

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ВНУТРІШНІХ СПРАВ**

Кафедра протидії кіберзлочинності факультету № 4

**МЕТОДИЧНІ МАТЕРІАЛИ
ДО ЛАБОРАТОРНИХ ЗАНЯТЬ**

навчальної дисципліни
«Електроніка та схемотехніка»
обов'язкових компонент освітньої програми
першого (бакалаврського) рівня вищої освіти

**125 Кібербезпека та захист інформації
(безпека інформаційних та комунікаційних систем)**

Харків 2023

ЗАТВЕРДЖЕНО

Науково-методичною радою
Харківського національного
університету внутрішніх справ
Протокол від 30.08.2023 № 7

СХВАЛЕНО

Вченою радою факультету № 6
Протокол від 16.08.2023 № 8

ПОГОДЖЕНО

Секцією Науково-методичної ради
ХНУВС з технічних дисциплін
Протокол від 29.08.2023 № 7

Розглянуто на засіданні кафедри протидії кіберзлочинності факультету № 4
(протокол від 15.08.2023 № 19)

Розробники:

1. Доцент кафедри протидії кіберзлочинності факультету № 4, кандидат технічних наук, доцент Клімушин П.С.
2. Доцент кафедри кібербезпеки та DATA-технологій факультету № 6, кандидат технічних наук, доцент Тулупов В.В.

Рецензенти:

1. Завідувач кафедри інформаційних управляючих систем ХНУРЕ, д.т.н., професор Петров К. Е.
2. Провідний науковий співробітник Науково-дослідної лабораторії з проблем розвитку інформаційних технологій ХНУВС, к.т.н., доцент Мордвинцев М.В.

Зміст

ЛАБОРАТОРНЕ ЗАНЯТТЯ №1. ДОСЛІДЖЕННЯ НАПІВПРОВІДНИКОВИХ ВИПРЯМЛЯЧІВ.....	5
1. ПРИНЦИПИ ПОБУДОВИ ОДНОПІВПЕРІОДНОГО ВИПРЯМЛЯЧА	6
2. ПРИНЦИПИ ПОБУДОВИ ДВОПІВПЕРІОДНОГО ВИПРЯМЛЯЧА	7
3. КОНТРОЛЬНІ ПИТАННЯ	10
ЛАБОРАТОРНЕ ЗАНЯТТЯ №2. ДОСЛІДЖЕННЯ ДИФЕРЕНЦІЮЮЧИХ ТА ІНТЕГРУЮЧИХ КОЛІВ ІМПУЛЬСНОЇ ЕЛЕКТРОНІКИ	11
1. ДОСЛІДЖЕННЯ ДІЛЬНИКА НАПРУГ	11
2. ДОСЛІДЖЕННЯ ДИФЕРЕНЦІЮЮЧИХ RC-КІЛ	13
3. ДОСЛІДЖЕННЯ ІНТЕГРУЮЧИХ RC-КІЛ	17
4. КОНТРОЛЬНІ ПИТАННЯ	18
ЛАБОРАТОРНЕ ЗАНЯТТЯ №3. ДОСЛІДЖЕННЯ ЛОГІЧНИХ ЕЛЕМЕНТІВ ТА ЛОГІЧНИХ ФУНКЦІЙ.....	18
1. РОЗВ’ЯЗАННЯ ЗАВДАНЬ ЩОДО МІНІМІЗАЦІЇ ЛОГІЧНИХ ФУНКЦІЙ.....	18
2. ІНДИВІДУАЛЬНІ ЗАВДАННЯ НА ТЕМУ «ЛОГІЧНІ ОСНОВИ ЦИФРОВОЇ СХЕМОТЕХНІКИ»	22
3. ДОСЛІДЖЕННЯ ЛОГІЧНИХ СХЕМ З ДОПОМОГОЮ СЕРЕДОВИЩА МОДЕЛЮВАННЯ	30
4. ДОСЛІДЖЕННЯ ФУНКЦІОНУВАННЯ ЛОГІЧНИХ ЕЛЕМЕНТІВ У ДИНАМІЧНОМУ РЕЖИМІ РОБОТИ	32
5. КОНТРОЛЬНІ ПИТАННЯ	34
ЛАБОРАТОРНЕ ЗАНЯТТЯ №4. ДОСЛІДЖЕННЯ БІПОЛЯРНИХ І ПОЛЬОВИХ ТРАНЗИСТОРІВ	34
1. ЗАГАЛЬНІ ВІДОМОСТІ ПРО БІПОЛЯРНІ ТА ПОЛЬОВІ ТРАНЗИСТОРИ	35
2. ВИЗНАЧЕННЯ ЗАЛЕЖНОСТІ ВИХІДНОЇ НАПРУГИ ВІД ВХІДНОЇ ДЛЯ СХЕМИ ІЗ ЗАГАЛЬНИМ ЕМІТЕРОМ	36
3. ВИМІРЮВАННЯ КОЕФІЦІЄНТА ПІДСИЛЕННЯ СХЕМИ ІЗ ЗАГАЛЬНИМ ЕМІТЕРОМ І НЕГАТИВНИМ ЗВОРОТНИМ ЗВ’ЯЗКОМ ПО СТРУМУ	36
4. ВИЗНАЧЕННЯ ЗАЛЕЖНОСТІ ВИХІДНОЇ НАПРУГИ ВІД ВХІДНОЇ ДЛЯ СХЕМ ІЗ ЗАГАЛЬНОЮ БАЗОЮ І ЗАГАЛЬНИМ КОЛЕКТОРОМ.....	37
5. ТРАНЗИСТОР ЯК ДЖЕРЕЛО СТАБІЛЬНОГО СТРУМУ.....	37
6. ВИЗНАЧЕННЯ ЗАЛЕЖНОСТІ ВИХІДНОЇ НАПРУГИ ВІД ВХІДНОЇ ДЛЯ СХЕМИ ІЗ ЗАГАЛЬНИМ ІСТОКОМ ТА ІСТОКОВОГО ПОВТОРЮВАЧА	37
7. КЕРОВАНИЙ ДІЛЬНИК НАПРУГИ	38
8. КОНТРОЛЬНІ ПИТАННЯ	38
ЛАБОРАТОРНЕ ЗАНЯТТЯ №5. ДОСЛІДЖЕННЯ ФУНКЦІОНУВАННЯ ТРИГЕРІВ.....	38
1. АНАЛІЗ ФУНКЦІОНУВАННЯ RS-ТРИГЕРІВ.....	39
2. АНАЛІЗ ФУНКЦІОНУВАННЯ JK-ТРИГЕРІВ	42
3. АНАЛІЗ ФУНКЦІОНУВАННЯ T-ТРИГЕРІВ.....	42
4. ДОСЛІДЖЕННЯ РОБОТИ ТРИГЕРІВ У СТАТИЧНОМУ РЕЖИМІ	43
5. КОНТРОЛЬНІ ПИТАННЯ	44
ЛАБОРАТОРНЕ ЗАНЯТТЯ №6. ДОСЛІДЖЕННЯ ФУНКЦІОНУВАННЯ ШИФРАТОРІВ, ДЕШИФРАТОРІВ, МУЛЬТИПЛЕКСОРІВ, ДЕМУЛЬТИПЛЕКСОРІВ	44
1. ПОБУДОВА ДЕШИФРАТОРІВ ТА ШИФРАТОРІВ	45
2. ДОСЛІДЖЕННЯ ФУНКЦІОНУВАННЯ МУЛЬТИПЛЕКСОРІВ, ДЕШИФРАТОРІВ	47
3. КОНТРОЛЬНІ ПИТАННЯ	51
ЛАБОРАТОРНЕ ЗАНЯТТЯ №7. ДОСЛІДЖЕННЯ ФУНКЦІОНУВАННЯ КОДОПЕРЕТВОРЮВАЧІВ ТА КОМБІНАЦІЙНИХ СУМАТОРІВ.....	51
1. ПОБУДОВА КОДОПЕРЕТВОРЮВАЧІВ ДВІЙКОВОГО КОДУ	52
2. ДОСЛІДЖЕННЯ ФУНКЦІОНУВАННЯ КОМБІНАЦІЙНИХ СУМАТОРІВ	53
3. КОНТРОЛЬНІ ПИТАННЯ	54
ЛАБОРАТОРНЕ ЗАНЯТТЯ №8. ДОСЛІДЖЕННЯ ФУНКЦІОНУВАННЯ РЕГІСТРІВ, ЛІЧИЛЬНИКІВ.....	54
1. ПОБУДОВА ТИПОВИХ ЦИФРОВИХ ВУЗЛІВ: РЕГІСТРІВ, ЛІЧИЛЬНИКІВ.....	55

2. Дослідження функціонування РЕГІСТРА.....	56
3. Дослідження функціонування ЛІЧІЛЬНИКІВ	57
4. КОНТРОЛЬНІ ПИТАННЯ	57
ЛАБОРАТОРНЕ ЗАНЯТТЯ №9. ДОСЛІДЖЕННЯ ПОСТІЙНОЇ ТА ОПЕРАТИВНОЇ ПАМ'ЯТІ КОМП'ЮТЕРА	58
1. Дослідження функціонування модуля ОПЕРАТИВНОЇ ПАМ'ЯТІ.....	58
2. Дослідження функціонування модуля ПОСТІЙНОЇ ПАМ'ЯТІ.....	60
3. КОНТРОЛЬНІ ПИТАННЯ	61
ЛАБОРАТОРНЕ ЗАНЯТТЯ № 10 ДОСЛІДЖЕННЯ ЗАСОБІВ ІНДИКАЦІЇ МПС	61
1 ХАРАКТЕРИСТИКА СИМУЛЯТОРА EdSim51 SIMULATOR	62
2 СПОСОБИ ФОРМУВАННЯ ЧАСОВИХ ІНТЕРВАЛІВ	67
3 ПРОГРАМУВАННЯ ОПЕРАЦІЙ ЧАСОВОЇ ЗАТРИМКИ	68
4 ПОРЯДОК ВИКОНАННЯ РОБОТИ.....	70
5 КОНТРОЛЬНІ ЗАПИТАННЯ	70
ЛАБОРАТОРНЕ ЗАНЯТТЯ №11. ДОСЛІДЖЕННЯ АЛГОРИТМУ ДОДАВАННЯ БАГАТОБАЙТОВИХ ЧИСЕЛ.....	71
1 АЛГОРИТМ ДОДАВАННЯ БАГАТОБАЙТОВИХ ЧИСЕЛ.....	72
2 ПРОГРАМУВАННЯ ОПЕРАЦІЙ ДОДАВАННЯ ОПЕРАНДІВ	74
3 АЛГОРИТМ ВІДНІМАННЯ БАГАТОБАЙТОВИХ ЧИСЕЛ	77
4 ПРОГРАМУВАННЯ ОПЕРАЦІЙ ВІДНІМАННЯ ОПЕРАНДІВ.....	79
5 ПОРЯДОК ВИКОНАННЯ РОБОТИ.....	80
5. КОНТРОЛЬНІ ПИТАННЯ	82
ЛАБОРАТОРНЕ ЗАНЯТТЯ №12. ДОСЛІДЖЕННЯ ЦИФРО-АНАЛОГОВИХ ТА АНАЛОГО-ЦИФРОВИХ ПЕРЕТВОРЮВАЧІВ... ..	82
1. ОСНОВНІ ХАРАКТЕРИСТИКИ ЦАП І АЦП	83
2. ДОСЛІДЖЕННЯ ЦИФРО-АНАЛОГОВОГО ПЕРЕТВОРЮВАЧА ІЗ ДВІЙКОВО-ЗВАЖЕНИМИ ОПОРАМИ	84
3. ДОСЛІДЖЕННЯ ЦАП НА ОСНОВІ МАТРИЦІ R-2R.....	86
4. ДОСЛІДЖЕННЯ РОБОТИ МІКРОСХЕМ АЦП	87
5. КОНТРОЛЬНІ ПИТАННЯ	88

1. Розподіл часу навчальної дисципліни за темами

Номер та назва навчальної теми	Кількість годин відведених на вивчення навчальної дисципліни					Вид контролю	
	Всього	з них:					
		лекції	Семінарські заняття	Практичні заняття	Лабораторні заняття		Самостійна робота
Семестр № 4							
Тема № 1. Матеріали та компоненти електронної техніки	12	2			4	6	
Тема № 2. Диференціюючі та інтегруючі кола імпульсної електроніки	12	2			4	6	
Тема № 3. Логічні основи цифрової схемотехніки	18	2			6	8	
Тема № 4. Схемотехніка логічних елементів	16	4			4	8	
Тема № 5. Схемотехніка цифрових елементів	16	4			4	8	
Тема № 6. Схемотехніка комбінаційних вузлів	16	6			4	8	
Всього за семестр № 4:	90	20			26	44	залік
Семестр № 5							
Тема № 6. Схемотехніка комбінаційних вузлів	16	2			4	8	
Тема № 7. Схемотехніка цифрових вузлів	16	4			4	8	
Тема № 8. Напівпровідникові запам'ятовуючі пристрої	16	4			4	8	
Тема № 9. Основи мікропроцесорної техніки	32	8			8	16	
Тема № 10. Цифро-аналогові і аналого-цифрові перетворювачі	12	2			4	6	
Всього за семестр № 5:	90	20			24	46	екзамен
Всього по дисципліні:	180	40			50	90	

2. Методичні вказівки до лабораторних занять

Тема 1. Матеріали та компоненти електронної техніки

Лабораторне заняття №1. Дослідження напівпровідникових випрямлячів

Навчальна мета заняття:

1. Ознайомитися із принципом дії і основними характеристиками фільтрів, що згладжують.

Кількість годин: 4 год.

Література:

1. Методичні вказівки до лабораторних робіт з курсу «Моделювання на ЕОМ» для студентів напряму підготовки 6.051001 «Метрологія та

інформаційно вимірювальні технології» денного та заочного навчання / Уклад. Харченко О. Л., Балєв В. М. Харків : НТУ «ХП», 2010. 68 с.

2. Крилик Л.В., Селецька О.О. Матеріали електронної техніки: навчальний посібник. Вінниця: ВНТУ, 2017. 120 с

3. Чешко І.В. Вступ до спеціальності «Електроніка» : навчальний посібник. Суми : Сумський державний університет, 2017. 148 с.

4. Квітка С.О., Яковлев В.Ф., Нікітіна О.В. Електроніка та мікросхемотехніка: Навчальний посібник / За ред. проф. В.Ф. Яковлева. Київ: Аграрна освіта, 2010. 329 с.

5. Болюх В. Ф., Данько В. Г. Основи електроніки і мікропроцесорної техніки: Навч. посібник. Харків: НТУ «ХП», 2011. 257 с.

Матеріально-технічне забезпечення: комп'ютерна мережа із системою модулювання електронних схем Electronics Workbench.

Навчальні питання

1. Принципи побудови однопівперіодного випрямляча
2. Принципи побудови двопівперіодного випрямляча
3. Контрольні питання

1. Принципи побудови однопівперіодного випрямляча

Для перетворення змінної напруги в постійну напругу застосовують випрямні пристрої. У випрямний пристрій звичайно входять трансформатор, один або кілька діодів, фільтр, що згладжує, електронний стабілізатор постійної напруги. Залежно від умов роботи окремі елементи випрямного пристрою можуть бути відсутніми.

