напруги ЦАП

2. Ввід даних з АЦП і відображення їх на «DAC output on scope» за допомогою ЦАП

До початку виконання завдання перемикачі в блоку перемикачів «Switch Bank» SV0...SV7 повинні бути розімкнуті (перемикачі стають сірого кольору, коли закриті). Це потрібно тому, що перемикачі і АЦП під'єднані до одного і того ж порту P2 (рис. 4).

Програма 2 зчитує аналогову напругу з реостата «Input ADC» на панелі АЦП і відображає його на область «DAC output on scope» за допомогою ЦАП.

Напруга на виході АЦП опитується через кожні 50 нс, що досягається шляхом встановлення таймера 0 на переривання основної програми кожні 50 мкс. Далі таймер 0 через ISR ініціює перетворення в АЦП. Коли перетворення завершується, лінія переривання АЦП, яка з'єднана з лінією зовнішнього переривання 0 MCS51, скидається в нуль. Зовнішнє переривання 0 ISR зчитує дані з АЦП в порт P2 і передає його на ЦАП через порт P1.

При виконанні програми потрібно переконатися, що АЦП включений (виведено на інтерфейс симулятора напис «ADC Enabled»).

Програма 2

```

ORG 0           ; задати початкову адресу пам'яті програм
JMP main       ; перехід на процедуру main
ORG 3          ; вектор зовнішнього переривання 0
JMP ext0ISR    ; перейти до зовнішнього переривання 0 ISR
ORG 0BH       ; вектор переривання від таймера 0
JMP timer0ISR  ; переривання від таймера 0 ISR
ORG 30H       ; початкова адреса програми в пам'яті програм
main:
SETB IT0      ; встановити зовнішнє переривання 0, як активне («1»)
SETB EX0      ; дозволити зовнішнє переривання 0
CLR P0.7      ; активувати лінію WR DAC
MOV TMOD, #2  ; встановити таймер 0, як 8-бітний таймер з автоперезавантаженням
               ; з заданим інтервалом
MOV TH0, #-50 ; записати -50 в старший байт таймера 0 - це значення інтервалу

```



```

; перезавантаження,
; це при тактовій частоті 12 МГц призведе до переповнення
; таймера 0 кожні 50 нс
MOV TL0, #-50 ; записати однакове значення молодшого байту, щоб забезпечити
; запуск таймера за час, що рівний 236 (256 - 50)

SETB TR0 ; запуск таймера 0
SETB ET0 ; дозволити переривання від таймера 0
SETB EA ; встановити біт дозволу переривання
JMP $ ; перейти назад до тієї ж лінії (тобто: нічого не робити)
; кінець основної програми
; таймер 0 ISR - просто запускає АЦП перетворення

timer0ISR:
CLR P3.6 ; скидає в нуль лінію WR ADC
SETB P3.6 ; потім встановити його - це призводить до необхідного
; позитивного фронту, щоб почати перетворення

RETI ; повернення з підпрограми переривання
; ext0ISR - виконує перетворення АЦП через підпрограму
; переривання

ext0ISR:
CLR P3.7 ; очистити лінію RD АЦП - це дозволяє приймати дані від АЦП
MOV P1, P2 ; на лінії порту P2 і відправити його на лінії даних ЦАП через P1
SETB P3.7 ; відключити АЦП шляхом установки RD в "1"
RETI ; повернення з підпрограми переривання

Результат виконання Програми 2 подано на рис. 6 (горизонтальна зелена лінія на «DAC
output on scope», що відповідає рівню вхідної напруги 2,05 В).

