ЛАБОРАТОРНА РОБОТА 4

Покрокова інструкція з виконання у Logisim Evolution 5.0.0

Дослідження арифметичних цифрових пристроїв

Напівсуматор • Повний суматор • Чотирибітний суматор • Суматор-віднімач

Параметр	Використане значення
Програма	Logisim Evolution
Версія	5.0.0, стабільний випуск
Основна платформа	Windows 10 або Windows 11, 64-розрядна Intel/AMD
Файл проєкту студента	LR4_Прізвище_Група.circ
Орієнтовний час	45–60 хвилин після встановлення програми

Інструкція розрахована на студента, який уперше відкриває програму.

1. Що буде створено

У межах роботи створюється один проєкт Logisim Evolution з чотирма схемами: HA, FA, RCA4 і ADD_SUB4. Спочатку будується напівсуматор, потім з двох напівсуматорів — повний суматор, далі чотири повні суматори об'єднуються у 4-бітний суматор. Завершальна схема виконує додавання або віднімання залежно від сигналу SUB.

Схема	Призначення	Основні входи та виходи
HA	Додавання двох бітів без вхідного перенесення	A, B → S, C
FA	Додавання двох бітів з вхідним перенесенням	A, B, Cin → S, Cout
RCA4	Додавання двох 4-бітних слів і Cin	A3…A0, B3…B0, Cin → S3…S0, Cout, V
ADD_SUB4	Кероване додавання або віднімання	A3…A0, B3…B0, SUB → R3…R0, Cout, Borrow, V

2. Що потрібно підготувати

- Комп'ютер з Windows 10 або Windows 11 та правом установлення програм.
- Номер індивідуального варіанта з основного файлу лабораторної роботи 4.
- Папку для збереження файлу .circ і знімків екрана для звіту.
- Приблизно 80 МБ для інсталятора і додаткове місце для встановленої програми.

Важливо. Не використовуйте старий Logisim 2.7.x або вебверсію Logisim. Розташування команд і набір властивостей у них відрізняються від цієї інструкції.

3. Завантаження Logisim Evolution 5.0.0

Для більшості навчальних комп'ютерів з процесором Intel або AMD оберіть інсталятор Windows x64. Він уже містить середовище Java, тому встановлювати Java окремо не потрібно.

Офіційна сторінка випусків: github.com/logisim-evolution/logisim-evolution/releases

Пряме завантаження для Windows Intel/AMD: logisim-evolution-5.0.0-amd64.msi

1. **Відкрийте офіційну сторінку.** У браузері натисніть посилання на сторінку випусків. На початку списку знайдіть Version 5.0.0 з позначкою Latest.
2. **Розгорніть список файлів.** Під описом версії натисніть Assets, якщо список файлів згорнуто.
3. **Виберіть правильний інсталятор.** Для звичайного ПК або ноутбука з Intel/AMD натисніть logisim-evolution-5.0.0-amd64.msi. Не обирайте Source code і не обирайте aarch64.msi, якщо у вас не ARM-комп'ютер.
4. **Дочекайтеся завантаження.** Файл зазвичай з'явиться у папці Downloads або Завантаження.

3.1. Як перевірити тип процесора Windows

1. **Відкрийте параметри.** Натисніть Start → Settings → System → About.

2. **Знайдіть рядок System type.** Якщо написано x64-based processor, потрібен файл amd64.msi. Якщо написано ARM-based processor, потрібен файл aarch64.msi.
3. **Для ARM використайте інший файл.** На тій самій сторінці випуску виберіть logisim-evolution-5.0.0-aarch64.msi. Подальші дії в програмі однакові.

4. Установлення програми у Windows

1. **Запустіть інсталятор.** Відкрийте папку Downloads і двічі клацніть logisim-evolution-5.0.0-amd64.msi.
2. **Підтвердьте запуск.** Якщо Windows показує запит User Account Control, перевірте назву файла і натисніть Yes. Якщо комп'ютер університету не дозволяє встановлення, зверніться до викладача або адміністратора; не вимикайте захист Windows.
3. **Пройдіть майстер установлення.** У вікні Logisim Evolution Setup натисніть Next, залиште запропоновану папку встановлення, натисніть Next, потім Install.
4. **Завершіть установлення.** Після копіювання файлів натисніть Finish.
5. **Запустіть програму.** Натисніть Start, введіть Logisim Evolution і відкрийте знайдену програму.
6. **Перевірте версію.** У програмі відкрийте Help → About. У вікні має бути вказано Version 5.0.0. Закрийте вікно кнопкою OK.

> **Якщо інсталятор не запускається.** Повторно завантажте файл лише з офіційної сторінки. Для portable-файла logisim-evolution-5.0.0-all.jar потрібна Java 21 або новіша, тому для початківців рекомендований саме MSI-пакет.

5. Перше налаштування програми

Щоб назви команд у всіх студентів збігалися з інструкцією, встановіть англійську мову інтерфейсу та фігурне подання логічних елементів ANSI.