Вибір тієї або іншої схеми джерела напруги вторинного живлення обумовлений параметрами живильної мережі, вимогами до вихідних електричних параметрів, конструктивними особливостями пристрою, температурним діапазоном роботи, терміном служби, гарантованою надійністю і переліком дозволених до застосування елементів.

У більшості випадків для живлення вимірювальних приладів використовується однофазна мережа, з діючим значенням напруги 220 В, 50 Гц.

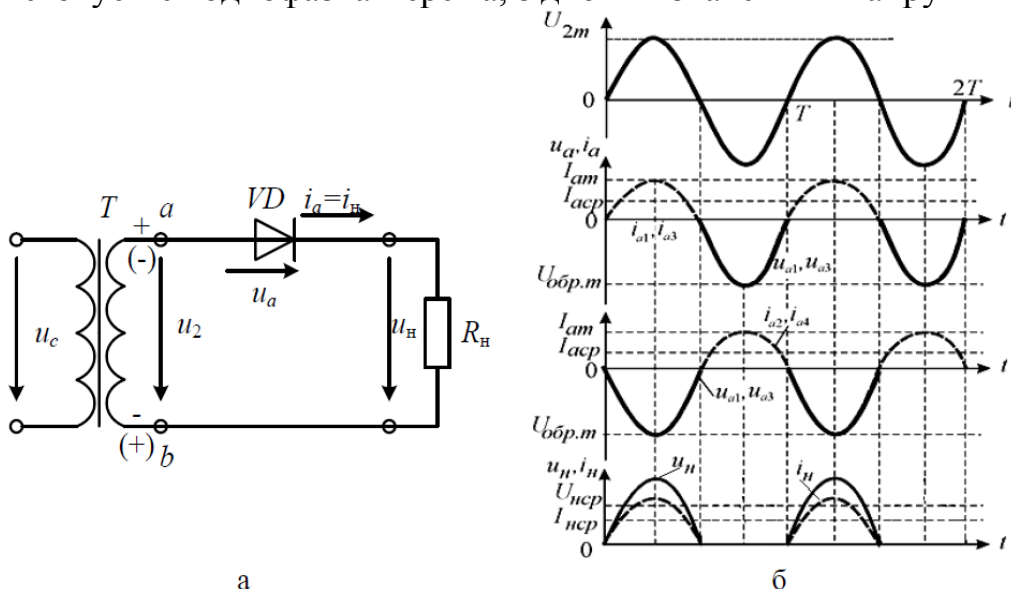


Рис. 1. Схема (а) і часові діаграми (б) однопівперіодного випрямляча

Схема однопівперіодного випрямляча наведена на рис. 1,а. Для спрощення аналізу будемо вважати діод ідеальним, тобто будемо думати, що його опір в прямому напрямку дорівнює нулю, а у зворотному – нескінченності. Тоді протягом першого півперіоду вхідної напруги, коли на аноді діода VD буде позитивний щодо катода потенціал, діод буде відкритий.

Напруга u_2 на вторинній обмотці трансформатора буде безпосередньо прикладена до навантаження R_n і у ній виникне струм i_n (рис. 1,б), що буде повторювати форму напруги на вторинній обмотці трансформатора. Під час другого півперіоду вхідної напруги на аноді діода VD буде негативний щодо катода потенціал, діод буде закритий, а струм у навантаженні виявиться рівним нулю.

Постійна складова вихідної напруги однопівперіодної схеми випрямляча:

$$u_{н.ср.} = u_0 = \frac{1}{2\pi} \int_0^{\pi} U_{2m} \sin \omega t dt = \frac{U_{2m}}{\pi} = \frac{\sqrt{2}u_2}{\pi},$$

де u_2 – діюче значення напруги на вторинній обмотці трансформатора.

Зі змінних складових максимальну амплітуду U_{1m} має складова з частотою ω , що може бути легко знайдена з розкладання вихідної напруги в ряд Фур'є. В однопівперіодній схемі випрямляча $U_{1m} = 1,57u_0$, тобто пульсації вихідної напруги великі, що є її істотним недоліком.

При проектуванні однопівперіодних випрямлячів важливо правильно вибрати тип діода, що задовільно працював би в такій схемі. Цей вибір проводять на основі двох міркувань. По-перше, припустимий струм діода повинен перевищувати величину I_m . По-друге, діод повинен мати певну електричну міцність.

Останнє пов'язане з тим, що протягом тих півперіодів, коли діод закритий, до нього прикладена напруга, рівна напрузі на вторинній обмотці трансформатора, причому ця напруга має зворотну для діода полярність («мінус» на аноді). Максимальна величина цієї напруги, називана зворотною напругою $U_{обр}$, у нашому випадку дорівнює амплітуді напруги на вторинній обмотці трансформатора, тобто в однопівперіодній схемі випрямляча

$$U_{обр} = U_{2m},$$

а припустима зворотна напруга діода повинна бути більше U_{1m} .

2. Принципи побудови двопівперіодного випрямляча

Один з варіантів двопівперіодного випрямляча, зібраного за мостовою схемою, наведений на рис. 2,а. Тут змінну напругу підводять до однієї діагоналі моста, а випрямлена напруга знімається з іншої.

Розглянемо роботу схеми. Нехай у деякий момент часу змінна напруга на вторинній обмотці трансформатора така, що потенціал точки а вище потенціалу точки в. Тоді від точки а («+» джерела напруги) струм буде проходити через діод VD1 до точки з, далі через навантаження до точки б і через діод VD3 до точки в («-» джерела напруги). Протягом наступного півперіоду, коли потенціал точки в вище потенціалу точки а, струм від точки в буде проходити через діод VD4 навантаження і діод VD2 до точки а. Для першого півперіоду напрям струму показаний суцільними стрілками, для другого півперіоду – пунктирними стрілками. У будь-який півперіод струм через навантаження проходить в одному напрямку.

Часові діаграми напруг і струмів для мостової схеми випрямляча наведені на рис. 2,б. Відношення діючого значення напруги вторинної обмотки до середнього значення випрямленої напруги дорівнює коефіцієнту форми синусоїдальної напруги, тому

$$u_2 = \frac{\pi}{2\sqrt{2}} u_n \approx 1,1 u_n.$$

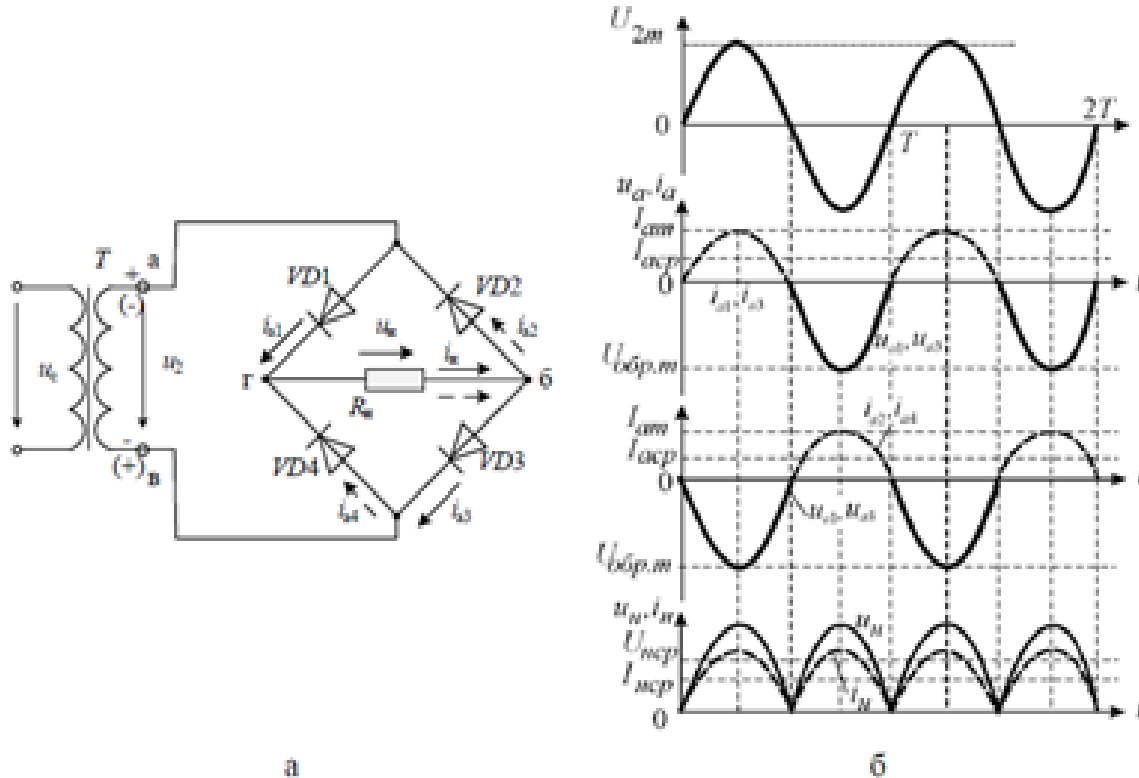


Рис. 2. Схема (а) і часові діаграми (б) двопівперіодного випрямляча

Максимальне значення зворотної напруги на вентилі дорівнює амплітудному значенню прикладеної напруги, тому що в один з півперіодів, коли струм проходить через діоди 1 і 3, діоди 2 і 4 виявляються включеними паралельно і до них прикладена напруга u_2 , а в інший півперіод напруга u_2 прикладена до паралельно включених діодів 1 і 3. Таким чином,

$$U_{обр.м} = U_{2m} = \sqrt{2} u_2 = \frac{\pi}{2} u \approx 1,57 u_n.$$

Середнє значення струму вентиля $I_{аср} = 0,5 I_n$.

Ємнісний фільтр C_ϕ включають паралельно навантажувальному резистору R_n (рис. 3,а). При такому включенні конденсатор C_ϕ заряджається через діод до амплітудного значення напруги U_{2m} у момент часу, коли напруга u_2 на вторинній обмотці трансформатора перевищує напруга u_c на конденсаторі (рис. 3,б). Цьому режиму відповідає інтервал часу $t_1 - t_2$. Протягом інтервалу часу $t_2 - t_3$ напруга $u_c > u_2$, діод закритий, а конденсатор розряджається через навантажувальний резистор R_n з постійною часу $\tau_{роз} = C_\phi \cdot R_n$. При цьому напруга u_n знижується до деякого найменшого значення.

Починаючи з моменту часу t_3 , напруга u_c на конденсаторі стає менше напруги u_2 . Діод відкривається, конденсатор C_ϕ починає заряджатися, і процеси повторюються. Як показують часові діаграми (рис. 3,б), при включенні ємнісного

фільтра напруга u_n не зменшується до нуля, а пульсує в деяких межах, збільшуючи середнє значення випрямленої напруги.

Часові діаграми двопівперіодного мостового випрямляча з ємнісним фільтром (рис. 3,в) наведені на рис. 3,г. Ємність конденсатора C_ϕ вибирають такої величини, щоб для основної гармоніки випрямленої напруги опір конденсатора був багато менше R_n , тобто

$$\frac{1}{2\pi f_{\text{осн. гарм.}} C_\phi} \ll R_n \quad \text{або} \quad C_\phi \gg \frac{1}{2\pi f_{\text{осн. гарм.}} R_n}.$$

При такому виборі величини ємності конденсатора постійна часу розряду $\tau_{\text{роз}}$ значно більше періоду зміни випрямленої напруги

$$\tau_{\text{роз.}} = C_\phi R_n \gg \frac{1}{2\pi f_{\text{осн. гарм.}}} = T$$

і конденсатор C_ϕ розряджається порівняно повільно, тобто напруга на ньому зменшується несуттєво. Це приводить до збільшення середнього значення напруги на навантажувальному резисторі й, у порівнянні з величиною $u_{\text{нр}}$ при відсутності фільтра і зменшенню змінної складової, а, отже, до зниження коефіцієнта пульсацій.

При використанні ємнісних фільтрів варто мати на увазі, що імпульси струму при відкритому діоді визначаються опорами діода.

Такі стрибки струму можуть привести до виходу діода з ладу. Це особливо небезпечно для напівпровідникових і іонних електровакуумних діодів, тому що їхні опори при прямому включенні мають невелике значення. Для обмеження величини струму через діод послідовно з ним варто включати додатковий резистор.

Застосування ємнісного фільтра більш ефективно при високоомному навантажувальному резисторі, тому що випрямлена напруга і коефіцієнт згладжування мають більші величини, чим при низькоомному навантажувальному резисторі.

Оцінка дії, що згладжує, фільтра виконується за коефіцієнтом фільтрації p , під яким розуміють відношення коефіцієнта пульсації по основній гармоніці на вході фільтра ϵ_1 до коефіцієнта пульсації на його виході

$$p = \frac{\epsilon_1}{\epsilon}.$$

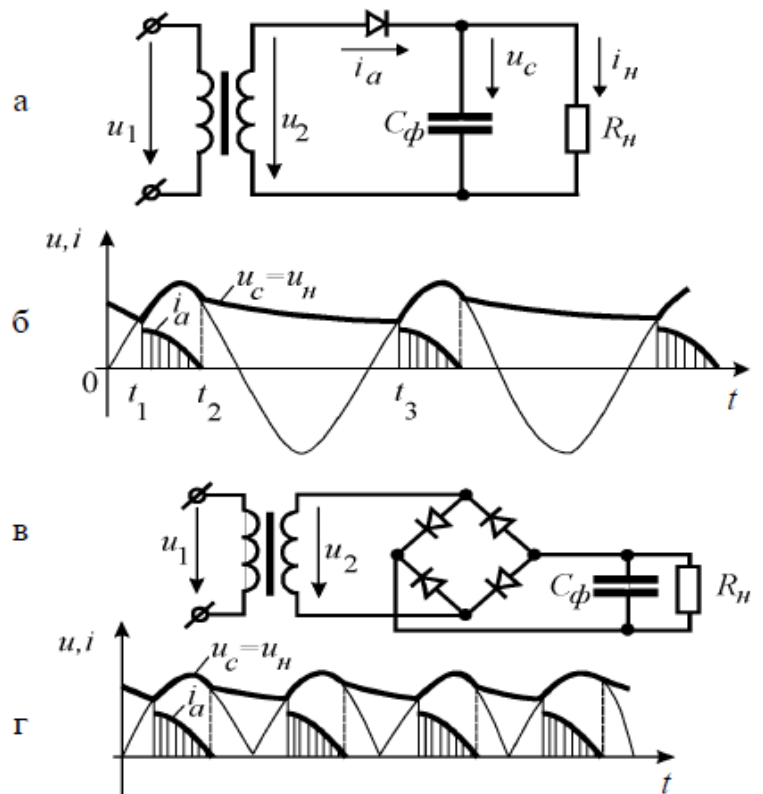


Рис. 3. Схеми ємнісних фільтрів з однопівперіодним (а) і мостовим (в) випрямлячами; часові діаграми напруг і струмів однопівперіодного (б) і мостового (г) випрямлячів з ємнісним фільтром

Для практичних розрахунків звичайно під коефіцієнтом пульсації розуміють відношення

$$\varepsilon = \frac{\Delta u_{m\sim}}{u_{cp}}.$$

де $\Delta u_{m\sim}$ – амплітуда змінної складової випрямленої напруги;

u_{cp} – середнє значення випрямленої напруги, рівне постійній складовій ряду Фур'є.

Приклад визначення коефіцієнта пульсації напруги на вході випрямляча з фільтром наведений на рис. 4.