```

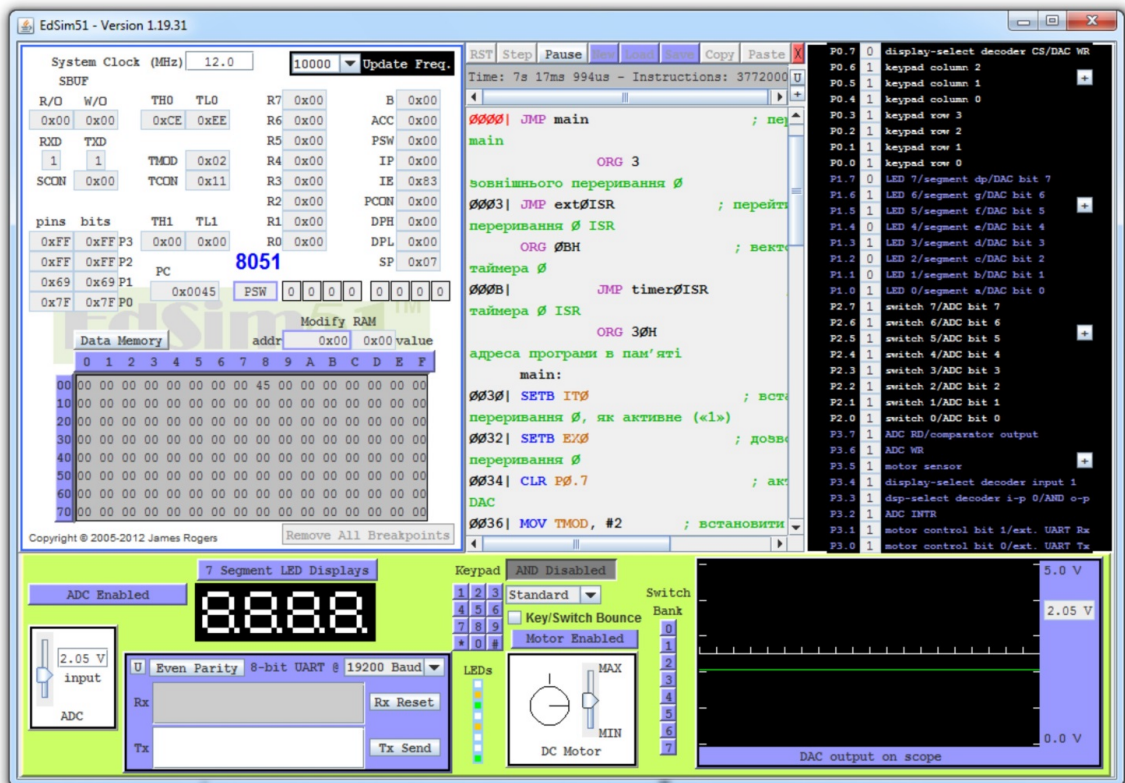


Рис. 6. Відображення результату виконання Програми 2 на EdSim51

Варіанти індивідуальних завдань

№	Зміст індивідуального завдання
1	<p>1. Написати і виконати на симуляторі програму для виведення сигналу на вихід ЦАП за прикладом Програми 1 для наступних даних:</p> <ul style="list-style-type: none"> - вивести на вихід ЦАП пилкоподібну напругу максимальною амплітудою 2,5В; - встановити крок квантування вихідної напруги $n=03$, виконати програму; - вивести на вихід ЦАП пилкоподібну напругу амплітудою 5В. Порівняти форму вихідної напруги з формою для попереднього значення амплітуди та кроку; - встановити крок квантування вихідної напруги $n=06$. Виконати програму. Порівняти форму вихідної напруги з формою напруги для попереднього значення кроку. <p>2. Написати і виконати на симуляторі програму для вводу напруги на вхід АЦП та її виводу через ЦАП за прикладом Програми 2 для наступних даних:</p> <ul style="list-style-type: none"> - встановити початкову адресу програми в пам'яті програм рівною 40Н; - встановити інтервал опитування напруги на вході АЦП рівним 40 нс.
2	<p>1. Написати і виконати на симуляторі програму для виведення сигналу на вихід ЦАП за прикладом Програми 1 для наступних даних:</p> <ul style="list-style-type: none"> - вивести на вихід ЦАП пилкоподібну напругу максимальною амплітудою 5В; - встановити крок квантування вихідної напруги $n=04$, виконати програму; - вивести на вихід ЦАП пилкоподібну напругу амплітудою 3В. Порівняти форму вихідної напруги з формою для попереднього значення амплітуди та кроку; - встановити крок квантування вихідної напруги $n=06$. Виконати програму. Порівняти форму вихідної напруги з формою напруги для попереднього значення кроку. <p>2. Написати і виконати на симуляторі програму для вводу напруги на вхід АЦП та її виводу через ЦАП за прикладом Програми 2 для наступних даних:</p> <ul style="list-style-type: none"> - встановити початкову адресу програми в пам'яті програм рівною 40Н; - встановити інтервал опитування напруги на вході АЦП рівним 40 нс.