1. **Відкрийте параметри.** У верхньому меню виберіть File → Preferences...
2. **Відкрийте вкладку International.** У полі Language виберіть English.
3. **Установіть форму елементів.** У полі Gate shape виберіть ANSI. Після цього AND, OR та XOR мають стандартну фігурну форму, а не прямокутники IEC.
4. **Закрийте параметри.** Закрийте вікно Preferences. Якщо назви меню не оновилися відразу, закрийте Logisim Evolution і запустіть його повторно.
5. **Перевірте моделювання.** У меню Simulate біля Auto-Propagate має стояти позначка. Якщо її немає, клацніть Auto-Propagate один раз.

6. Основні частини вікна

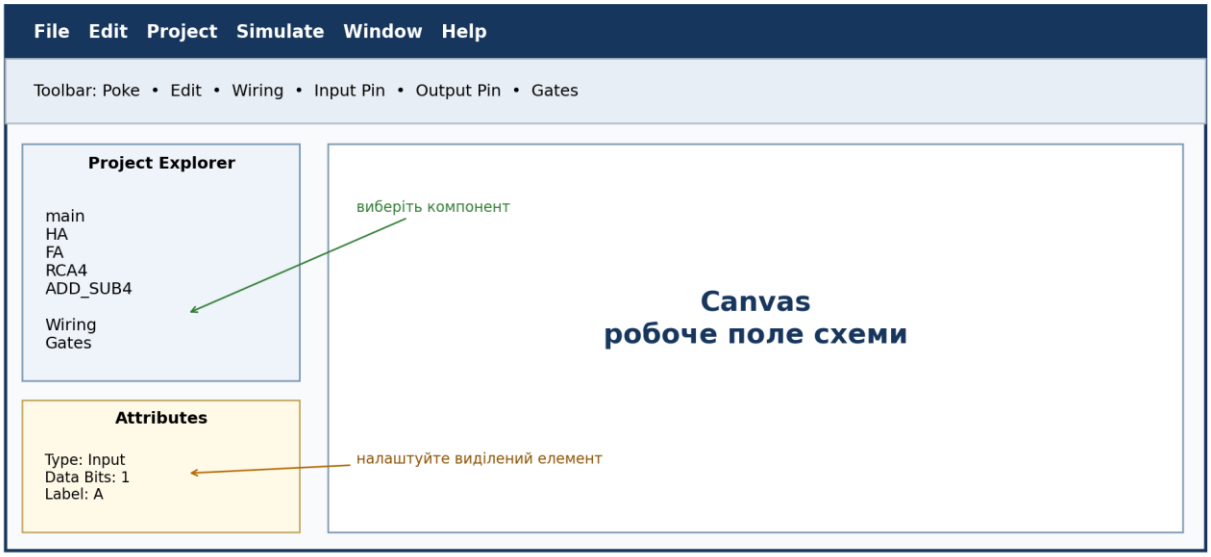

Рисунок 1. Орієнтири в інтерфейсі Logisim Evolution 5.0.0

Частина вікна	Для чого потрібна
Project Explorer	Містить схеми проєкту та бібліотеки компонентів Wiring, Gates, Arithmetic тощо.
Attributes	Показує властивості вибраного компонента: Type, Facing, Data Bits, Label, Number Of Inputs.
Canvas	Робоче поле, у якому розміщують елементи і прокладають з'єднання.
Poke Tool	Інструмент із піктограмою руки; перемикає входи між 0 і 1.
Edit Tool	Стрілка вибору; дозволяє виділяти, переміщувати й видаляти компоненти.
Wiring Tool	Прокладає дроти між контактами компонентів.

7. Створення та збереження проєкту

1. **Створіть новий проєкт.** Виберіть File → New.
2. **Одразу збережіть файл.** Виберіть File → Save As...
3. **Оберіть папку.** Створіть окрему папку LR4 або виберіть папку для лабораторних робіт.
4. **Задайте ім'я.** Введіть LR4_Прізвище_Група.circ, наприклад LR4_Ivanenko_KN21.circ, і натисніть Save.
5. **Зберігайте після кожного етапу.** Натискайте Ctrl+S після створення HA, FA, RCA4 і ADD_SUB4.

Правило імен. У назвах схем використовуйте латинські літери, цифри та знак підкреслення. Це зменшує ризик помилок під час перенесення проєкту між комп'ютерами.