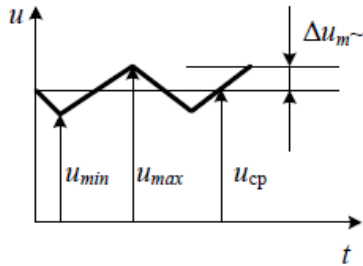


Рис. 4. Визначення пульсацій

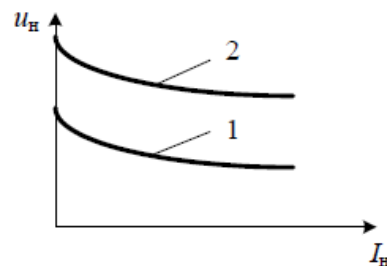


Рис. 5. Зовнішні характеристики

При роботі випрямляючого пристрою частина випрямленої напруги падає на активному опорі вторинної обмотки трансформатора і на прямому опорі відкритого діода. Отже, з ростом величини випрямленого струму i_{ncp} збільшується спадання напруги на цих опорах і напруга на навантажувальному пристрої u_n зменшується.

Залежність $u_n = f(i_n)$ називається зовнішньою характеристикою. Ця характеристика є однією з найважливіших характеристик випрямляючого пристрою. Опір відкритого діода залежить від величини струму, тому і залежність $u_n = f(i_n)$ нелінійна (рис. 5, крива 1). Якщо у випрямляючий пристрій включений фільтр, то залежність $u_n = f(i_n)$ зміниться. Ємнісному фільтру відповідає крива 2. Крім ємнісних фільтрів, для фільтрації випрямленої напруги можна використовувати індуктивність (дросель), що включають послідовно з опором навантаження.

3. Контрольні питання

1. Що таке випрямлячі і для чого вони потрібні?
2. Які види випрямлячів ви знаєте?
3. У чому відмінність між двопівперіодної із середньою точкою і мостовою схемами?
4. Як залежить напруга пульсацій випрямляча з фільтром від струму навантаження: напруга пульсацій падає, росте, залишається незмінною?
5. Чи однакові частоти вхідного і вихідного сигналів у схемах одно і двопівперіодного випрямлячів?
6. Які розходження між вхідним і вихідним сигналами одно і двопівперіодних випрямлячів?
7. Які фактори впливають на величину коефіцієнта пульсацій випрямляча з ємнісним фільтром на виході?

Тема 2. Фізичні основи надання інформації

Лабораторне заняття №2. Дослідження диференціюючих та інтегруючих кіл імпульсної електроніки

Навчальна мета заняття: Закріплення теоретичного матеріалу, набуття навиків створення і моделювання схем аналогових пристроїв, робота з різними вимірювальними приладами.

Кількість годин: 4 год.

Література:

1. Крилик Л.В., Селецька О.О. Матеріали електронної техніки: навчальний посібник. Вінниця: ВНТУ, 2017. 120 с

2. Чешко І.В. Вступ до спеціальності «Електроніка»: навчальний посібник. Суми: Сумський державний університет, 2017. 148 с.

3. Квітка С.О., Яковлев В.Ф., Нікітіна О.В. Електроніка та мікросхемотехніка: Навчальний посібник / За ред. проф. В.Ф. Яковлева. Київ: Аграрна освіта, 2010. 329 с.

4. Кравець В.О., Сокол Є.І., Рисований О.М. Комп'ютерна схемотехніка. Підручник. Харків: НТУ «ХПІ», 2007. 480 с.

Матеріально-технічне забезпечення: комп'ютерна мережа із системою модулювання електронних схем Electronics Workbench.

Навчальні питання

- 1 Дослідження дільника напруг
- 2 Дослідження диференціюючих RC-кіл
- 3 Дослідження інтегруючих RC-кіл
- 4 Контрольні питання

1. Дослідження дільника напруг

Дослідження простого дільника напруг за схемою рис. 1а, який виконано з використанням двох постійних резисторів R_1 і R_2 .

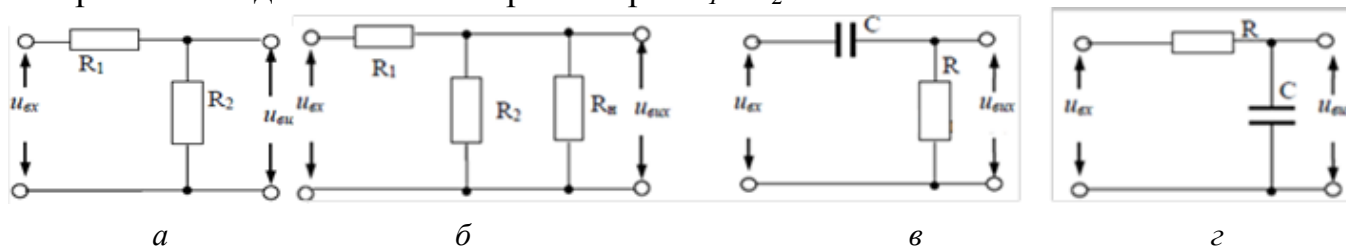


Рис. 1. Схеми дільників напруг: а - простого, б - з навантаженням, в - диференціюючого та г - інтегруючого RC-кола

Простий дільник напруги – це схема, яка для даної напруги на вході створює на виході напругу, яка є деякою частиною вхідної. У послідовно сполучених резисторах струм визначається таким чином:

$$I = U_{вх} / (R_1 + R_2).$$

Тоді для R_2

$$U_{вих} = IR_2 = U_{вх} R_2 / (R_1 + R_2).$$

З останньої формули видно, що $U_{вих} \leq U_{вх}$. З цієї формули згідно з варіантом до лабораторної роботи необхідно визначити номінал резистора, який є відсутнім в табл. 1.

№ з/п	$U_{вх},$ U	$U_{вих},$ U	$R_1,$ Ом	$R_2,$ Ом	$R_{н},$ Ом	№ з/п	$U_{вх},$ U	$U_{вих},$ U	$R_1,$ Ом	$R_2,$ Ом	$R_{н},$ Ом
1	1	0,2	100	-	500	9	25	10	200	-	100
2	2	0,3	-	100	$1 \cdot 10^3$	10	30	15	-	200	400
3	5	1	400	-	600	11	35	24	300	-	600
4	10	2	-	400	$1 \cdot 10^3$	12	40	20	-	300	900
5	12	2	$1 \cdot 10^3$	-	$2 \cdot 10^3$	13	45	25	$10 \cdot 10^3$	-	$1 \cdot 10^3$
6	15	5	-	$1 \cdot 10^3$	$1 \cdot 10^3$	14	50	20	-	$10 \cdot 10^3$	$5 \cdot 10^3$
7	17	7	$2 \cdot 10^3$	-	$1 \cdot 10^3$	15	110	20	$20 \cdot 10^3$	-	$10 \cdot 10^3$
8	20	5	-	$2 \cdot 10^3$	$4 \cdot 10^3$	16	220	25	-	$20 \cdot 10^3$	$1 \cdot 10^3$

Зробити заміри U_{R1} , U_{R2} і I_0 (де I_0 – струм дільника, що протікає через резистори R_1 і R_2). Зібрати і дослідити дільник напруг за схемою рис. 1б, що складається з двох постійних резисторів R_1 і R_2 (один із яких визначено в першому експерименті) та резистора навантаження R_n .

Зібрати і дослідити дільник напруг за схемою рис. 3а, що складається з двох постійних резисторів R_1 і R_2 (один із яких визначено в першому експерименті) та резистора навантаження R_n . Зробити заміри амплітудних значень $U_{вх}$, $U_{вих}$ і I_{R1} , I_{R2} і I_{Rn} (де I_{R1} , I_{R2} і I_{Rn} – струми, що протікають через відповідні резистори).

Завдання виконується згідно з варіантом із табл. 1.

Інструментальні засоби пакета EWB дозволяють проводити побудову електронних схем, дослідження (моделювання) їх роботи як у статичних так і динамічних режимах із застосуванням вимірювальних приладів (таких, як: універсальний генератор сигналів різної форми, осцилограф, вимірник частотних характеристик (плоттер), мультиметр, вольтметр, амперметр), а також джерел напруги і струму, зміни статичних і динамічних параметрів, частотних і часових характеристик досліджуваних схем (таких, як амплітудні значення струмів і напругу різ них точках досліджуваних схем, частота і період проходження вхідних і вихідних сигналів, активна тривалість фронтів і тривалості імпульсів, шпаруватість, амплітудно-частотні характеристики (АЧХ) і фазочастотні характеристики (ФЧХ)).


Порядок проведення роботи для розробки принципової електричної схеми. Запустіть Electronics Workbench. Підготуйте новий файл для роботи. Для цього необхідно виконати такі операції з меню: File/New і File/Save as. При виконанні операції Save as буде необхідно вказати ім'я файлу і каталог, в якому буде зберігатися схема. Рекомендується називати схему на прізвище виконавця.

Перенесіть необхідні елементи з заданої викладачем схеми на робочу область Electronics Workbench. Для цього необхідно вибрати розділ на панелі інструментів (Sources, Basic, Diodes, Transistors, Analog Ics, Mixed Ics, Digital Ics, Logic Gates, Digital, Indicators, Controls, Miscellaneous, Instruments), у якому знаходиться потрібний вам елемент, потім перенести його на робочу область.

З'єднайте контакти елементів і розташуйте елементи в робочій області для одержання необхідної вам схеми. Для з'єднання двох контактів необхідно натиснути на один із контактів кнопкою мишки і, не відпускаючи клавішу, довести курсор до другого контакту. У разі потреби можна використовувати додаткові вузли (розгалуження). Натисканням на елементі правою кнопкою мишки можна одержати швидкий доступ до найпростіших операцій, таких, як обертання (rotate), розворот (flip), копіювання/вирізання (copy/cut), вставка (paste).

Проставите необхідні номінали і властивості кожному елементу схеми. Для цього потрібно двічі натиснути мишкою на зображення елемента. Перед зміною параметрів слід відключати джерела живлення схеми, інакше можна отримати невірні результати.

Виберіть на панелі інструментів у нижньому рядку піктограму з накресленням елемента живлення. Якщо до неї підвести мишку, то відобразиться надпис Sources. При натисканні на цей елемент піктограма розкриється. Після чого виберіть елемент живлення та перенесіть його на поле програми.

Виберіть на панелі інструмента піктограму із зображення резистора  та перенесіть два рази зображення цього елемента на робоче поле.

Виберіть із піктограми Instrument два Multimeter та також перенесіть їх на поле монтажу елементів та з'єднайте їх таким чином, як наведено на рис.2.

Для зображення підписів елементів треба виділити елемент (натиснути ліву клавішу мишки). Виділений елемент повинен стати червоного кольору. Навести стрілку курсору на елемент та натиснути праву клавішу мишки. Вибрати підменю Component Properties. У ньому вибрати піктограму з назвою Label. Ввести назву елемента.

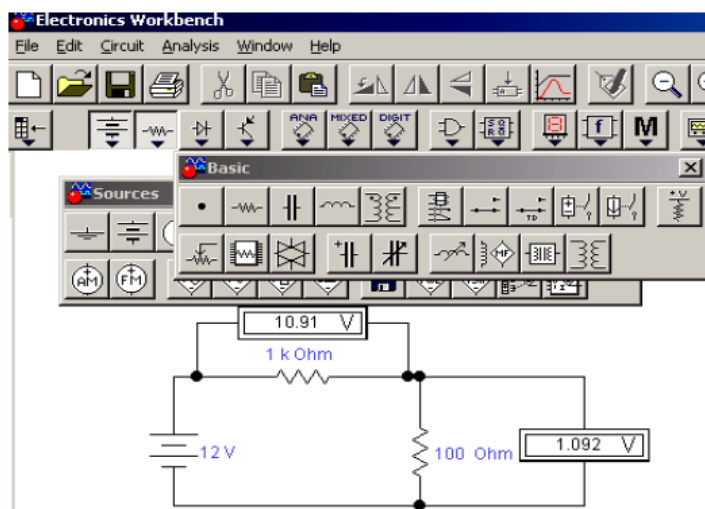


Рис. 2. Схема дослідження дільника напруг

Коли схема зібрана і готова до запуску, натисніть кнопку ввімкнення живлення на панелі інструментів. У разі серйозної помилки в схемі (замикання елемента живлення закоротко, відсутність нульового потенціалу в схемі) буде видане попередження.

Зробіть аналіз схеми, використовуючи інструменти індикації. Виведення терміналу здійснюється подвійним натисканням клавіші мишки на зображення елемента. У разі потреби можна скористатися кнопкою Pause.

При необхідності зробіть доступні аналізи в розділі меню Analysis.

2. Дослідження диференціюючих RC-кіл

Для рис. 1в виконати дослідження диференціюючого RC-кола. Варіанти завдання наведені в табл. 2.

Розрахувати постійні часу $\tau = RC$. Одержати і зобразити осцилограми з виходів елементів R та C .

При виконанні досліджень необхідно пам'ятати величини ємності:

$$1 \text{ Ф} = 10^{-3} \text{ мФ}; 1 \text{ Ф} = 10^{-6} \text{ мкФ}; 1 \text{ Ф} = 10^{-9} \text{ нФ}; 1 \text{ Ф} = 10^{-12} \text{ пФ}.$$

Наприклад, необхідно розрахувати диференціююче RC-коло при таких даних: $f = 10 \text{ кГц}$, $C = ?$, $R = 500 \text{ Ом}$. Для цього необхідно:

№ з/п	Частота f , Гц	Амп. $U_{вх}$	C , F	R , Ом	№ з/п	Частота f , Гц	Амп. $U_{вх}$	C , F	R , Ом
1	$1 \cdot 10^3$	5	$10 \cdot 10^{-9}$	-	9	$0,5 \cdot 10^3$	10	-	200
2	$0,5 \cdot 10^3$	10	$10 \cdot 10^{-10}$	-	10	$5 \cdot 10^3$	15	-	300
3	$10 \cdot 10^3$	15	$10 \cdot 10^{-10}$	-	11	$10 \cdot 10^3$	20	-	500
4	$20 \cdot 10^3$	20	$10 \cdot 10^{-9}$	-	12	$30 \cdot 10^3$	25	-	$1 \cdot 10^3$
5	$50 \cdot 10^3$	25	$50 \cdot 10^{-8}$	-	13	$50 \cdot 10^3$	30	-	$5 \cdot 10^3$
6	$4 \cdot 10^4$	30	$150 \cdot 10^{-9}$	-	14	$1 \cdot 10^4$	35	-	$10 \cdot 10^3$
7	$5 \cdot 10^4$	40	$75 \cdot 10^{-10}$	-	15	$10 \cdot 10^4$	40	-	750
8	200	10	$10 \cdot 10^{-10}$	-	16	$10 \cdot 10^5$	20	-	150

а) визначити період проходження імпульсів:

$$T = 1/f = 1/10 \times 10^3 = 0,1 \times 10^{-3} \text{ C};$$

б) визначити величину, яка є зворотною шпаруватості $Q = T/t_i$. Ця величина потрібна для установки в програмі EWB з позначкою на генераторі сигналів Duty cycle. Обчислювана величина тільки в цій програмі обчислюється в процентах.

$$1/Q = t_i/T = 50 \%$$

З цього випливає:

$$t_i = (50\%/100\%) \times T = 0,5 \times 0,1 \times 10^{-3} = 0,05 \times 10^{-3} \text{ C}.$$

При $T = 2 \times t_i$ (що еквівалентно величині $1/Q = 50\%$) такий прямокутний сигнал називається меандром;

в) визначаємо постійну часу досліджуваного RC-кола:

$$\tau_i = RC < t_i.$$

Зробимо підстановку в цей вираз:


$$500 \times C < 0,05 \times 10^{-3}.$$

З останньої формули визначаємо C , яка й була необхідна для дослідження RC-кола:

$$C < 0,05 \times 10^{-3} / 5 \times 10^2 = 0,05 \times 10^{-5}.$$

Таким чином, величина ємності C повинна бути набагато меншою, ніж $0,05 \times 10^{-5} = 500 \times 10^{-9}$.