3	<p>1. Написати і виконати на симуляторі програму для виведення сигналу на вихід ЦАП за прикладом Програми 1 для наступних даних:</p> <ul style="list-style-type: none"> - вивести на вихід ЦАП пилкоподібну напругу максимальною амплітудою 5В; - встановити крок квантування вихідної напруги $n=02$, виконати програму; - вивести на вихід ЦАП пилкоподібну напругу амплітудою 2В. Порівняти форму вихідної напруги з формою для попереднього значення амплітуди та кроку; - встановити крок квантування вихідної напруги $n=06$. Виконати програму. Порівняти форму вихідної напруги з формою напруги для попереднього значення кроку. <p>2. Написати і виконати на симуляторі програму для вводу напруги на вхід АЦП та її виводу через ЦАП за прикладом Програми 2 для наступних даних:</p> <ul style="list-style-type: none"> - встановити початкову адресу програми в пам'яті програм рівною 60Н; - встановити інтервал опитування напруги на вході АЦП рівним 60 нс.
4	<p>1. Написати і виконати на симуляторі програму для виведення сигналу на вихід ЦАП за прикладом Програми 1 для наступних даних:</p> <ul style="list-style-type: none"> - вивести на вихід ЦАП напругу трикутної форми максимальною амплітудою 5В; - встановити крок квантування вихідної напруги $n=01$, виконати програму; - вивести на вихід ЦАП напругу трикутної форми амплітудою 3В. Порівняти форму вихідної напруги з формою для попереднього значення амплітуди та кроку; - встановити крок квантування вихідної напруги $n=02$. Виконати програму. Порівняти форму вихідної напруги з формою напруги для попереднього значення кроку. <p>2. Написати і виконати на симуляторі програму для вводу напруги на вхід АЦП та її виводу через ЦАП за прикладом Програми 2 для наступних даних:</p> <ul style="list-style-type: none"> - встановити початкову адресу програми в пам'яті програм рівною 45Н; - встановити інтервал опитування напруги на вході АЦП рівним 45 нс.