8. Загальні правила роботи з компонентами

1. **Розміщення.** У Project Explorer розкрийте потрібну бібліотеку, клацніть назву компонента, а потім один раз клацніть на Canvas.
2. **Налаштування.** Перейдіть на Edit Tool, клацніть компонент і змініть потрібні поля в панелі Attributes.
3. **Поворот.** Для більшості елементів встановлюйте Facing = East, щоб сигнали рухалися зліва направо.
4. **З'єднання.** Оберіть Wiring Tool. Натисніть на маленьку точку контакту, протягніть дріт до потрібного контакту і відпустіть кнопку миші.
5. **Відгалуження.** Щоб зробити гілку, почніть новий дріт із уже прокладеного дроту. У місці електричного з'єднання з'являється вузол.
6. **Видалення.** Виберіть Edit Tool, клацніть компонент або дріт і натисніть Delete.
7. **Скасування.** Якщо дія виконана неправильно, натисніть Ctrl+Z або Edit → Undo.
8. **Перевірка сигналів.** За стандартних кольорів темно-зелений дріт означає 0, яскраво-зелений — 1, синій — невизначений сигнал, червоний — помилку, помаранчевий — невідповідність розрядності.

9. Створення схеми HA

HA (Half Adder — напівсуматор) додає два однорозрядні числа A і B. Вихід S зберігає молодший біт суми, а C — перенос у наступний розряд.

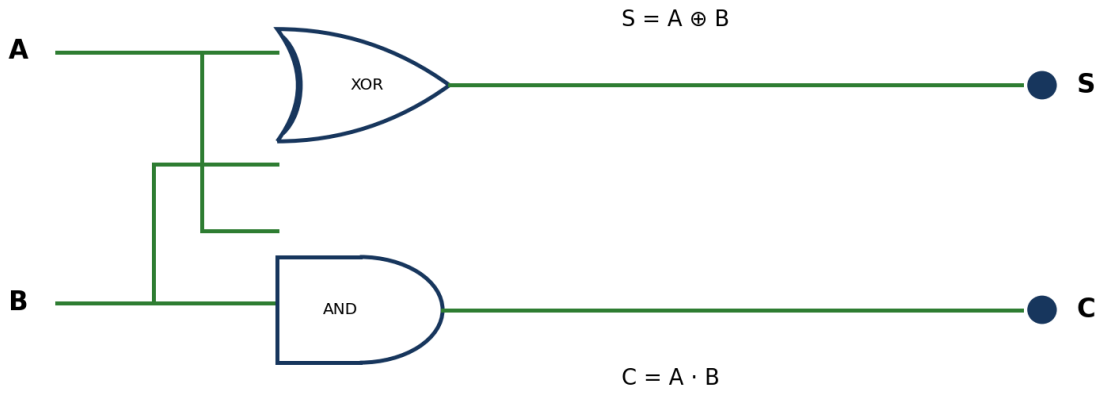

Рисунок 2. Структура напівсуматора

9.1. Додайте окрему схему

1. **Відкрийте команду.** Виберіть Project → Add Circuit...
2. **Назвіть схему.** Введіть HA і натисніть OK. У Project Explorer з'явиться HA, а робоче поле переключиться на цю схему.

3. **Переконайтеся, що редагуєте HA.** Назва HA має бути виділена у списку схем.

9.2. Розмістіть входи A і B

1. **Виберіть Pin.** У бібліотеці Wiring клацніть Pin.
2. **Створіть вхід A.** Клацніть у лівій верхній частині Canvas. У Attributes встановіть Type = Input, Facing = East, Data Bits = 1, Label = A.
3. **Створіть вхід B.** Ще раз виберіть Wiring → Pin, клацніть нижче A і задайте Type = Input, Facing = East, Data Bits = 1, Label = B.

9.3. Розмістіть виходи S і C

1. **Створіть S.** Виберіть Wiring → Pin, клацніть праворуч у верхній частині. У Attributes встановіть Type = Output, Facing = West, Data Bits = 1, Label = S.
2. **Створіть C.** Створіть ще один вихід нижче S з Type = Output, Facing = West, Data Bits = 1, Label = C.

9.4. Додайте вентилі та проведіть дроти

1. **Додайте XOR.** Відкрийте Gates → XOR Gate, клацніть між входами і виходом S. У Attributes установіть Facing = East, Data Bits = 1, Number Of Inputs = 2.
2. **Додайте AND.** Відкрийте Gates → AND Gate, розмістіть його між входами і виходом C. Установіть Facing = East, Data Bits = 1, Number Of Inputs = 2.
3. **З'єднайте A з XOR.** Wiring Tool: проведіть дріт від контакту A до верхнього входу XOR.
4. **З'єднайте B з XOR.** Проведіть дріт від B до нижнього входу XOR.
5. **Подайте ті самі сигнали на AND.** Створіть відгалуження від ліній A і B до двох входів AND. Не з'єднуйте A і B між собою.
6. **Підключіть виходи.** З'єднайте вихід XOR з S, а вихід AND з C.
7. **Збережіть.** Натисніть Ctrl+S.

> **Ознака правильної схеми.** Усі контакти з'єднані, синіх або червоних дротів немає, а перемикання A або B відразу змінює S і C.