У нашому прикладі можна вибрати $C = 100 \times 10^{-9} = 100 \text{ нФ}$.

При виконанні цього дослідження необхідно вміти користуватися осцилографом та функціональним генератором. Знак функціональних пристроїв вибирається з піктограми, яка розміщується також в останньому рядку віконця програми EWB за кнопкою з зображенням .

Осцилограф (Oscilloscope). Осцилограф має два канали (Channel) А та В з роздільним регулюванням чутливості в діапазоні від 10 мкВ/діл (mV/Div) до 5 кВ/діл (kV/Div) і регулюванням зсуву по вертикалі (Y Position) (рис. 3).

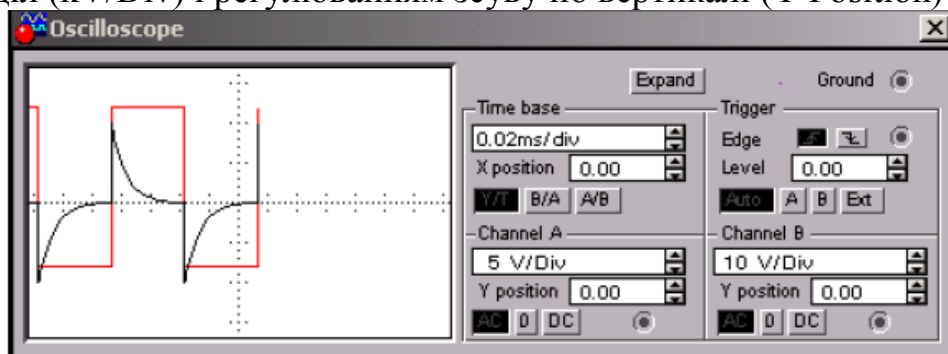



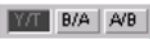


Рис. 3. Лицева панель осцилографа

Вибір режиму по входу здійснюється натисненням кнопок   .

Режим АС призначений для спостереження сигналів тільки змінного струму (його називають ще режимом «закритого входу», оскільки в цьому режимі на вході підсилювача включається розділовий конденсатор, який не пропускає постійної складової напруги).


У режимі «0» вхідний сигнал замикається на землю. У режимі DC (включений за замовченням) можна проводити осцилографічні вимірювання як постійного, так і змінного струму. Цей режим ще називають режимом «відкритого входу», оскільки вхідний сигнал надходить на вхід вертикального підсилювача безпосередньо. З правого боку від кнопки DC розташований вхідний затиск.

Режим розгортки вибирається кнопками . У режимі Y/T (звичайний режим, включений за замовченням) реалізуються такі режими розгортки: по вертикалі – напруга сигналу; по горизонталі – час;

– у режимі B/A: по вертикалі – сигнал каналу B; по горизонталі – сигнал каналу A;

– у режимі A/B: по вертикалі – сигнал каналу A; по горизонталі – сигнал каналу B.

У режимі Y/T тривалість розгортки (Time base) може бути задана в діапазоні від 0,1 нс/діл (ns/div) до 1 с/діл (s/div) з можливістю установки зсуву в тих же одиницях по горизонталі, тобто по осі X (X Position).

У режимі Y/T передбачений також режим очікування (Trigger) із запуском розгортки (Edge) по передньому або задньому фронту сигналу, що запускає (вибирається натисканням кнопок  при регульованому рівні (Level) запуску, а також в режимі Auto (від каналу A або B), від каналу A, від каналу B або від зовнішнього джерела (Ext), що підключається до контакту в блоці управління Trigger.

Названі режими запуску розгортки вибираються кнопками .

Заземлення осцилографа здійснюється за допомогою клеми Ground, яка розташована в правому верхньому кутку приладу.

Якщо при достатньо великій частоті функціонального генератора не вдається на осцилографі зупинити розгортку, а бо коли вона дуже мерехтить, необхідно натиснути кнопку Expand (рис. 4).

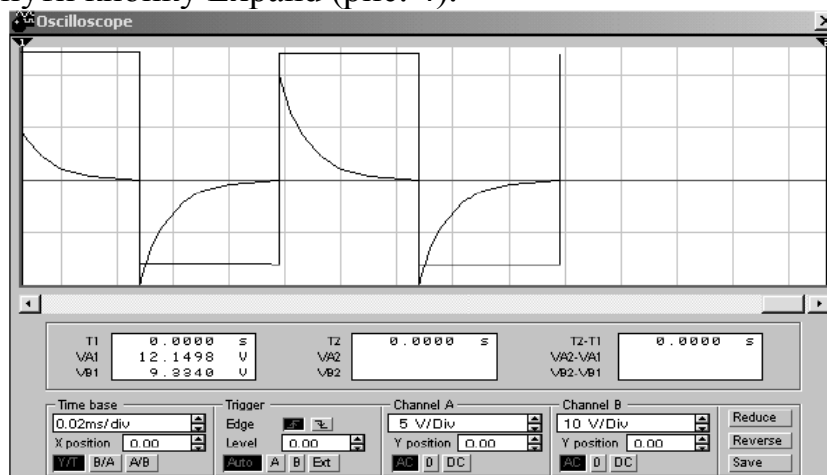


Рис.4. Панель осцилографа в режимі Expand

При натисканні на кнопку Expand лицева панель осцилографа істотно міняється – збільшується розмір екрана, з'являється можливість «прокрутки» зображення по горизонталі і його сканування з допомогою вертикальних візерних ліній (синього і червоного кольору), які за трикутні «вушка» (вони позначені цифрами 1 і 2) можуть бути встановлені курсором у будь-яке місце екрана. При цьому в індикаторних віконцях під екраном наводяться результати вимірювання напруги, тимчасових інтервалів і їх приростів (між візерними лініями).

При натисканні на кнопку Expand лицева панель осцилографа істотно міняється – збільшується розмір екрана, з'являється можливість «прокрутки» зображення по горизонталі і його сканування з допомогою вертикальних візерних ліній (синього і червоного кольору), які за трикутні «вушка» (вони позначені цифрами 1 і 2) можуть бути встановлені курсором у будь-яке місце екрана. При цьому в індикаторних віконцях під екраном наводяться результати вимірювання напруги, тимчасових інтервалів і їх приростів (між візерними лініями).

Зображення можна інвертувати натисканням кнопки Reverse і записати дані у файл натисканням кнопки Save.

Повернення до початкового стану осцилографа проводиться натисканням кнопки Reduce.

Функціональний генератор (*Function Generator*). Управління генератором здійснюється за допомогою таких органів управління (рис. 5):

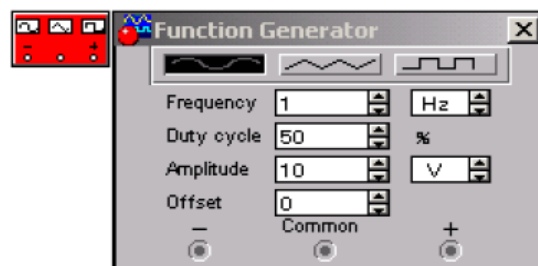


Рис.5. Зовнішній вигляд та лицева панель функціонального генератора


–  вибір форми вихідного сигналу: синусоїдальної (вибір за умовчанням), трикутної і прямокутної;

– Frequency - установка частоти вихідного сигналу

– Duty cycle - установка коефіцієнта заповнення в %: для імпульсних сигналів це відношення тривалості імпульсу до періоду повторення (величина, зворотна шпаруватості), для трикутних сигналів - співвідношення між тривалістю переднього і заднього фронтів;

– Amplitude - установка амплітуди вихідного сигналу;

– Offset - установка зсуву (постійної складової) вихідного сигналу;

–  вихідні затиски: при заземленні клеми Common (загальний) на клеммах "-" і "+" одержуємо парафазний сигнал.

Порядок з'єднання елементів для дослідження RC-кіл є аналогічним діям, виконаним у підпункті 1.

Одержати і зобразити осцилограми вхідної напруги та вихідної з виходу елемента R у даному кілі. Один з можливих варіантів з'єднання і дослідження диференціюючого RC-кола наведений на рис. 6.

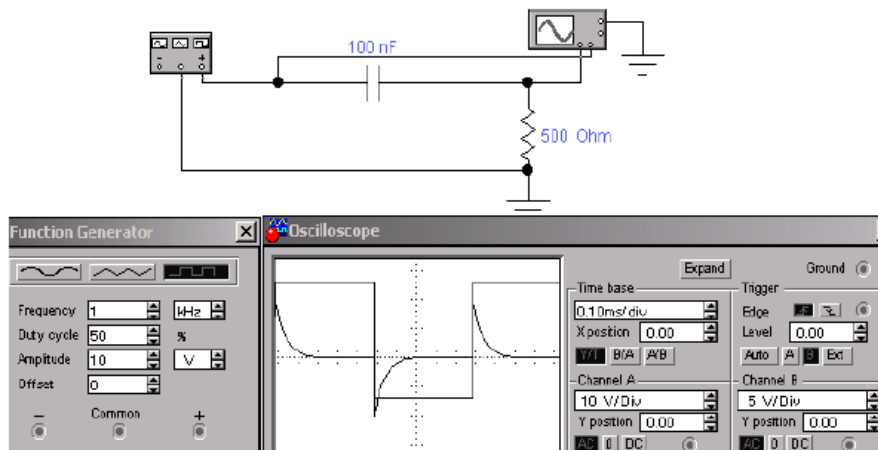


Рис.6. Вікно програми EWB 5.12 зі схемою дослідження диференціюючого RC-кола

3. Дослідження інтегруючих RC-кіл

Розрахунок інтегруючого RC-кола виконується аналогічно, за винятком того, що вихідна напруга знімається з ємності (рис. 12). Виконати розрахунок інтегруючого RC-кола відповідно до варіанта, наведених в табл. 3.

Таблиця 3

№ з/п	Частота f , Гц	Амп. $U_{вх}$	C , F	R , Ом	№ з/п	Частота f , Гц	Амп. $U_{вх}$	C , F	R , Ом
1	50	2	$200 \cdot 10^{-9}$	-	9	50	40	-	200
2	100	5	$100 \cdot 10^{-9}$	-	10	100	50	-	300
3	200	10	$300 \cdot 10^{-9}$	-	11	200	60	-	400
4	300	20	$300 \cdot 10^{-9}$	-	12	300	70	-	500
5	500	25	$100 \cdot 10^{-9}$	-	13	500	1	-	500
6	$1 \cdot 10^3$	30	$50 \cdot 10^{-9}$	-	14	$1 \cdot 10^3$	5	-	600
7	$10 \cdot 10^3$	35	$10 \cdot 10^{-9}$	-	15	$10 \cdot 10^3$	10	-	700
8	$50 \cdot 10^3$	40	$10 \cdot 10^{-9}$	-	16	$50 \cdot 10^3$	20	-	500

Одержати і зобразити осцилограми вхідної напруги та вихідної з виходу елемента C у даному кілі. Один з можливих варіантів з'єднання і дослідження кола наведений на рис. 7.

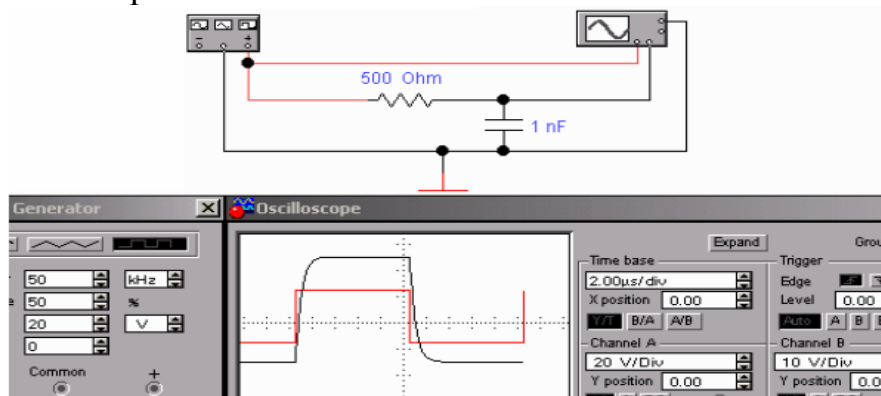


Рис.7. Вікно програми EWB 5.12 зі схемою дослідження інтегруючого RC-кола

Порядок передачі зображення в текстовий редактор Word. Увійти в меню Edit та натиснути кнопку Copy; натиснути на клавіатурі кнопку Print Screen. Для запам'ятовування в буфері зображення активної сторінки в Word необхідно натиснути одночасно дві клавіші Alt та Print Screen. Для видалення непотрібних частин зображення слід скористатися програмою Paint.

4. Контрольні питання

1. Як визначається аналітично і геометрично постійна часу кола і який її фізичний сенс?
2. При виконанні якої умови на RC-колі можливо отримання двох коротких імпульсів?
3. Визначте області застосування диференціюючих і інтегруючих кіл?
4. За яким законом змінюється величина напруги на конденсаторі в RC-колі?
5. Який вплив на вихідну напругу здійснюють паразитні параметри кола?
6. В чому полягає інтегрування електричних сигналів?
7. В чому полягає диференціювання електричних сигналів?
8. Що називається електричним фільтром?

Навчальні питання:

Тема 3. Логічні основи цифрової схемотехніки

Лабораторне заняття №3. Дослідження логічних елементів та логічних функцій

Навчальна мета заняття:

1. Закріплення теоретичного матеріалу, набуття навиків аналізу та синтезу простих логічних пристроїв.
2. Закріплення теоретичного матеріалу, набуття навиків створення і моделювання схем цифрової схемотехніки, робота з різними вимірювальними приладами.

Кількість годин: 6 год

Література:

1. Рябенський В.М. Жуйков В.Я. Ямненко Ю.С. Заграничний А.В. Схемотехніка: Пристрої цифрової електроніки. Київ: 2016, КПІ. 399 с.
2. Кравець В.О., Сокол Є.І., Рисований О.М. Комп'ютерна схемотехніка. Підручник. Харків: НТУ «ХПІ», 2007. 480 с.

Матеріально-технічне забезпечення: комп'ютерна мережа із системою модулювання електронних схем Electronics Workbench.