№	Зміст індивідуального завдання
5	<p>1. Написати і виконати на симуляторі програму для виведення сигналу на вихід ЦАП за прикладом Програми 1 для наступних даних:</p> <ul style="list-style-type: none"> - вивести на вихід ЦАП напругу трикутної форми амплітудою 2,5В; - встановити крок квантування вихідної напруги $n=02$, виконати програму; - вивести на вихід ЦАП півперіод напруги синусоїдальної форми амплітудою 5В. <p>Початкова адреса таблиці синусів 40Н.</p> <p>2. Написати і виконати на симуляторі програму для вводу напруги на вхід АЦП та її виводу через ЦАП за прикладом Програми 2 для наступних даних:</p> <ul style="list-style-type: none"> - встановити початкову адресу програми в пам'яті програм рівною 55Н; - встановити інтервал опитування напруги на вході АЦП рівним 55 нс.
6	<p>1. Написати і виконати на симуляторі програму для виведення сигналу на вихід ЦАП за прикладом Програми 1 для наступних даних:</p> <ul style="list-style-type: none"> - вивести на вихід ЦАП напругу пилкоподібної форми амплітудою 3,5В; - встановити крок квантування вихідної напруги $n=03$, виконати програму; - вивести на вихід ЦАП напругу синусоїдальної форми амплітудою 5В від значення $x=0^0$ до $x=90^0$ з інтервалом в 10^0. Початкова адреса таблиці синусів 50Н. <p>2. Написати і виконати на симуляторі програму для вводу напруги на вхід АЦП та її виводу через ЦАП за прикладом Програми 2 для наступних даних:</p> <ul style="list-style-type: none"> - встановити початкову адресу програми в пам'яті програм рівною 60Н; - встановити інтервал опитування напруги на вході АЦП рівним 60 нс.
7	<p>1. Написати і виконати на симуляторі програму для виведення сигналу на вихід ЦАП за прикладом Програми 1 для наступних даних:</p> <ul style="list-style-type: none"> - вивести на вихід ЦАП напругу пилкоподібної форми амплітудою 2,5В; - встановити крок квантування вихідної напруги $n=04$, виконати програму; - вивести на вихід ЦАП напругу синусоїдальної форми амплітудою 5В від значення $x=90^0$ до $x=0^0$ з інтервалом в 10^0. Початкова адреса таблиці синусів 60Н. <p>2. Написати і виконати на симуляторі програму для вводу напруги на вхід АЦП та її виводу через ЦАП за прикладом Програми 2 для наступних даних:</p> <ul style="list-style-type: none"> - встановити початкову адресу програми в пам'яті програм рівною 40Н; - встановити інтервал опитування напруги на вході АЦП рівним 40 нс.
8	<p>1. Написати і виконати на симуляторі програму для виведення сигналу на вихід ЦАП за прикладом Програми 1 для наступних даних:</p> <ul style="list-style-type: none"> - вивести на вихід ЦАП напругу трикутної форми амплітудою 2,5В; - встановити крок квантування вихідної напруги $n=02$, виконати програму; - вивести на вихід ЦАП напругу синусоїдальної форми амплітудою 5В від значення $x=90^0$ до $x=0^0$ з інтервалом в 10^0. Початкова адреса таблиці синусів 45Н. <p>2. Написати і виконати на симуляторі програму для вводу напруги на вхід АЦП та її виводу через ЦАП за прикладом Програми 2 для наступних даних:</p> <ul style="list-style-type: none"> - встановити початкову адресу програми в пам'яті програм рівною 60Н; - встановити інтервал опитування напруги на вході АЦП рівним 60 нс.
9	<p>1. Написати і виконати на симуляторі програму для виведення сигналу на вихід ЦАП за прикладом Програми 1 для наступних даних:</p> <ul style="list-style-type: none"> - вивести на вихід ЦАП напругу трикутної форми амплітудою 3,5В; - встановити крок квантування вихідної напруги $n=03$, виконати програму; - вивести на вихід ЦАП напругу синусоїдальної форми амплітудою 5В від значення $x=0^0$ до $x=90^0$ з інтервалом в 10^0. Початкова адреса таблиці синусів 55Н. <p>2. Написати і виконати на симуляторі програму для вводу напруги на вхід АЦП та її виводу через ЦАП за прикладом Програми 2 для наступних даних:</p> <ul style="list-style-type: none"> - встановити початкову адресу програми в пам'яті програм рівною 45Н; - встановити інтервал опитування напруги на вході АЦП рівним 60 нс.