10. Перевірка напівсуматора

1. **Оберіть Poke Tool.** Натисніть піктограму руки на панелі інструментів.
2. **Установіть A=0, B=0.** Клацайте вхідні піни, доки біля них не буде 0. Запишіть S і C.
3. **Перевірте решту комбінацій.** Послідовно задайте 01, 10 і 11 та порівняйте покази з таблицею.
4. **Виправте невідповідність.** Якщо рядок не збігається, перевірте, чи S підключено до XOR, C — до AND і чи обидва вентилі отримують A та B.

A	B	Очікуване S	Очікуване C
0	0	0	0
0	1	1	0

A	B	Очікуване S	Очікуване C
1	0	1	0
1	1	0	1

11. Створення схеми FA

FA (Full Adder — повний суматор) додає A, B та вхідне перенесення Cin. Тут він будується з двох готових HA і вентиля OR, щоб показати ієрархічне проєктування.

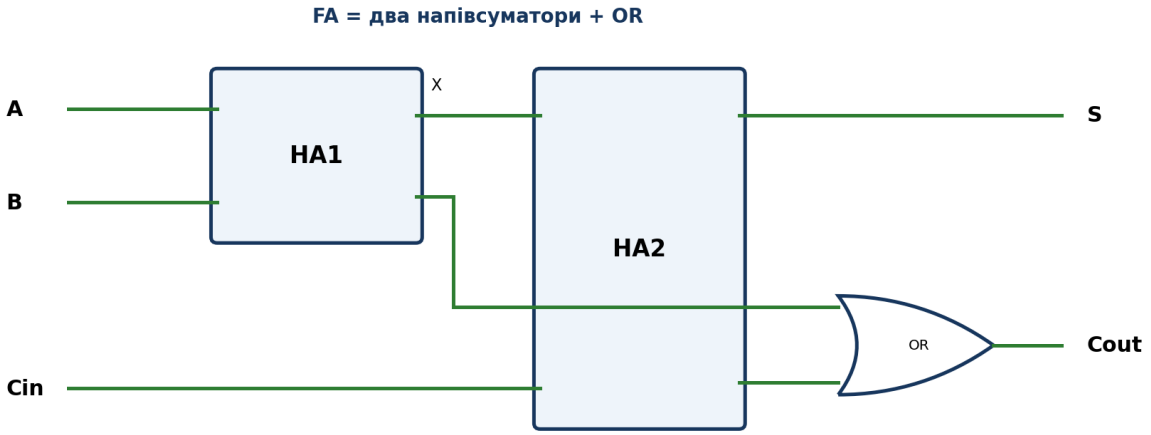

Рисунок 3. Повний суматор із двох напівсуматорів

1. **Створіть нову схему.** Виберіть Project → Add Circuit..., введіть FA і натисніть OK.
2. **Створіть входи.** Додайте три Wiring → Pin з Type = Input, Data Bits = 1, Facing = East. Задайте мітки A, B і Cin.
3. **Створіть виходи.** Додайте два Wiring → Pin з Type = Output, Data Bits = 1, Facing = West. Задайте мітки S і Cout.
4. **Розмістіть перший HA.** У Project Explorer клацніть схему HA як компонент і поставте її в лівій центральній частині Canvas.
5. **Розмістіть другий HA.** Ще раз виберіть HA і поставте його правіше першого.
6. **Додайте OR.** Виберіть Gates → OR Gate, установіть Number Of Inputs = 2, Data Bits = 1, Facing = East і розмістіть нижче другого HA.
7. **Підключіть перший HA.** A і B підключіть до входів першого HA.
8. **Підключіть другий HA.** Вихід S першого HA підключіть до одного входу другого HA, а Cin — до другого.
9. **Сформуйте S.** Вихід S другого HA підключіть до вихідного піна S схеми FA.
10. **Сформуйте Cout.** Виходи C обох HA підключіть до двох входів OR; вихід OR підключіть до Cout.
11. **Збережіть.** Натисніть Ctrl+S.

Як розпізнати контакти HA. Якщо підписи контактів на блоці дрібні, наведіть курсор на контакт або відкрийте HA подвійним клацанням у Project Explorer і звірте положення входів та виходів. Не підключайте дріт навмання.

12. Перевірка повного суматора

Перевірте всі вісім комбінацій. Cout дорівнює 1, коли серед A, B і Cin є щонайменше дві одиниці.

A	B	Cin	Очікуване S	Очікуване Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

1. **Перейдіть до FA.** Клацніть FA у Project Explorer.
2. **Увімкніть Poke Tool.** По черзі задавайте комбінації з таблиці.
3. **Звірте S і Cout.** Після кожної зміни дочекайтеся оновлення виходів і порівняйте їх з очікуваними.
4. **Зробіть контрольний знімок.** Задайте A=1, B=1, Cin=1. На виходах має бути S=1, Cout=1. Збережіть знімок екрана для звіту.

13. Створення чотирибітного суматора RCA4

RCA (Ripple Carry Adder — суматор із послідовним поширенням перенесення) складається з чотирьох FA. FA0 обробляє молодший біт, FA3 — старший. Вихід Cout кожного розряду подається на Cin наступного.