Навчальні питання:

1. Розв'язання завдань щодо мінімізації логічних функцій
2. Індивідуальні завдання на тему «Логічні основи цифрової схемотехніки»
3. Дослідження логічних схем з допомогою середовища моделювання
4. Дослідження функціонування логічних елементів у динамічному режимі роботи
5. Контрольні питання

1. Розв'язання завдань щодо мінімізації логічних функцій

Завдання 1. Перетворимо до ВДНФ логічну функцію: $f(x) = \overline{x_1}x_2 \vee x_2\overline{x_3}$. Оскільки функція залежить від трьох змінних x_1, x_2, x_3 , першу кон'юнкцію помножимо на вираз $(x_3 \vee \overline{x_3})$, а другу – на $(x_1 \vee \overline{x_1})$:

$$f(x) = \overline{x_1}x_2(x_3 \vee \overline{x_3}) \vee (x_1 \vee \overline{x_1})x_2\overline{x_3} = \overline{x_1}x_2x_3 \vee \overline{x_1}x_2\overline{x_3} \vee x_1x_2\overline{x_3} \vee \overline{x_1}x_2\overline{x_3} = \overline{x_1}x_2x_3 \vee \overline{x_1}x_2\overline{x_3} \vee x_1x_2\overline{x_3}$$

Характерною властивістю ВДНФ є те, що представлення в ній логічної функції єдині. Звідси виходить, що кожна ВДНФ містить стільки конститuent одиниці, скільки одиничних наборів має логічна функція. Так, функція, розглянута в прикладі, задана на трьох одиничних наборах, отже, її ВДНФ має три конституенти одиниці. Логічна функція константи одиниці у ВДНФ представляється диз'юнкцією 2^n конститuent одиниці.

Завдання 2. Логічна функція трьох змінних задана табл. 1. $f(x_1, x_2, x_3) = K_1(1) + K_3(1) + K_6(1) + K_7(1) = \overline{x_1} \overline{x_2} x_3 \vee \overline{x_1} x_2 x_3 \vee x_1 \overline{x_2} x_3 \vee x_1 x_2 x_3$.

Кон'юнктивна нормальна форма називається вдосконаленою, якщо кожна елементарна диз'юнкція, що входить до неї, має в прямому або інверсійному вигляді всі змінні, від яких залежить функція.

Характерною властивістю вдосконаленої КНФ є те, що представлення в ній логічної функції єдині. Елементарні диз'юнкції, які входять у вдосконалену КНФ функції, носять назву конститuent нуля. Кожна конститuenta нуля, що входить у вдосконалену КНФ, набуває нульового значення на єдиному наборі значень змінних, який є нульовим набором функції. Отже, кількість нульових наборів логічної функції збігається з кількістю конститuent нуля, які входять в її вдосконалену КНФ.

Для кожного рядка таблиці істинності, у якій функція дорівнює нулю, складається елементарна диз'юнкція всіх змінних. При цьому в диз'юнкцію входить сама змінна, якщо її значення дорівнює нулю, або заперечення, якщо його значення дорівнює одиниці. Одержані елементарні диз'юнкції об'єднуються знаком кон'юнкції.

Таблиця 1

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$f(x_1, x_2, x_3) = K_0(0) + K_2(0) + K_4(0) + K_5(0) = (x_1 \vee x_2 \vee x_3)(x_1 \vee \overline{x_2} \vee x_3)(\overline{x_1} \vee x_2 \vee x_3)(\overline{x_1} \vee x_2 \vee \overline{x_3})$$

Слід зазначити, що для функцій, кількість одиничних наборів яких перевищує кількість нульових наборів, більш компактним є запис у вигляді ВКНФ і навпаки.

Завдання 3. Представити логічну функцію $f(x_1, x_2, x_3, x_4) = \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4}$ у ДНФ.

Послідовно виконуючи теореми де Моргана перетворимо функцію в ДНФ:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1 x_2 x_3 x_4} = \overline{x_1} + \overline{x_2 x_3 x_4} = \overline{x_1} + \overline{x_2} \overline{x_3 x_4} = \overline{x_1} + \overline{x_2} (\overline{x_3} + \overline{x_4}) = \overline{x_1} + \overline{x_2} \overline{x_3} + \overline{x_2} \overline{x_4}$$

Завдання 4. Логічну функцію, яка подана табл.2 істинності виразити у ВДНФ та ВКНФ, мінімізувати їх методом безпосередніх перетворень. Побудувати функціональні схеми, які реалізують ці функції у загальному базисі та базисах *I-НІ* та *АБО-НІ*.

Таблиця 2

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0	1	0	0	1
0	0	1	1	1	0	1	1
0	1	0	0	1	1	0	0
0	1	1	1	1	1	1	0

Представимо спочатку функцію у ВДНФ, тобто визначимо її одиничні значення за допомогою конститuent одиниць.

$$F(x_1, x_2, x_3) = \Sigma(1, 3, 4, 5) = \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 \bar{x}_3 + x_1 x_2 \bar{x}_3$$

Мінімізуємо цю функцію методом безпосередніх перетворень:

$$F(x_1, x_2, x_3) = \bar{x}_1 x_3 (\bar{x}_2 + x_2) + x_1 \bar{x}_2 (x_3 + \bar{x}_3) = \bar{x}_1 x_3 + x_1 \bar{x}_2$$

Далі побудуємо функціональну схему, яка реалізує цю функцію в основному базисі.

Щоб реалізувати цю ж функцію в базисі І-НІ, необхідно ДНФ функції надати до такої форми, яка б не вміщала операції логічного додавання. Для цього необхідно від функції взяти подвійне заперечення (що не змінює його значення) та за теоремою де Моргана виконати перетворення: $F(x_1, x_2, x_3) = \overline{\overline{\bar{x}_1 x_3 + x_1 \bar{x}_2}} = \overline{\overline{\bar{x}_1} \overline{x_3} \overline{x_1} \overline{\bar{x}_2}} = \overline{\overline{\bar{x}_1} \overline{x_3} \overline{x_1} x_2}$. Цей логічний вираз вміщує тільки операції І-НІ. Побудуємо його функціональну схему.

Представимо цю ж функцію у ВКНФ. Це означає, що потрібно визначити її за нульовими наборами за допомогою конститuent нуля:

$$F(x_1, x_2, x_3) = \Pi(0, 2, 6, 7) = (x_1 + x_2 + x_3)(x_1 + \bar{x}_2 + x_3)(\bar{x}_1 + \bar{x}_2 + x_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3)$$

Для мінімізації отриманої функції використаємо закон склеювання:

$$(x_1 + x_2 + x_3)(x_1 + \bar{x}_2 + x_3) = (x_1 + x_3);$$

$$(\bar{x}_1 + \bar{x}_2 + x_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3) = (\bar{x}_1 + \bar{x}_2).$$

Побудуємо функціональну схему, яка реалізує цю функцію в основному базисі.

Для побудови цієї функції у базисі АБО-НІ необхідно в КНФ позбавитись операції кон'юнкції. Ця операція відбувається при використанні подвійного заперечення від функції за допомогою теореми де-Моргана:

$$F(x_1, x_2, x_3) = \overline{\overline{(x_1 + x_3)(\bar{x}_1 + \bar{x}_2)}} = \overline{\overline{x_1 + x_3} \overline{\bar{x}_1 + \bar{x}_2}} = \overline{\overline{x_1 + x_3} \overline{x_1 + x_2}}$$

Будуємо функціональну схему в базисі АБО-НІ:

Завдання 5. Навести у мінімальній ДНФ та мінімальній КНФ логічну функцію трьох змінних, які подані у таблиці істинності.

Таблиця 3

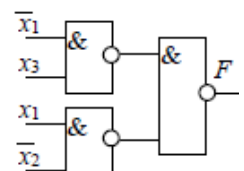
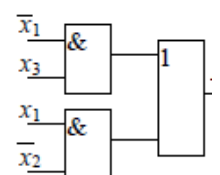
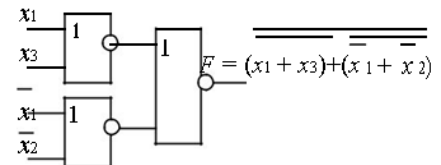
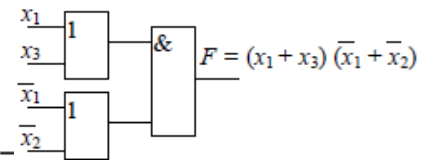
x ₁	x ₂	x ₃	f(x ₁ , x ₂ , x ₃)	x ₁	x ₂	x ₃	f(x ₁ , x ₂ , x ₃)
0	0	0	0	1	0	0	1
0	0	1	0	1	0	1	1
0	1	0	0	1	1	0	0
0	1	1	0	1	1	1	0

Запишемо функцію у ВДНФ та мінімізуємо її:

$$F(x_1, x_2, x_3) = x_1 \bar{x}_2 \bar{x}_3 + x_1 \bar{x}_2 x_3 = x_1 \bar{x}_2$$

Представимо цю ж функцію у ВКНФ і також мінімізуємо її:

$$F(x_1, x_2, x_3) = (x_1 + x_2 + x_3)(x_1 + x_2 + \bar{x}_3)(x_1 + \bar{x}_2 + \bar{x}_3) + (x_1 + \bar{x}_2 + x_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3) = (x_1 + x_2)(x_1 + x_2)(x_1 + \bar{x}_2) = x_1 \bar{x}_2$$

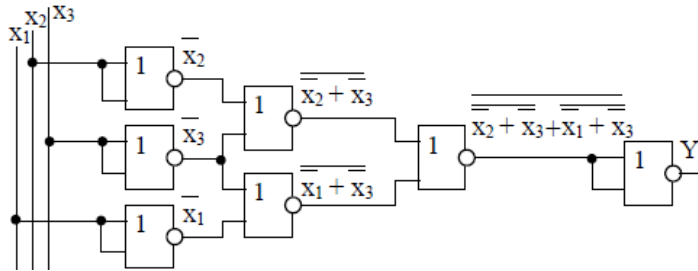


Наведені приклади дозволяють ще раз переконатися, що функції у ДНФ та КНФ дозволяють отримати одні й ті ж результати.

Завдання 6. Мінімізувати логічну функцію та привести до базису АБО-НІ. Побудувати функціональну схему на двовходових елементах.

$$Y = \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 = \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3 = \\ = \overline{\overline{x_2 x_3} + \overline{x_1 x_3}} = \overline{\overline{x_2 x_3} \cdot \overline{x_1 x_3}} = \overline{(\overline{x_2 + x_3})(\overline{x_1 + x_3})} = \overline{\overline{x_2 + x_3}} \cdot \overline{\overline{x_1 + x_3}} = x_2 + x_3 + x_1 + x_3.$$

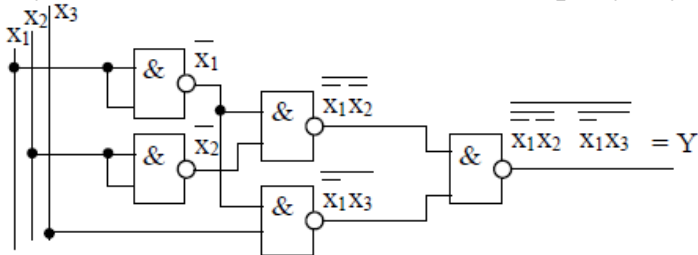
Функціональна схема, яка реалізує функцію, наведена на рисунку:



Завдання 7. Мінімізувати логічну функцію та привести до базису І-НІ. Побудувати функціональну схему на двовходових елементах.

$$Y = \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 = \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 = \\ = \bar{x}_1 x_2 + x_1 x_2 x_3 = \bar{x}_1 (\bar{x}_2 + x_2 x_3) = \bar{x}_1 (\bar{x}_2 + x_3) = \bar{x}_1 \bar{x}_2 + \bar{x}_1 x_3 = \overline{\overline{\bar{x}_1 \bar{x}_2} \cdot \overline{\bar{x}_1 x_3}}.$$

Функціональна схема наведена на рисунку:



Завдання 8. Мінімізувати логічну функцію та привести до базису І-НІ.

$$Y = x_1 x_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3 = x_2 x_3 + x_1 x_2 x_3 = x_2 (x_3 + x_1 x_3) = \\ = x_2 (\bar{x}_3 + \bar{x}_1) = x_2 \bar{x}_3 + x_2 \bar{x}_1 = \overline{\overline{x_2 \bar{x}_3} \cdot \overline{x_2 \bar{x}_1}}.$$

Завдання 9. Мінімізувати логічну функцію та привести до базису АБО-НІ.

$$Y = x_1 x_3 x_4 + x_2 x_3 x_4 + x_1 x_3 x_4 + x_2 x_4 = x_1 x_3 x_4 + x_4 (x_2 x_3 + x_2) + x_1 x_3 x_4 = \\ = x_1 x_3 x_4 + x_4 (x_3 + \bar{x}_2) + x_1 \bar{x}_3 x_4 = x_1 x_3 x_4 + x_3 x_4 + \bar{x}_2 x_4 + x_1 \bar{x}_3 x_4 = \\ = x_3 x_4 + \bar{x}_2 x_4 + x_1 \bar{x}_3 x_4 = x_2 x_4 + x_3 x_4 x_1 = \overline{\overline{x_2 x_4} \cdot \overline{x_3 x_4 x_1}} = \overline{\overline{x_2 + x_4} \cdot \overline{x_3 + x_4 + x_1}}.$$

Завдання 10. Приклади використання карт Карно.

Для визначення мінімальної ДНФ за допомогою карт Карно необхідно поєднати усі клітинки, які вміщують одиницю, у замкнені прямокутні області (контури). У цих контурах кількість клітинок у кожній може дорівнювати 2^k , де $k = 0, 1, 2, 3, \dots, n$. Зазначені області можуть перетинатися, і одна й та сама клітинка може належати кільком областям. Але слід пам'ятати, що будь-який контур повинен мати хоча б одну клітинку, яка належить виключно йому.

Кількість таких областей має бути якомога меншою, а кількість клітинок у кожній такій області – якомога більшою. За цих умов кількість аргументів для членів

мінімізованої функції буде мінімальною. При охопленні усіх точок карти Карно слід пам'ятати, що права та ліва границі карти – це сусідні стовпчики, так само, як верхній та нижній рядки карти Карно – це сусідні рядки (наприклад, чотири одиниці, розташовані в кутах карти, утворюють один контур).

Слід також зазначити, що якщо для карти Карно кількість нулів значно менша за кількість одиниць, то зручніше склеювати саме їх. Не варто тільки забувати, що у цьому випадку ми отримаємо не саму мінімальну функцію, а її інверсію, яку потім необхідно проінвертувати.

Дві змінних

$$f(x_1, x_2) = (\overline{x_1} + \overline{x_2})(x_1 + \overline{x_2})(x_1 + x_2).$$

x_2	0	1
x_1		
0	0	0
1	1	0

$$f(x_1, x_2) = \overline{x_1 x_2}.$$

Три змінних

x_3	00	01	11	10
x_1				
0	0	0	1	1
1	1	0	0	1

$$f(x_1, x_2) = \overline{x_1} x_2 + x_1 \overline{x_3}.$$

Чотири змінних

x_4	00	01	11	10
x_2				
00	1	0	0	1
01	1	0	1	0
11	1	0	1	0
10	1	0	0	1

$$f(x_1, x_2) = \overline{x_3} \overline{x_4} + \overline{x_2} \overline{x_4} + x_2 x_3 x_4.$$

Відомо, що деякі комбінації вхідних змінних логічних функцій з'явитися не можуть (їх значення неістотно). У таких випадках говорять про невизначені набори.

На картах Карно невизначені набори позначаються яким-небудь символом, наприклад, “*”. Такі елементи можуть довільним чином входити до груп як одиничних, так й нульових об'єднань.

$$f(x) = \Pi_0(2, 4, 5, 6, 7, 10, 11) + \Pi_{\#}(14, 15).$$

x_3	00	01	11	10
x_1				
00	1	1	1	0
01	0	0	0	0
11	1	1	*	*
10	1	1	0	0

$$f(x) = (x_3 + \overline{x_4})(x_2 + \overline{x_1})(x_4 + x_2).$$

2. Індивідуальні завдання на тему «Логічні основи цифрової схемотехніки»

Контрольна робота є важливою складовою навчального процесу. Вона доповнює знання курсантів, отриманні на лекційних та лабораторних заняттях.