№	Зміст індивідуального завдання
10	<p>1. Написати і виконати на симуляторі програму для виведення сигналу на вихід ЦАП за прикладом Програми 1 для наступних даних:</p> <ul style="list-style-type: none"> - вивести на вихід ЦАП пилкоподібну напругу максимальною амплітудою 5В; - встановити крок квантування вихідної напруги $n=04$, виконати програму; - вивести на вихід ЦАП пилкоподібну напругу амплітудою 2В. Порівняти форму вихідної напруги з формою для попереднього значення амплітуди та кроку; - встановити крок квантування вихідної напруги $n=01$. Виконати програму. Порівняти форму вихідної напруги з формою напруги для попереднього значення кроку. <p>2. Написати і виконати на симуляторі програму для вводу напруги на вхід АЦП та її виводу через ЦАП за прикладом Програми 2 для наступних даних:</p> <ul style="list-style-type: none"> - встановити початкову адресу програми в пам'яті програм рівною 65Н; - встановити інтервал опитування напруги на вході АЦП рівним 60 нс.
11	<p>1. Написати і виконати на симуляторі програму для виведення сигналу на вихід ЦАП за прикладом Програми 1 для наступних даних:</p> <ul style="list-style-type: none"> - вивести на вихід ЦАП напругу трикутної форми максимальною амплітудою 5В; - встановити крок квантування вихідної напруги $n=02$, виконати програму; - вивести на вихід ЦАП напругу трикутної форми амплітудою 4В. Порівняти форму вихідної напруги з формою для попереднього значення амплітуди та кроку; - встановити крок квантування вихідної напруги $n=04$. Виконати програму. Порівняти форму вихідної напруги з формою напруги для попереднього значення кроку. <p>2. Написати і виконати на симуляторі програму для вводу напруги на вхід АЦП та її виводу через ЦАП за прикладом Програми 2 для наступних даних:</p> <ul style="list-style-type: none"> - встановити початкову адресу програми в пам'яті програм рівною 65Н; - встановити інтервал опитування напруги на вході АЦП рівним 60 нс.
12	<p>1. Написати і виконати на симуляторі програму для виведення сигналу на вихід ЦАП за прикладом Програми 1 для наступних даних:</p> <ul style="list-style-type: none"> - вивести на вихід ЦАП напругу трикутної форми амплітудою 3,5В; - встановити крок квантування вихідної напруги $n=02$, виконати програму; - вивести на вихід ЦАП півперіод напруги синусоїдальної форми амплітудою 5В. Початкова адреса таблиці синусів 60Н. <p>2. Написати і виконати на симуляторі програму для вводу напруги на вхід АЦП та її виводу через ЦАП за прикладом Програми 2 для наступних даних:</p> <ul style="list-style-type: none"> - встановити початкову адресу програми в пам'яті програм рівною 40Н; - встановити інтервал опитування напруги на вході АЦП рівним 60 нс.

4. Послідовність виконання роботи

4.1. Виконати програми відповідно до індивідуального завдання. Виконувати команди кожної програми потрібно в такій послідовності:

4.1.1. Відкрити інтерфейс емулятора, двічі клацнувши клавішею миші на архівованому файлі «EdSim51.jar». Відкриється інтерфейс програмного симулятора, зображений на рис. 7.

Середнє поле симулятора, що називається “Панель коду Асемблера”, в верхній частині містить кнопки “Reset”, “Assm”, “Run”, “Load”, “Save”, “Copy”, “Past”.

Панель коду використовується для:

- **набору команд** програми з клавіатури. Для цього курсор встановлюється в верхній частині панелі і вводиться програма по одній команді в рядку (при потребі, з міткою та коментарем)(див. рис. 7);
- **завантаження** вже існуючої програми. Для цього необхідно на панелі вгорі натиснути кнопку “Load” і вказати шлях до потрібного файлу;
- **запису** набраного файлу. Для цього потрібно натиснути кнопку “Save” і вказати шлях для збереження файлу.

4.1.2. Перед виконанням програми необхідно натиснути кнопку “Assm” панелі для асемблювання програми. Після цього, якщо команда записана невірною, в рядку під верхнім

рядом кнопок панелі (на рис.7 виділений сірим кольором) з'явиться повідомлення про помилку, а колір рядка зміниться на червоний. Червоним кольором буде виділена також невірно написана команда.

Якщо помилки відсутні, зліва від команд набраної програми з'являться адреси, і сама програма буде готова до виконання. Після асемблювання кнопка “Assm” зміниться на кнопку “Step”. Таким чином, є можливим виконувати програму покомандно в кроковому режимі, натискаючи кнопку “Step” після виконання кожної команди, або в автоматичному режимі, коли виконується вся програма, натиснувши один раз кнопку “Run”. В останньому випадку програму слід закінчувати директивою “End”.