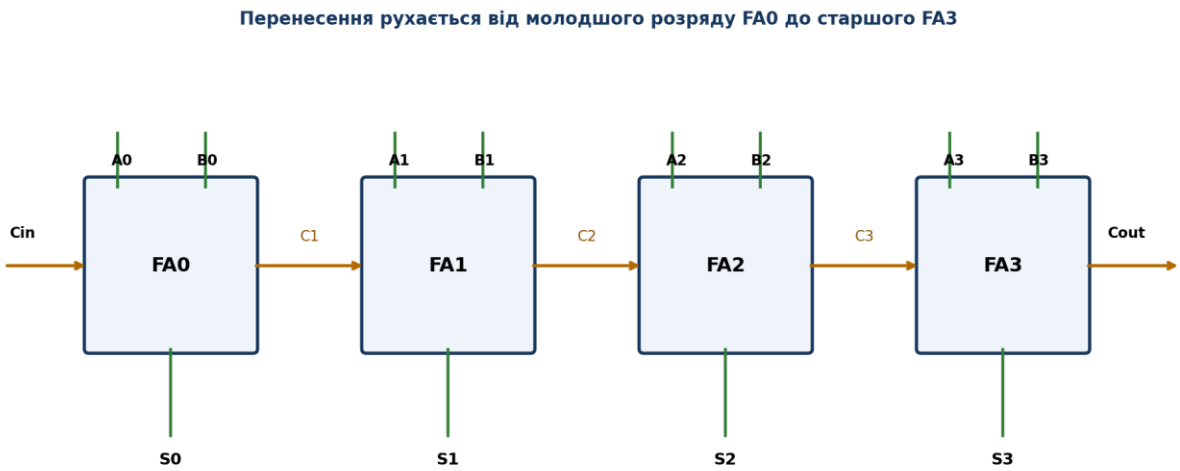

Рисунок 4. Ланцюг перенесення у RCA4

13.1. Створіть схему та компоненти

1. **Додайте схему.** Project → Add Circuit... → введіть RCA4 → OK.
2. **Додайте входи A.** Розмістіть чотири вхідні піни та підпишіть A0, A1, A2, A3. A0 — молодший біт, A3 — старший.
3. **Додайте входи B.** Розмістіть чотири вхідні піни B0, B1, B2, B3.
4. **Додайте Cin.** Створіть ще один вхідний пін з Label = Cin.
5. **Додайте виходи суми.** Розмістіть чотири вихідні піни S0, S1, S2, S3.
6. **Додайте ознаки.** Створіть вихідні піни Cout і V.
7. **Розмістіть чотири FA.** Чотири рази виберіть FA у Project Explorer і розмістіть блоки зліва направо. Лівий блок вважайте FA0, правий — FA3.
8. **Додайте XOR для V.** Виберіть Gates → XOR Gate, установіть Number Of Inputs = 2 і розмістіть біля старшого розряду.

13.2. З'єднайте розряди

1. **Підключіть молодший розряд.** A0 і B0 подайте на FA0, а зовнішній Cin — на Cin блока FA0. Вихід S FA0 подайте на S0.
2. **Підключіть розряд 1.** A1 і B1 подайте на FA1; Cout FA0 з'єднайте з Cin FA1; S FA1 подайте на S1.
3. **Підключіть розряд 2.** A2 і B2 подайте на FA2; Cout FA1 з'єднайте з Cin FA2; S FA2 подайте на S2.

4. **Підключіть старший розряд.** A3 і B3 подайте на FA3; Cout FA2 з'єднайте з Cin FA3; S FA3 подайте на S3.
5. **Сформуйте Cout.** Cout FA3 подайте на зовнішній вихід Cout.
6. **Сформуйте Overflow.** Подайте C3, тобто Cout FA2, на один вхід доданого XOR. Подайте C4, тобто Cout FA3, на другий вхід XOR. Вихід XOR з'єднайте з V.
7. **Збережіть.** Натисніть Ctrl+S.

Не переплутайте порядок бітів. У двійковому слові A3A2A1A0 біт A0 має вагу 1, A1 — 2, A2 — 4, A3 — 8. Уводьте значення варіанта справа наліво: крайня права цифра переходить у A0.

14. Перевірка RCA4 за своїм варіантом

1. **Випишіть дані.** З основного завдання перепишіть A, B і Cin без зміни порядку бітів.
2. **Перейдіть до RCA4.** Клацніть RCA4 у Project Explorer і виберіть Poke Tool.
3. **Задайте A.** Для слова A3A2A1A0 встановіть відповідні значення на пінах A3, A2, A1, A0.
4. **Задайте B.** Так само встановіть B3, B2, B1, B0.
5. **Задайте Cin.** Клацніть Cin, щоб встановити 0 або 1 відповідно до варіанта.
6. **Зчитайте результат.** Запишіть S3S2S1S0, Cout і V. Читайте результат від S3 до S0.
7. **Зробіть ручну перевірку.** Переведіть A і B у десяткову систему, обчисліть A+B+Cin і порівняйте молодші 4 біти з S3S2S1S0.
8. **Перевірте Cout.** Якщо беззнакова сума більша за 15, Cout має дорівнювати 1.
9. **Перевірте V.** Для знакової інтерпретації діапазон 4-бітного числа у додатковому коді дорівнює від −8 до +7. V=1 означає вихід математичного результату за цей діапазон.
10. **Збережіть доказ.** Зробіть знімок екрана з установленими входами й видимими виходами.