Синтез логічних схем передбачає побудову структурної схеми пристрою. Визначення складу логічних елементів і з'єднань між ними, при яких забезпечується перетворення вхідних цифрових сигналів у вихідні. Під час синтезу зазвичай виконується мінімізація апаратних витрат на реалізацію пристрою. Синтез можна розділити на такі етапи:

1. Визначення умов функціонування цифрового пристрою. Ці умови можуть бути задані за допомогою таблиць істинності, аналітичних виразів, або словами.

2. Визначення виразу логічної функції. Мінімізація виразу з допомогою метода безпосередніх перетворень або карт Карно (діаграм Вейча).

3. Перетворення мінімізованого виразу логічної функції в заданий базис її схемної реалізації.

4. Складання структурної схеми, тобто зображення потрібних елементів і зв'язків між ними.

Аналіз логічних схем передбачає оцінювання деяких характеристик структури цифрових пристроїв.

Таблиця 6. Вибір варіанту

	x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1		0	0	0	0	0	0	1	0	0	1	0	1	1	1	1	1
2		1	0	1	0	0	0	1	1	0	0	1	0	0	0	1	1
3		1	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0
4		1	0	0	0	1	1	1	1	0	0	0	0	1	0	1	0
5		0	1	0	1	0	1	0	1	0	0	0	0	0	1	1	1
6		0	1	0	0	0	0	0	0	0	1	0	1	1	1	1	1
7		1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	1
8		1	0	1	0	1	0	1	1	0	0	1	0	0	0	1	0
9		1	1	1	1	0	0	1	1	0	1	0	0	0	0	0	0
10		0	0	0	0	1	0	0	0	0	1	0	1	1	1	1	1
11		0	0	1	0	0	0	1	1	1	0	1	0	0	0	1	1
12		0	1	1	1	0	1	0	1	0	1	0	1	0	0	0	0
13		0	0	1	0	1	1	1	1	0	0	0	0	1	0	1	0
14		0	1	0	1	0	1	1	1	0	0	0	0	0	1	0	1
15		0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1
16		1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	0
17		1	0	1	0	1	0	1	0	0	0	1	1	0	0	1	0
18		1	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0
19		0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	1
20		0	0	1	0	0	1	1	1	0	0	1	0	0	0	1	1
21		0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	0
22		0	0	0	0	1	1	1	1	0	0	1	0	1	0	1	0
23		0	1	0	1	1	1	0	1	0	0	0	0	0	1	0	1
24		0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0
25		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0

Приклад виконання завдання.

1. Формування варіанту завдання відповідно до таблиці істинності (табл. 7).

Таблиця 7

№	x_1	x_2	x_3	x_4	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1

7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

2. Знаходження ВДНФ або ВКНФ.

Для знаходження ВДНФ формуємо табл. 8. Для цього виписуємо з табл. 7 індивідуального завдання ті рядки, в яких функція приймає значення одиниці.

Таблиця 8

№	x_1	x_2	x_3	x_4	F
2	0	0	1	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
10	1	0	1	0	1
14	1	1	1	0	1
15	1	1	1	1	1

Знаходимо конституенти одиниці для цих наборів. Одержуємо їх як кон'юнкцію n змінних, що входять до аналітичного виразу у прямому виді, якщо значення даної змінної в наборі $x_i = 1$, або в інверсному вигляді, якщо $x_i = 0$. Таким чином, отримуємо:

для набору 2 – $K_2 = (\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4)$;

для набору 5 – $K_5 = (\bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4)$;

для набору 6 – $K_6 = (\bar{x}_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4)$;

для набору 7 – $K_7 = (\bar{x}_1 \cdot x_2 \cdot x_3 \cdot x_4)$;

для набору 10 – $K_{10} = (x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4)$;

для набору 14 – $K_{14} = (x_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4)$;

для набору 15 – $K_{15} = (x_1 \cdot x_2 \cdot x_3 \cdot x_4)$.

Об'єднуємо конституенти одиниці цих наборів, використовуючи операцію логічної суми $K_2 \vee K_5 \vee K_6 \vee K_7 \vee K_{10} \vee K_{14} \vee K_{15}$. Це буде аналітичний вираз для функції F . Таке подання функції має назву "досконала диз'юнктивна нормальна форма" (ВДНФ). Записуємо отриманий вираз у вигляді

$$F_{\text{ВДНФ}} = (\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4) \vee (\bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4) \vee (\bar{x}_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4) \vee (\bar{x}_1 \cdot x_2 \cdot x_3 \cdot x_4) \vee (x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4) \vee (x_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4) \vee (x_1 \cdot x_2 \cdot x_3 \cdot x_4)$$

Для знаходження ДКНФ формуємо табл. 9. Для цього виписуємо з табл. 1 індивідуального завдання ті рядки, в яких функція приймає значення нуля.

Таблиця 9

№	x_1	x_2	x_3	x_4	F
0	0	0	0	0	0
1	0	0	0	1	0

3	0	0	1	1	0
4	0	1	0	0	0
8	1	0	0	0	0
9	1	0	0	1	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0

Знаходимо конституенти 0 для цих наборів. Одержуємо їх як диз'юнкцію усіх змінних, що входять до аналітичного виразу у прямому вигляді, коли значення даної змінної в наборі $x_i = 0$, або в інверсному вигляді, якщо значення $x_i = 1$. Таким чином, отримуємо:

для набору 0 – $M_0 = x_1 \vee x_2 \vee x_3 \vee x_4$;

для набору 1 – $M_1 = x_1 \vee x_2 \vee x_3 \vee \bar{x}_4$;

для набору 3 – $M_3 = x_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4$;

для набору 4 – $M_4 = x_1 \vee \bar{x}_2 \vee x_3 \vee x_4$;

для набору 8 – $M_8 = \bar{x}_1 \vee x_2 \vee x_3 \vee x_4$;

для набору 9 – $M_9 = \bar{x}_1 \vee x_2 \vee x_3 \vee \bar{x}_4$;

для набору 11 – $M_{11} = \bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4$;

для набору 12 – $M_{12} = \bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee x_4$;

для набору 13 – $M_{13} = \bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4$.

Об'єднуємо конституенти нуля тих наборів, де функція приймає значення нуля. Для цього використовуємо операцію логічного добутку. В результаті отримуємо досконалу кон'юнктивну нормальну форму функції F у вигляді

$$\begin{aligned}
 F_{\text{ДКНФ}} = & (x_1 \vee x_2 \vee x_3 \vee x_4) \cdot (x_1 \vee x_2 \vee x_3 \vee \bar{x}_4) \cdot (x_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4) \cdot \\
 & \cdot (x_1 \vee \bar{x}_2 \vee x_3 \vee x_4) \cdot (\bar{x}_1 \vee x_2 \vee x_3 \vee x_4) \cdot (\bar{x}_1 \vee x_2 \vee x_3 \vee \bar{x}_4) \cdot \\
 & \cdot (\bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4) \cdot (\bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee x_4) \cdot (\bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4)
 \end{aligned}$$

3. Мінімізація булевих функцій.

Знаходження МДНФ. Заповнюємо діаграму Вейча. Для цього в комірках діаграми обраної форми, що зображена на рис. 7а, записуємо одиниці. Це робимо в тих комірках, номери яких збігаються з номерами наборів, де задана функція приймає значення одиниці. В результаті отримуємо робочу діаграму Вейча для знаходження МДНФ, показану на рис. 7б.

		x_2					
x_1		12	13	9	8		x_3
		14	15	11	10		
		6	7	3	2		
		4	5	1	0		
		x_4					
a							

	x_2				
x_1					x_3
	1	1		1	
	1	1		1	
		1			
	x_4				
6					

Рис. 7. Діаграма Вейча для знаходження МДНФ (а – форма діаграми Вейча, що використовується при мінімізації, б – заповнена робоча діаграма для заданої функції)

	x_2				
x_1	0	0	0	0	x_3
	1	1	0	1	
	1	1	0	1	
	0	1	0	0	
	x_4				

Рис. 8. Формування прямокутних конфігурацій на діаграмі Вейча для мінімізації ДНФ заданої функції

Далі виконуємо операцію склеювання. Склеюванню підлягають прямокутні конфігурації, які заповнені одиницями і містять 2, 4, 8 або 16 комірок. Верхні й нижні рядки, крайні ліві і крайні праві стовпці діаграми ніби склеюються, утворюючи поверхню циліндра. Перша область – поруч розташовані одиниці 6, 7, 14, 15 наборів, утворюють квадрат з чотирьох поруч розташованих елементів. Друга область – одиниці 2, 10, 6, 14 наборів розташовані поруч, якщо розглядати діаграму, як поверхню циліндра. Залишилось розглянути одиницю з 5 набору. Її можливо склеїти лише з одиницею з 7 набору. Тоді утворюється область з двох поруч розташованих елементів (рис.8).

Формула, виходить в результаті мінімізації, містить 3 елементарних кон'юнкції (за числом прямокутників у покритті). Кожна кон'юнкція містить тільки ті змінні, які не змінюють свого значення в наборах, що склеюються у відповідному прямокутнику. Змінна вибирається у прямому виді, якщо її значення в наборі дорівнює 1, та з запереченням в іншому випадку.

$$F_{\text{МДНФ}} = (x_2 \cdot x_3) \vee (x_3 \cdot \bar{x}_4) \vee (\bar{x}_1 \cdot x_2 \cdot x_4).$$

Знаходження МКНФ. Заповнюємо діаграму Вейча. В ті клітинки номера, яких збігаються з номерами наборів, де функція приймає значення нуля записуємо 0 (рис. 9).

	x_2				
x_1	12	13	9	8	x_3
	14	15	11	10	
	6	7	3	2	
	4	5	1	0	
	x_4				

	x_2				
x_1	0	0	0	0	x_3
			0		
			0		
	0		0	0	
	x_4				

Рис. 9. Діаграма Вейча для знаходження МКНФ (а – форма діаграми Вейча, що використовується при мінімізації, б – заповнена робоча діаграма для заданої функції)

	x_2				
x_1	0	0	0	0	x_3
	1	1	0	1	
	1	1	0	1	
	0	1	0	0	
	x_4				

Рис. 10. Формування прямокутних конфігурацій на діаграмі Вейча для мінімізації КНФ заданої функції

Далі проводимо операцію склеювання. Склеюванню підлягають прямокутні конфігурації, які заповнені одиницями і містять 2, 4, 8 або 16 клітинок. Верхні й нижні рядки, крайні ліві і праві стовпчики діаграми ніби склеюються, створюючи поверхню циліндра (рис. 10). Перша область це верхній рядок, є чотири поруч розташованих нуля, які утворюють прямокутник (набори 8, 9, 12, 13). Друга область – вертикальний рядок з чотирьох нулів (набори 9, 11, 3, 1), які також утворюють прямокутник. Залишається два нуля на 4 та 0 наборах (у кутах) їх можна склеїти з іншими кутами. Тобто в третю область склеюємо кутові комірки (набори 0, 4, 8, 12)

Формула, виходить в результаті мінімізації, містить 3 елементарних диз'юнкції (за числом прямокутників у покритті). Кожна диз'юнкція містить тільки ті змінні, які не змінюють свого значення в наборах, що склеюються у відповідному прямокутнику. Змінна вибирається у прямому виді, якщо її значення в наборі дорівнює 0, та з запереченням в іншому випадку.

$$F_{\text{МКНФ}} = (\bar{x}_1 \vee x_3) \cdot (x_2 \vee \bar{x}_4) \cdot (x_3 \vee x_4).$$

4. Побудова та дослідження схем. Для побудови схеми використаємо отримані вирази для МДНФ та МКНФ. Ці вирази мають операції інверсії, кон'юнкції, диз'юнкції. Відповідно вибираємо логічні елементи, які реалізують ці операції схемотехнічно.

У виразі МДНФ заданої функції $F_{\text{МДНФ}}$ є три кон'юнкції (операції у дужках). Кожній операції на схемі відповідає логічний елемент «І». Вихід цих елементів з'єднаний з входом елемента «АБО», що відповідає диз'юнкції у виразі для МДНФ заданої функції. На вхід елементів «І» подаємо вхідні сигнали згідно виразу: на перший x_2, x_3 ; на другий \bar{x}_1, x_2, x_4 ; на третій x_3, \bar{x}_4 (рис. 11).

У виразі МКНФ заданої функції $F_{\text{МКНФ}}$ є три диз'юнкції (операції у дужках). Кожній операції на схемі відповідає логічний елемент «АБО». Вихід цих елементів з'єднаний з входом елемента «І», що відповідає кон'юнкції у виразі для МКНФ заданої функції. На вхід елементів «АБО» подаємо вхідні сигнали згідно виразу: на перший \bar{x}_1, x_3 ; на другий x_2, \bar{x}_4 ; на третій x_3, x_4 (рис. 12).