При написанні програми можна користуватися для копіювання її фрагментів та вставки в будь-якому місці “Панелі коду Асемблера” кнопками “Copy” та “Past”.

Щоб зупинити виконання програми і скинути в початковий стан реєстри мікроконтролера симулятора необхідно натиснути кнопку “Reset”.

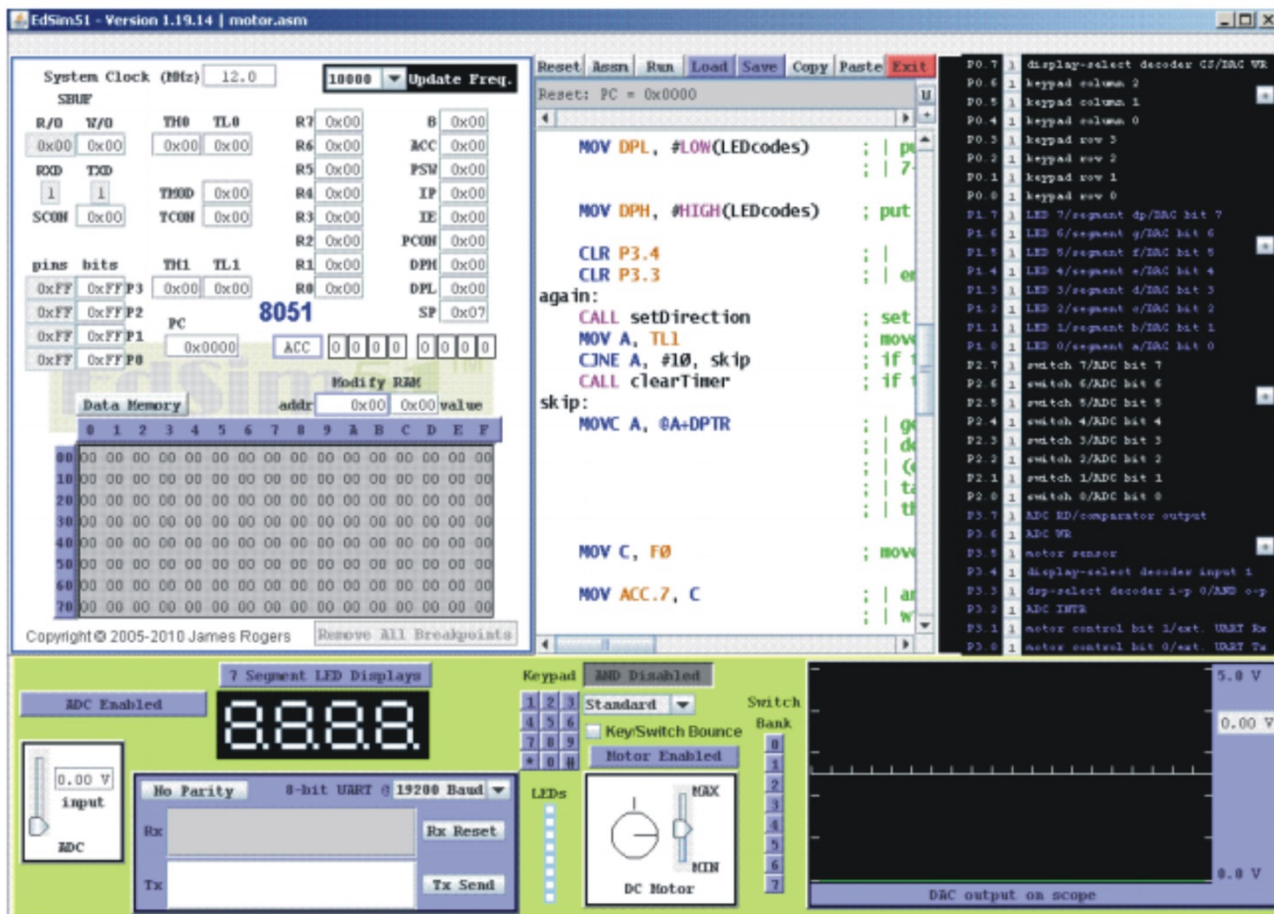


Рис. 7. Інтерфейс програмного симулятора

*Примітка

1. Якщо Ви хочете виконати якусь з команд над вмістом реєстру чи комірки пам'яті, наприклад, команду пересилання з реєстру в реєстр, необхідно в реєстр, з якого буде здійснене пересилання, командою MOV попередньо записати якесь значення операнду (адресу чи константу).