Що записати	Форма запису
Вхідні слова	A = A3A2A1A0; B = B3B2B1B0; Cin = …
Результат схеми	S = S3S2S1S0; Cout = …; V = …
Беззнакова перевірка	$A_{10} + B_{10} + Cin = …$
Знакова перевірка	A_signed + B_signed + Cin = …; чи входить результат у −8…+7

15. Створення керованого суматора-віднімача ADD_SUB4

Сигнал SUB керує операцією. За SUB=0 вентилі XOR передають біти B без зміни, а C0=0, тому схема додає A+B. За SUB=1 кожний B_i інвертується і до молодшого розряду додається 1, тому схема виконує A + побітово інвертоване B + 1 = A−B.

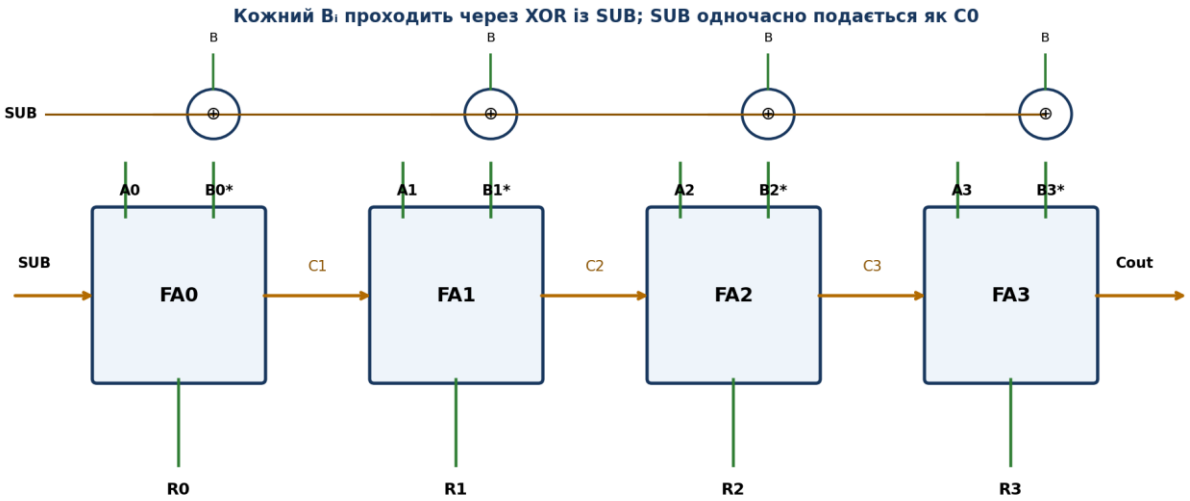

Рисунок 5. Керування суматором-віднімачем сигналом SUB

15.1. Розмістіть компоненти

1. **Створіть схему.** Project → Add Circuit... → введіть ADD_SUB4 → OK.
2. **Додайте входи A.** Створіть A0, A1, A2, A3 як одноразрядні Input Pins.
3. **Додайте входи B.** Створіть B0, B1, B2, B3 як одноразрядні Input Pins.
4. **Додайте SUB.** Створіть одноразрядний Input Pin з Label = SUB.
5. **Додайте виходи.** Створіть Output Pins R0, R1, R2, R3, Cout, Borrow і V.
6. **Розмістіть чотири FA.** Поставте чотири екземпляри FA зліва направо: FA0, FA1, FA2, FA3.
7. **Розмістіть XOR для B.** Додайте чотири Gates → XOR Gate. Для кожного встановіть Number Of Inputs = 2, Data Bits = 1, Facing = East.
8. **Додайте XOR для V.** Додайте ще один двовходовий XOR біля FA3.
9. **Додайте NOT і AND для Borrow.** Додайте Gates → NOT Gate і Gates → AND Gate з двома входами.

15.2. Підключіть керовані біти B

1. **Підключіть B0.** B0 подайте на перший XOR. Другий вхід цього XOR з'єднайте з SUB. Вихід XOR подайте на B-вхід FA0.
2. **Повторіть для B1.** B1 → XOR із SUB → B-вхід FA1.
3. **Повторіть для B2.** B2 → XOR із SUB → B-вхід FA2.
4. **Повторіть для B3.** B3 → XOR із SUB → B-вхід FA3.