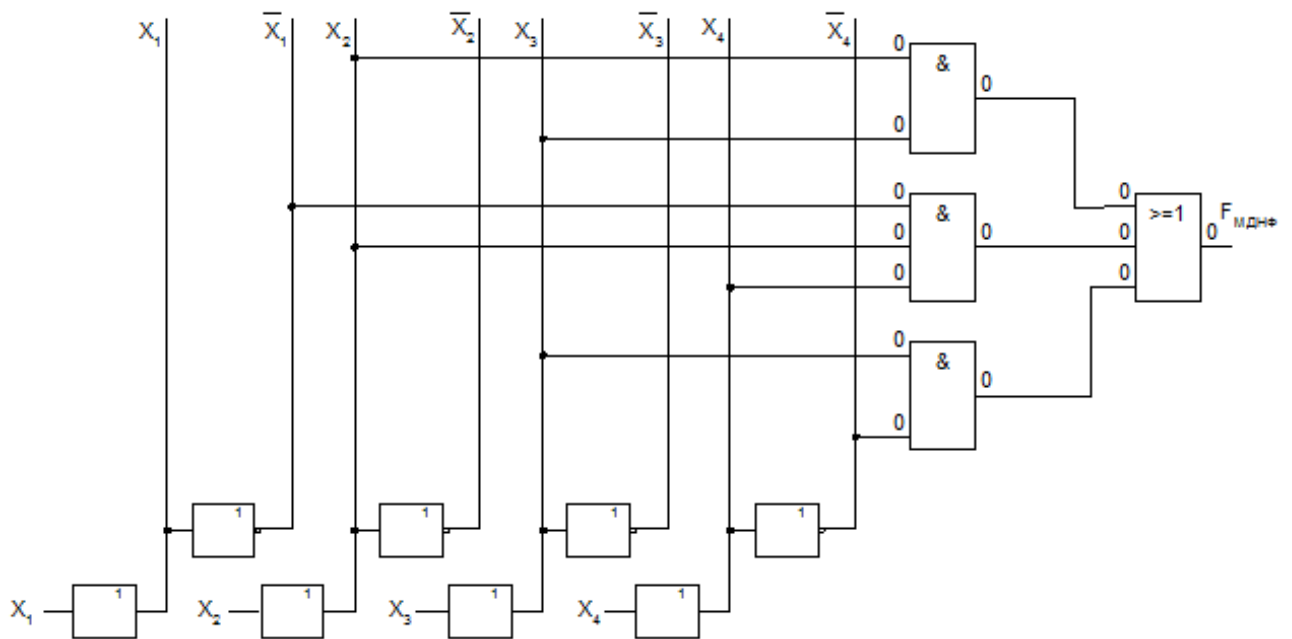


Рис. 11. Схема пристрою реалізації функції МДНФ

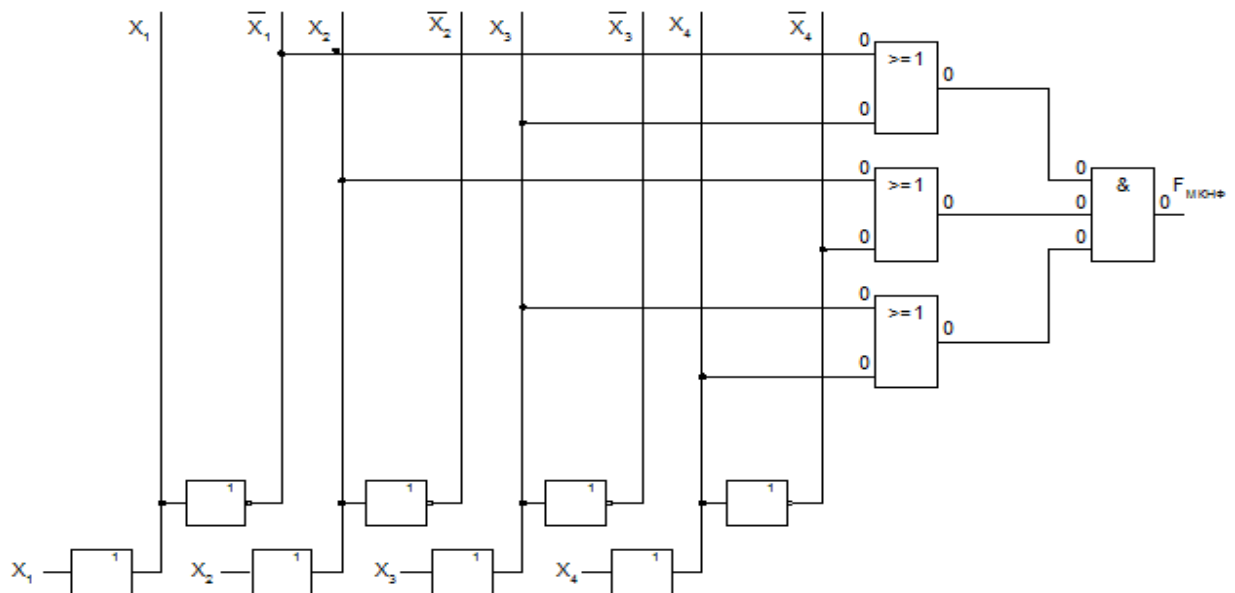


Рис. 12. Схема пристрою реалізації функції МКНФ

5. Побудова та дослідження схеми в базисі І-НІ або АБО-НІ. Для переходу в заданий базис беремо МДНФ. Спочатку ставимо над виразом подвійну інверсію.

$$F_{\text{МДНФ}} = (x_2 \cdot x_3) \vee (x_3 \cdot \bar{x}_4) \vee (\bar{x}_1 \cdot x_2 \cdot x_4) = \overline{\overline{(x_2 \cdot x_3) \vee (x_3 \cdot \bar{x}_4) \vee (\bar{x}_1 \cdot x_2 \cdot x_4)}}$$

Далі користуючись правилом де Моргана ($\overline{a \vee b} = \bar{a} \cdot \bar{b}$), розділяємо одну з операцій інверсії

$$\overline{(x_2 \cdot x_3) \vee (x_3 \cdot \bar{x}_4) \vee (\bar{x}_1 \cdot x_2 \cdot x_4)} = \overline{(x_2 \cdot x_3) \cdot (x_3 \cdot \bar{x}_4) \cdot (\bar{x}_1 \cdot x_2 \cdot x_4)}$$

Операції між дужками кон'юнкція, над нею є інверсія, тобто це операція Шеффера («І-НІ»)

$$\overline{(x_2 \cdot x_3) \cdot (x_3 \cdot \bar{x}_4) \cdot (\bar{x}_1 \cdot x_2 \cdot x_4)} = \overline{(x_2 \cdot x_3)} \mid \overline{(x_3 \cdot \bar{x}_4)} \mid \overline{(\bar{x}_1 \cdot x_2 \cdot x_4)}$$

Операції в дужках кон'юнкції, над кожною дужкою є інверсія, тобто це операції Шеффера («І-НІ»)

$$\overline{(x_2 \cdot x_3)} | \overline{(x_3 \cdot \bar{x}_4)} | \overline{(\bar{x}_1 \cdot x_2 \cdot x_4)} = (x_2 | x_3) | (x_3 | \bar{x}_4) | (\bar{x}_1 | x_2 | x_4)$$

Кожній операції на схемі відповідає логічний елемент «І-НІ». Тому схема складається з чотирьох елементів «І-НІ». На вхід трьох елементів «І-НІ», які відповідають операціям у дужках подаємо вхідні сигнали згідно виразу: на перший x_2, x_3 ; на другий \bar{x}_1, x_2, x_4 ; на третій x_3, \bar{x}_4 . Вихід цих елементів з'єднаний з входами елемента «І-НІ», що відповідає операції між дужками заданої функції (рис. 13).

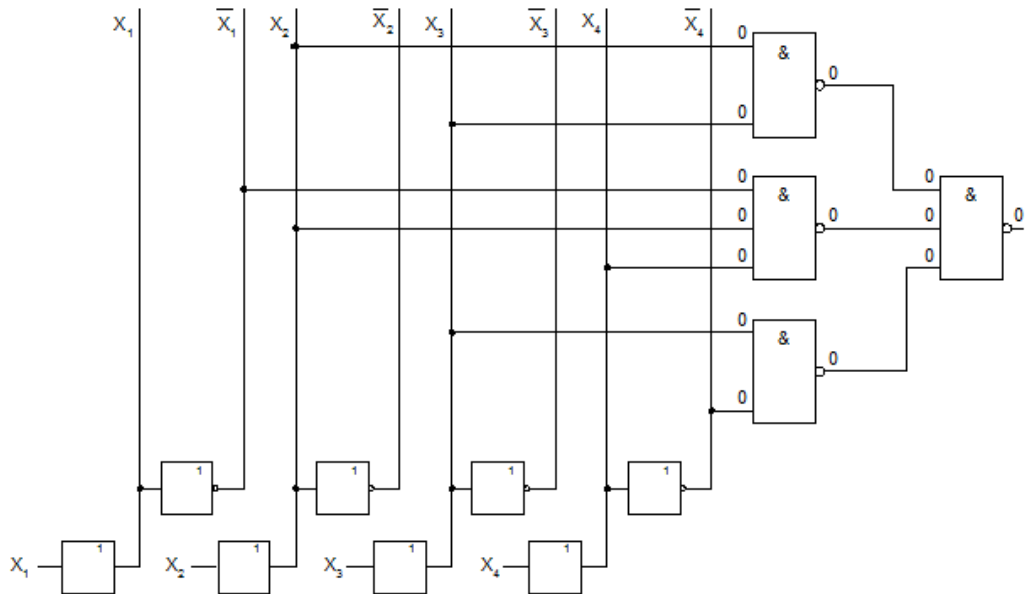


Рис. 13. Схема реалізації пристрою в базисі І-НІ

Для переходу в базис АБО-НІ беремо МКНФ. Спочатку ставимо над виразом подвійну інверсію.

$$F_{\text{МКНФ}} = (\bar{x}_1 \vee x_3) \cdot (x_2 \vee \bar{x}_4) \cdot (x_3 \vee x_4) = \overline{\overline{(\bar{x}_1 \vee x_3)} \cdot \overline{(x_2 \vee \bar{x}_4)} \cdot \overline{(x_3 \vee x_4)}}$$

Далі користуючись правилом де Моргана ($\overline{a \cdot b} = \bar{a} \vee \bar{b}$), розділяємо одну з операцій інверсії

$$\overline{\overline{(\bar{x}_1 \vee x_3)} \cdot \overline{(x_2 \vee \bar{x}_4)} \cdot \overline{(x_3 \vee x_4)}} = \overline{\overline{(\bar{x}_1 \vee x_3)} \vee \overline{(x_2 \vee \bar{x}_4)} \vee \overline{(x_3 \vee x_4)}}$$

Операції між дужками диз'юнкція, над нею є інверсія, тобто це операція Пірса («АБО-НІ»)

$$\overline{\overline{(\bar{x}_1 \vee x_3)} \vee \overline{(x_2 \vee \bar{x}_4)} \vee \overline{(x_3 \vee x_4)}} = \overline{\overline{(\bar{x}_1 \vee x_3)} \downarrow \overline{(x_2 \vee \bar{x}_4)} \downarrow \overline{(x_3 \vee x_4)}}$$

Операції в дужках диз'юнкції, над кожною дужкою є інверсія, тобто це операції Пірса («АБО-НІ»)

$$= \overline{\overline{(\bar{x}_1 \vee x_3)} \downarrow \overline{(x_2 \vee \bar{x}_4)} \downarrow \overline{(x_3 \vee x_4)}} = \overline{(\bar{x}_1 \downarrow x_3) \downarrow (x_2 \downarrow \bar{x}_4) \downarrow (x_3 \downarrow x_4)}$$

Кожній операції на схемі відповідає логічний елемент «АБО-НІ». Тому схема складається з чотирьох елементів «АБО-НІ». На вхід трьох елементів «АБО-НІ», які відповідають операціям у дужках подаємо вхідні сигнали згідно виразу: на перший \bar{x}_1, x_3 ; на другий x_2, \bar{x}_4 ; на третій x_3, x_4 . Вихід цих елементів з'єднаний з входами елемента «АБО-НІ», що відповідає операції між дужками заданої функції (рис. 14).

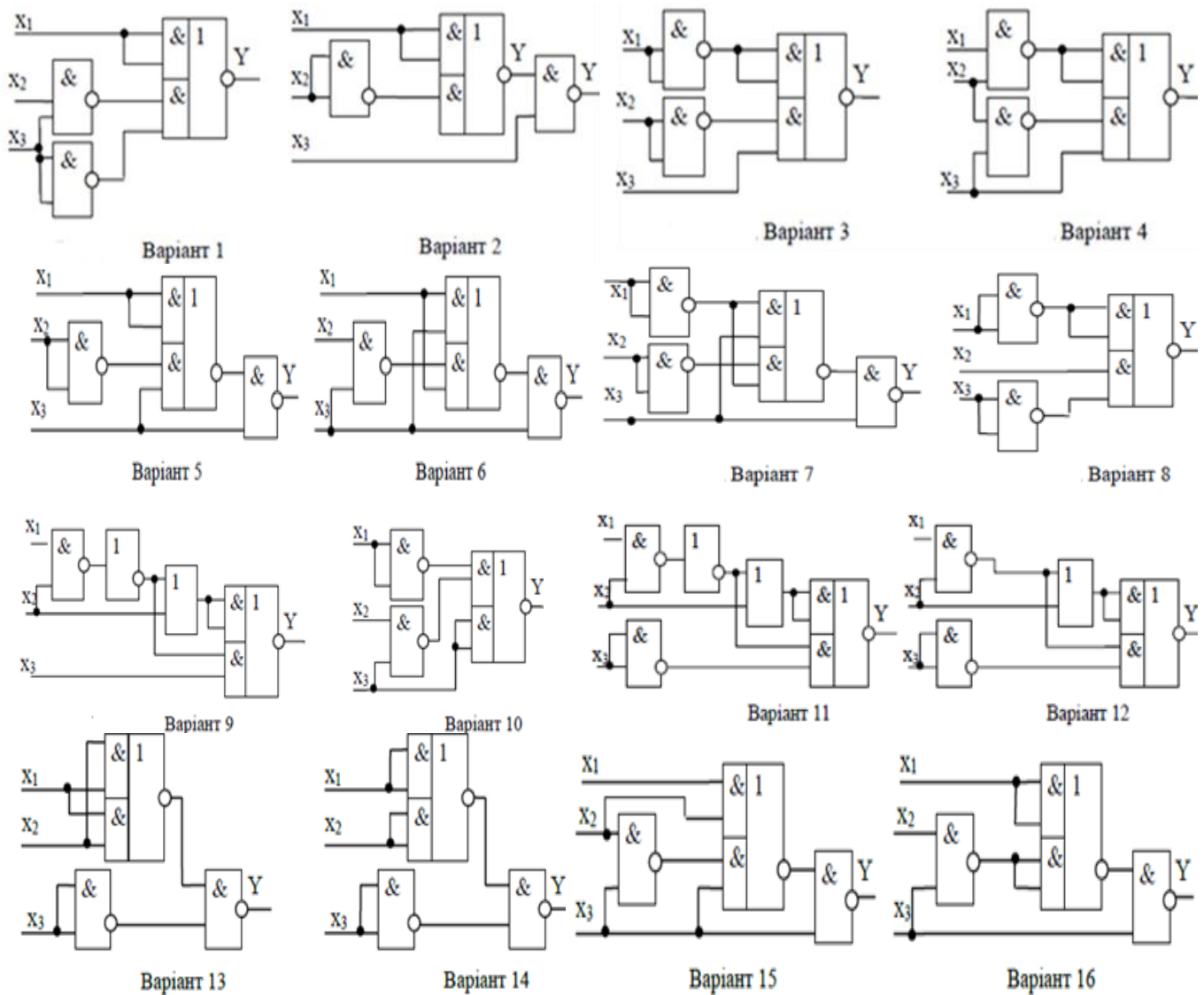


Рис. 1. Варіанти завдань логічних схем

За допомогою кнопки логічного конвертора у вигляді



отримати із логічного виразу нову схему. Аналітично перетворити значення логічної функції у базис І-НІ та перевірити отриманий вираз за допомогою кнопки логічного конвертора у вигляді



із побудованою схемою в базисі І-НІ.

Під'єднаємо до досліджуваної схеми генератор слів з позначенням у вигляді



Схема логічного пристрою із під'єднаними генератором слів та логічним аналізатором наведена на рис. 3.

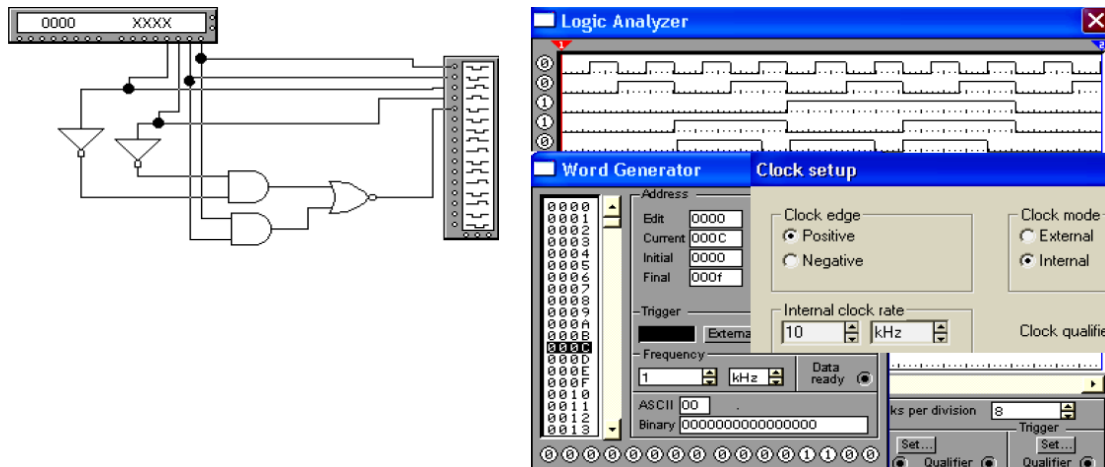


Рис.3. Схема логічної схеми із під'єднаними генератором слів та логічним аналізатором

У генератор слів у 16-річній системі числення необхідно ввести двійкові набори в кількості, значення яких відповідають кількості входів побудованої схеми.