2. Програма, що виконується, буде записана в пам'ять програм, вміст якої можна побачити, натиснувши на кнопку “Data memory” в нижній частині “Панелі пам'яті даних та програмної пам'яті”, що знаходиться зліва від “Панелі коду Асемблера”. Після натискання кнопка “Data memory” зміниться на кнопку “Code memory”, тобто буде висвічуватися в полі пам'яті вміст пам'яті програм.

5. Контрольні запитання

1. Яке призначення АЦП?

2. Яке призначення ЦАП?
3. Які методи перетворення вхідного сигналу реалізуються схемами АЦП?
4. Від чого залежить похибка вимірювання АЦП?
5. Від яких факторів залежить точність відтворення сигналу на виході ЦАП?
6. Пояснити алгоритм роботи Програми 1.
7. Пояснити алгоритм роботи Програми 2.

Література

1. The 8051 Simulator for Teachers and Students\ \ Інтернет ресурс: <http://www.edsim51.com>.
2. Основи вимірювальної техніки: методичний вказівки до курсу для студентів денної та заочної форм навчання спеціальності 122 «Комп'ютерні науки». ДВНЗ «Ужгородський національний університет». 2020 р.

Додаток 1

1. Програма виведення пілкоподібного сигналу на «DAC output on score» симулятора з можливістю регулювання частоти сигналу при сталому значенні амплітуди та кроку квантування через організацію циклу затримки в часі.

```

const EQU 0EAH    ; встановити максимальне значення напруги на виході ЦАП
  n EQU 04H      ; встановити значення кроку квантування
  m EQU 08H      ; встановити значення затримки в часі для зміни частоти
                  ; вихідної напруги ЦАП
start: MOV R3,#m  ; завантажити значення затримки в регістр R3
      MOV R2,#00  ; завантажити початкове значення напруги в регістр R2
      CLR P0.7    ; включити лінію WR ЦАП
loop:  MOV A,R2    ; переслати вміст регістра R2 в акумулятор
loop1: MOV P1, A   ; переслати дані з акумулятора на входи ЦАП (на P1)
      DJNZ R3, loop1 ; якщо вміст R3=0, то перейти на наступну команду, якщо ні, то
                  ; перейти на мітку loop1
      ADD A, #n    ; збільшити вміст акумулятора на крок квантування
      MOV R2,A     ; зберегти вміст акумулятора в регістрі R2
      SUBB A,#const ; відняти від вмісту A значення максимальної амплітуди const
      JC loop      ; перейти за міткою loop, якщо вміст A менший за const (флажок
                  ; C=«1»)
      JMP start    ; якщо вміст A більший за const (флажок C=«0»)
                  ; перехід на початок програми за міткою start

```

2. Програма виведення сигналу у формі трикутника на «DAC output on score» симулятора.

```

start: const EQU 0EAH    ; встановити максимальне значення вихідної напруги на виході ЦАП
      n EQU 04H      ; встановити значення кроку квантування
      CLR P0.7      ; включити лінію WR ЦАП
      MOV R2,#00    ; завантажити початкове значення напруги в регістр R2
loop:  MOV A,R2      ; переслати вміст регістра R2 в акумулятор
      MOV P1, A     ; переслати дані з акумулятора на входи ЦАП (на P1)
      ADD A, #n     ; збільшити вміст акумулятора
      MOV R2,A      ; зберегти вміст акумулятора в регістрі R2
      SUBB A,#const ; відняти від вмісту A значення максимальної амплітуди const
      JC loop      ; перейти за міткою loop, якщо вміст A менший за const (флажок
                  ; C= «1»)