5. **Зробіть спільну лінію SUB.** Проведіть одну горизонтальну лінію від SUB і створіть від неї чотири відгалуження до XOR. Перевірте наявність вузлів у місцях приєднання.

15.3. Підключіть суматори та ознаки

1. **Підключіть A.** Подайте A0, A1, A2, A3 безпосередньо на A-входи відповідних FA.
2. **Задайте C0.** Підключіть SUB безпосередньо до Cin блока FA0.
3. **Створіть ланцюг перенесення.** Cout FA0 → Cin FA1; Cout FA1 → Cin FA2; Cout FA2 → Cin FA3.
4. **Підключіть результат.** Виходи S блоків FA0...FA3 подайте відповідно на R0...R3.
5. **Підключіть Cout.** Cout FA3 подайте на зовнішній вихід Cout.
6. **Підключіть V.** Cout FA2, тобто C3, подайте на перший вхід XOR для V; Cout FA3, тобто C4, — на другий. Вихід XOR подайте на V.
7. **Сформуйте Borrow.** Cout FA3 подайте на NOT. Вихід NOT подайте на один вхід AND, а SUB — на другий. Вихід AND подайте на Borrow. Таким чином Borrow активний лише в режимі віднімання.
8. **Збережіть.** Натисніть Ctrl+S.

> **Чому Borrow не дорівнює Carry.** Під час віднімання за додатковим кодом Cout=1 означає, що позика не потрібна. Тому за SUB=1 сигнал Borrow є інверсією Cout. Додаткове AND із SUB примусово встановлює Borrow=0 у режимі додавання.

16. Перевірка ADD_SUB4

16.1. Швидка перевірка режиму додавання

1. **Установіть SUB=0.** У цьому режимі B не інвертується, а C0=0.
2. **Задайте простий приклад.** Установіть A=0011 і B=0101.
3. **Звірте результат.** Має бути R=1000, Cout=0, Borrow=0. Для знакової інтерпретації 3+5=8 не вміщується у діапазон −8...+7, тому V=1.

16.2. Перевірка віднімання за своїм варіантом

1. **Перепишіть X і Y.** У схемі ADD_SUB4 використовуйте X як слово A, а Y як слово B.
2. **Установіть SUB=1.** Це одночасно інвертує всі біти Y і подає 1 на Cin FA0.
3. **Задайте X.** Встановіть X3, X2, X1, X0 на пінах A3, A2, A1, A0.
4. **Задайте Y.** Встановіть Y3, Y2, Y1, Y0 на пінах B3, B2, B1, B0.
5. **Зчитайте виходи.** Запишіть R3R2R1R0, Cout, Borrow і V.
6. **Перевірте Borrow.** Для беззнакових чисел Borrow=1, якщо X<Y; Borrow=0, якщо X≥Y.
7. **Інтерпретуйте від'ємний результат.** Якщо R3=1 і потрібне знакове значення, обчисліть R_unsigned−16. Наприклад, 1101_2=13, а як 4-бітне signed число це 13−16=−3.
8. **Збережіть доказ.** Зробіть знімок екрана з SUB=1, усіма входами та виходами.

17. Індивідуальні варіанти

Для додавання в схемі RCA4 використовуйте A, B і Cin зі свого рядка. Для віднімання в схемі ADD_SUB4 встановіть A=X, B=Y і SUB=1. Усі слова мають чотири біти; крайній правий біт є молодшим.

№	A	B	Cin	X	Y для X−Y
1	0101	0011	0	0111	0011
2	1001	0110	0	0011	0111
3	1110	0011	0	1010	0101
4	0111	0001	0	1000	0011
5	1101	1110	0	0100	1001
6	1011	0101	1	0110	0010
7	0011	1100	0	0010	0101
8	1111	0001	0	0111	1110
9	1000	1000	0	1001	0011
10	0110	0101	1	0101	1010
11	0010	1111	0	1100	0011
12	1010	0111	0	0011	1000
13	0100	1011	1	0110	0111
14	1100	0010	0	0010	1100
15	0001	0111	1	1000	0001
16	0111	0111	0	0101	1011
17	1110	1111	1	0011	1101

18. Як підставити біти варіанта

Приклад нижче показує лише спосіб введення бітів. У звіті використовуйте числа свого варіанта.

Слово	Запис	Значення пінів
A	1011	A3=1, A2=0, A1=1, A0=1

Слово	Запис	Значення пінів
B	0110	B3=0, B2=1, B1=1, B0=0
Cin	1	Cin=1

1. **Починайте зі старшого біта.** Ліва цифра слова 1011 є A3.
2. **Закінчуйте молодшим бітом.** Права цифра слова 1011 є A0.
3. **Не читайте піни за положенням на екрані.** Орієнтуйтеся на мітки A3...A0, а не лише на те, який пін розташований вище або нижче.
4. **Після введення перевірте слово.** Вголос або письмово прочитайте A3A2A1A0 і порівняйте з варіантом.

19. Що має бути у звіті

- Титульна частина, тема, мета та номер варіанта.
- Таблиця істинності повного суматора та формули S і Cout.
- Знімок перевіреної схеми HA.
- Знімок перевіреної схеми FA.
- Схема RCA4 та результат A+B+Cin за варіантом: S, Cout і V.
- Схема ADD_SUB4 та результат X−Y за варіантом: R, Cout, Borrow і V.
- Ручна перевірка обох арифметичних операцій.
- Короткий висновок про призначення Cin, Cout, Borrow, V і SUB.
- Файл проєкту LR4_Прізвище_Група.circ.

20. Контрольний перелік перед здаванням

Перевірка	Позначка
У Help → About вказано Logisim Evolution 5.0.0	☐
Gate shape встановлено ANSI	☐
HA проходить 4 тестові комбінації	☐
FA проходить 8 тестових комбінацій	☐
У RCA4 FA0 працює з молодшими бітами A0 і B0	☐
У RCA4 Cout кожного FA підключений до Cin наступного	☐
V підключено як XOR перенесень C3 і C4	☐
У ADD_SUB4 SUB підключено до всіх XOR для B та до Cin FA0	☐

Перевірка	Позначка
Borrow формується лише для SUB=1	☐
Результати свого варіанта збігаються з ручними обчисленнями	☐
Файл .circ збережено і повторно відкривається	☐

21. Типові помилки та спосіб виправлення

Ознака	Ймовірна причина	Що зробити
Вихід синій або невизначений	Контакт не підключений або Auto-Propagate вимкнено	Перевірити кінці дротів; увімкнути Simulate → Auto-Propagate.
Дріт червоний	Два виходи з'єднані разом або виник конфлікт сигналів	Перевірити, чи дріт іде від виходу до входу, а не між двома виходами.
Дріт помаранчевий	Не збігається Data Bits	Для всіх елементів цієї роботи встановити Data Bits = 1.
Пін не перемикається	Обрано Edit Tool або пін має Type = Output	Вибрати Poke Tool; для вхідного піна встановити Type = Input.
HA дає неправильний C	Для C використано OR замість AND	Замінити вентиль на AND Gate.
FA правильний не для всіх рядків	Переплутано виходи S і C підсхем HA	Звірити з рисунком 3 і перевірити кожне з'єднання.
Сума виглядає у зворотному порядку	Переплутано біти 0 і 3	Пам'ятати: індекс 0 — молодший біт; читати результат S3S2S1S0.
Віднімання дає A+B	SUB=0 або не підключено до XOR	Установити SUB=1 і перевірити відгалуження до всіх чотирьох XOR.
Віднімання відрізняється на 1	SUB не підключено до Cin FA0	Підключити SUB до молодшого перенесення C0.
Borrow інвертований	Borrow підключено прямо до Cout	У режимі віднімання використати інверсію Cout; для спільної схеми — AND із SUB.
V не відповідає знаковому результату	Використано лише Cout	Підключити V=C3⊕C4, де C3 входить у FA3, а C4 виходить із FA3.

22. Як правильно завершити роботу

1. Збережіть проєкт. Натисніть Ctrl+S.

2. **Перевірте повторне відкриття.** Закрийте файл через File → Close, потім File → Open... і відкрийте свій .circ.
3. **Перевірте всі схеми.** Клацніть HA, FA, RCA4 і ADD_SUB4 у Project Explorer; переконайтеся, що компоненти та дроти на місці.
4. **Збережіть знімки.** На знімку мають бути видимі назва схеми, входи, виходи і результат тесту.
5. **Передайте два матеріали.** Завантажте звіт у потрібному форматі та окремо файл .circ, якщо це передбачено вимогами викладача.

23. Короткий словник інтерфейсу

Назва в програмі	Українське пояснення
File	Файл: створення, відкриття і збереження проєкту.
Project	Проєкт: додавання та керування схемами.
Simulate	Моделювання: запуск, скидання та покрокове поширення сигналів.
Pin	Контакт; може бути входом або виходом схеми.
Poke Tool	Інструмент взаємодії для зміни значень входів.
Edit Tool	Інструмент вибору і редагування.
Wiring Tool	Інструмент прокладання дротів.
Facing	Напрямок компонента.
Data Bits	Кількість бітів сигналу.
Label	Текстова мітка компонента.
Number Of Inputs	Кількість входів вентиля.
Auto-Propagate	Автоматичне поширення сигналів під час зміни входів.
Input / Output	Тип контакту: вхід / вихід.

24. Офіційні джерела

1. Репозиторій і сторінка завантажень Logisim Evolution: github.com/logisim-evolution/logisim-evolution

2. Випуск Logisim Evolution 5.0.0 від 12 вересня 2026 року: github.com/logisim-evolution/logisim-evolution/releases/tag/v5.0.0

Перевірено для версії 5.0.0. Назви меню та властивостей у цій інструкції наведено для англійської мови інтерфейсу.