```

```

; якщо вміст A більший за const (флажок C=«0»), перейти на
; наступну команду
loop1:
MOV A, R2      ; переслати дані з регістра R2 в акумулятор
MOV P1, A     ; переслати дані з акумулятора на входи ЦАП (на P1)
SUBB A, #n    ; зменшити вміст акумулятора на крок квантування n
MOV R2,A     ; зберегти вміст акумулятора в регістрі R2
SUBB A, #00   ; порівняти вміст акумулятора з нулем
JNZ loop1    ; якщо вміст акумулятора не рівний «0» (флажок Z=«0»), перехід за
; міткою loop1
JMP loop     ; якщо вміст акумулятора рівний «0» (флажок Z=«1»), перехід за
; міткою loop

```

3. Програма виведення сигналу у формі півперіоду синусоїди на «DAC output on score» симулятора. Значення $\sin x$ зберігається в R3 і виводиться в порт P1 на ЦАП.

```

; Обчислення півперіоду SIN (x) по таблиці значень
; вхід: X в A (акумуляторі) в межах від 0 до 89 градусів,
; вихід: значення синуса в акумуляторі

MOV DPTR,#30H ; встановити початкову адресу для таблиці синусів
m EQU 12H     ; встановити як m кількість чисел в масиві даних пам'яті
; програм
CLR P0.7     ; включити лінію WR ЦАП
CLR A       ; обнулити вміст акумулятора
MOV R4,#m   ; завантажити лічильник циклів
SINX:
MOV R2,A    ; зберегти поточне значення A
MOVC A,@A+DPTR ; завантаження значення синуса з таблиці в акумулятор
MOV R3,A   ; зберегти в R3 значення  $\sin x$ 
MOV P1,R3  ; і вивести його в порт P1
MOV A,R2   ; відновити значення A для адресації таблиці синусів
INC A     ; інкремент акумулятора
DJNZ R4,SINX ; якщо R4 нуль, то наступна команда, інакше перехід на
; мітку SINX
CLR A     ; обнулити вміст акумулятора
MOV R4,#m ; завантажити лічильник циклів
JMP SINX

; таблиця значень синусів
ORG 30H
DB 0 ; sin (0) = 0
DB 00100111b ; sin (10) = 0.156
DB 01011000b ; sin (20) = 0.342
DB 10000000b ; sin (30) = 0.5
DB 10100100b ; sin (40) = 0.643
DB 11000100b ; sin (50) = 0.766
DB 11011101b ; sin (60) = 0.866
DB 11110000b ; sin (70) = 0.940
DB 11111110b ; sin (80) = 0.985
DB 11111111b ; sin (90) = 1.0
DB 11111111b ; sin (100) = 0.985
DB 11111110b ; sin (110) = 0.940
DB 11011101b ; sin (120) = 0.866

```

DB 11000100b	; sin (130) = 0.766
DB 10100100b	; sin (140) = 0.643
DB 10000000b	; sin (150) = 0.5
DB 01011000b	; sin (160) = 0.342
DB 00100111b	; sin (170) = 0.156

Рекомендована література

1. Проектування мікропроцесорних систем керування : навчальний посібник, перевидання / Медвідь В.Р., Письціо В.П., Козбур І.Р. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 360 с.
2. Handbook of Microcontrollers/Predko Michael. NYс. McGraw-Hill. 1998. 861 p.
3. Бойко В. І., Гуржій А. М., Жуйков В. Я. та ін.Схемотехніка електронних схем: У 3 кн. Кн.3 Мікропроцесори та мікроконтролери: підручник. 2-ге вид., допов. і переробл. К.: Вища шк., 2004. 399 с.
- 4 Міліх В. І., Шавьолкін О. О. Електротехніка, електроніка та мікропроцесорна техніка: підручник; за ред. В. І. Міліх. 2-е вид. К.: Каравела, 2008. 688 с.