За допомогою логічного аналізатора отримаємо часові діаграми функціонування схеми, які повинні відповідати таблиці істинності.

4. Дослідження функціонування логічних елементів у динамічному режимі роботи

Включити EWB. Вибрати логічний елемент згідно з табл. 4. Для цього необхідно натиснути на піктограму з іменем DIGIT. Із вікна, що з'явилося натисканням на мишку витягнути на робоче поле необхідну серію елементів, наприклад 74xx. Вибрати згідно з варіантом мікросхему та натиснути кнопку Ассерт. Для коректного проведення досліджень елементів з відкритими колекторами необхідно на вихід елемента навантажити резистор $R = 1\text{ кОм}$.

Зібрати електричну схему проведення дослідження з отриманим логічним елементом, як наведено на рис. 4. Вибрати в нижньому рядку піктограм піктограму з ім'ям Sources та витягнути на робоче поле зображення заземлення \perp та джерела напруги Vcc. Сигнал заземлення під'єднати до контакту GND мікросхеми, а джерело напруги Vcc – до контакту Vcc мікросхеми.

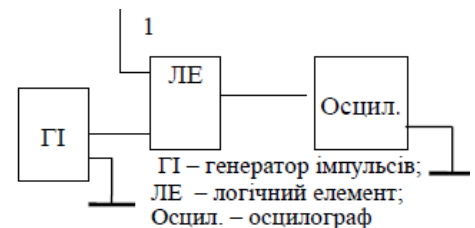


Рис. 4. Схема проведення дослідження елемента 2І-НІ

На один із входів логічного елемента подати сигнал з виходу генератора прямокутних імпульсів, а на останні входи – такі логічні сигнали, які дозволяють провести дослідження зміни сигналу на виході.

На виході генератора прямокутних імпульсів одержати позитивні імпульси з амплітудою $U_{вх} = 4\text{ В}$ та частотою згідно з варіантом із табл. 5.

Занести до звіту схему та результати досліджень згідно з прикладом (рис. 5).

Визначити тривалість фронтів t_F сигналу на виході, час затримки вихідного сигналу та занести результати до звіту. Тривалість імпульсу можливо отримати шляхом підведення лівої червоної вертикальної лінії до початку інтервалу сигналу, який вимірюються, та підведення синьої вертикальної лінії до закінчення інтервалу.

В правому нижньому віконці осцилографа відображається необхідний параметр сигналу.

Дослідження впливу ємності навантаження на форму вихідних сигналів.

Таблиця 4

№ з/п	Серія SN74	Вітчизняні МС	Функціональне призначення
1	7400	155ЛА3	4 елементи 2І-НІ
2	7402	155ЛЕ1	4 елементи 2АБО-НІ
3	7403	155ЛА9	4 елементи 2І-НІ з відкритим колектором
4	7408	155ЛІ1	4 елементи 2І
5	7409	155ЛІ2	4 елементи 2І з відкритим колектором
6	7410	155ЛА4	3 елементи 3І-НІ
7	7411	555ЛІ3	3 елементи 3І
8	7412	155ЛА10	3 елементи 3І-НІ з відкритим колектором
9	7420	155ЛА1	2 елементи 4І-НІ
10	7421	155ЛІ6	2 елементи 4І
11	7422	155ЛА7	2 елементи 4І-НІ з відкритим колектором
12	7426	155ЛА11	4 елементи 2І-НІ з відкритим колектором
13	7428	155ЛЕ5	4 елементи 2АБО-НІ
14	7430	155ЛА2	Елемент 8І-НІ
15	7432	155ЛІ1	4 елементи 2АБО
16	7438	155ЛА13	4 елементи 2І-НІ з відкритим колектором

Таблиця 5

№ з/п	Частота f, кГц	№ з/п	Частота f, кГц
1	0,5	9	1
2	1	10	2
3	2	11	3
4	4	12	4
5	5	13	5
6	6	14	6
7	7	15	7
8	8	16	9

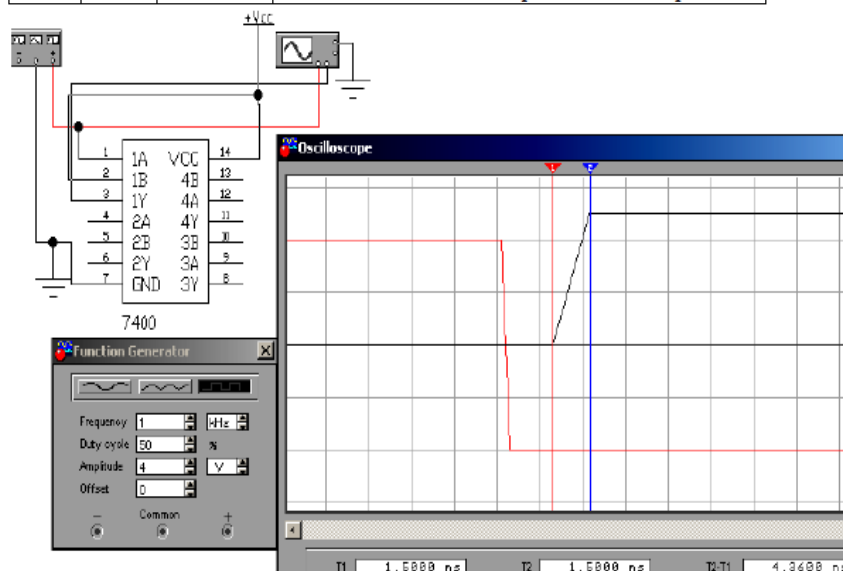


Рис.5. Схема та результати досліджень логічного елементу

Для цього до виходу логічного елемента підключити спочатку ємність $C=10\text{пФ}$. Навести схему, таблицю та графіки фронтів та спадів вихідного сигналу в залежності від різних ємностей. Схема дослідження та діаграми наведені на рис. 6.

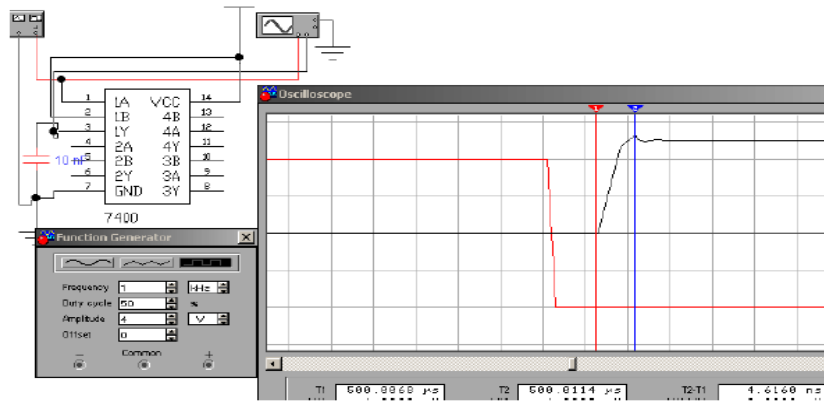


Рис.6. Схема та результати досліджень з ємністю $C = 10$ пФ

5. Контрольні питання

1. Що таке логічна функція?
2. Які існують способи подання логічних функцій?
3. Назвіть основні аксіоми та закони алгебри логіки.
4. Що називають функціонально повною системою логічних функцій?
5. Що таке кон'юнктивна та диз'юнктивна нормальні форми подання логічних функцій?
6. Коли КНФ та ДНФ вважаються досконалими?

Тема 4. Схемотехніка логічних елементів

Лабораторне заняття №4. Дослідження біполярних і польових транзисторів

Навчальна мета заняття:

1. Вивчення принципу роботи, статичних і динамічних характеристик, експериментального дослідження біполярних і польових транзисторів.
2. Закріплення теоретичного матеріалу, набуття навиків створення і моделювання схем аналогових пристроїв, робота з різними вимірювальними приладами.

Кількість годин: 4 год.

Література:

1. Методичні вказівки до лабораторних робіт з курсу «Моделювання на ЕОМ» для студентів напряму підготовки 6.051001 «Метрологія та інформаційновимірювальні технології» денного та заочного навчання / Уклад. Харченко О. Л., Балев В. М. Харків : НТУ «ХПІ», 2010. 68 с.
2. Крилик Л.В., Селецька О.О. Матеріали електронної техніки: навчальний посібник. Вінниця: ВНТУ, 2017. 120 с
3. Чешко І.В. Вступ до спеціальності «Електроніка» : навчальний посібник. Суми : Сумський державний університет, 2017. 148 с.
4. Квітка С.О., Яковлев В.Ф., Нікітіна О.В. Електроніка та мікросхемотехніка: Навчальний посібник / За ред. проф. В.Ф. Яковлева. Київ: Аграрна освіта, 2010. 329 с.
5. Болюх В. Ф., Данько В. Г. Основи електроніки і мікропроцесорної техніки: Навч. посібник. Харків: НТУ «ХПІ», 2011. 257 с.

Матеріально-технічне забезпечення: комп'ютерна мережа із системою модулювання електронних схем Electronics Workbench.

Навчальні питання

1. Загальні відомості про біполярні та польові транзистори
2. Визначення залежності вихідної напруги від вхідної для схеми із загальним емітером
3. Вимірювання коефіцієнта підсилення схеми із загальним емітером і негативним зворотним зв'язком по струму
4. Визначення залежності вихідної напруги від вхідної для схем із загальною базою і загальним колектором
5. Транзистор як джерело стабільного струму
6. Визначення залежності вихідної напруги від вхідної для схеми із загальним істоком та істокового повторювача
7. Керований дільник напруги
8. Контрольні питання

1. Загальні відомості про біполярні та польові транзистори

Транзистор – напівпровідниковий елемент із трьома електродами, що служить для посилення або перемикавання сигналу. Транзистор – активний елемент, що підсилює потужність електричного сигналу. Це посилення відбувається за рахунок енергії зовнішніх джерел живлення. Змінюючи струм у вхідному ланцюзі за певним законом, можна одержати посилений сигнал на виході тієї ж форми. Транзистори діляться на дві великі групи: біполярні і польові.

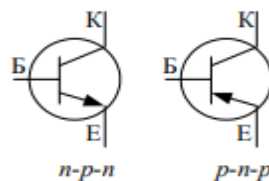


Рис. 1. Позначення біполярних транзисторів

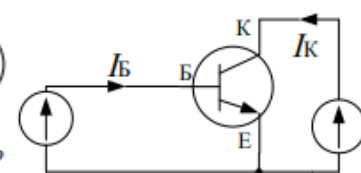


Рис.2. Полярність включення n-p-n транзистора

Біполярні транзистори. Розрізняють кремнієві і германієві транзистори. Вони бувають р-п-р і п-р-п типу, на рис. 1 показані їхні позначення. Біполярний транзистор можна розглядати як два протилежно включених діоди, які мають один загальний п- або р-шар. Електрод, пов'язаний з ним, називається базою (Б). Два інших електроди називаються емітером (Е) і колектором (К).

Основна особливість транзистора полягає в тому, що його колекторний струм є кратним базовому струму. Їхнє відношення $K = I_K / I_B$ називають коефіцієнтом підсилення по струму (рис. 2).

Іншою особливістю є те, що малої зміни вхідної напруги виявляється досить для того, щоб викликати відносно велику зміну колекторного струму. В електричне коло транзистор включають таким чином, що один з його виводів (електродів) є вхідним, другий – вихідним, а третій – загальним для вхідного і вихідного кіл (рис. 3).

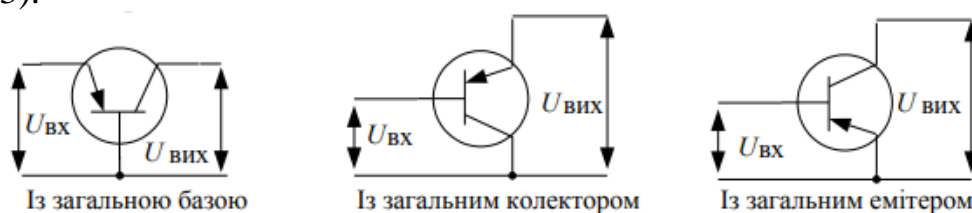


Рис. 3. Схеми включення біполярних транзисторів

Польовий транзистор управляється електричним полем, практично без витрати потужності управляючого сигналу. Серед польових транзисторів розрізняють шість типів, їхні умовні позначки в електричних принципових схемах представлені на рис. 4.

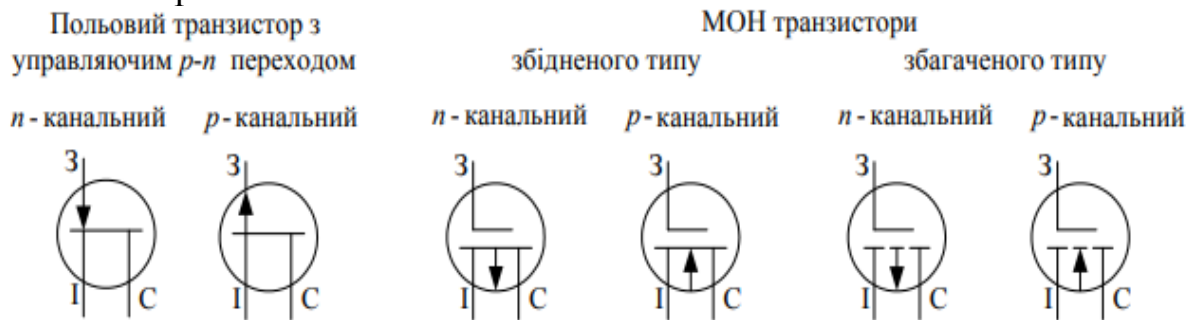


Рис. 4. Позначення польових транзисторів

Керуючим електродом транзистора є затвор 3. Він дозволяє управляти величиною опору між стоком С та истоком І. Керуючою напругою є напруга $U_{з1}$. Більшість польових транзисторів є симетричними, тобто їхні властивості не змінюються якщо електроди І і С поміняти місцями. Через польовий транзистор з управляючим переходом при напрузі $U_{з1} = 0$ протікає найбільший струм стоку. Такі транзистори називають нормально відкритими. Аналогічні властивості мають МОП транзистори збідненого типу. Навпаки, МОП транзистори збагаченого типу закриваються при величинах $U_{з1}$ близьких до нуля. Їх називають нормально закритими.

2. Визначення залежності вихідної напруги від вхідної для схеми із загальним емітером

Складіть схему, представлену на рис. 5. Змінюючи вхідну напругу від 0 до 4 В із кроком 0,2 В занесіть у табл. 1 значення вихідної напруги, за отриманими даними побудуйте залежність $U_{вих} = f(U_{вх})$.

3. Вимірювання коефіцієнта підсилення схеми із загальним емітером і негативним зворотним зв'язком по струму

Складіть схему, представлену на рис. 6. Змінюючи вхідний сигнал від 0 до 10В із кроком 1 В, занесіть у табл. 2

значення вихідної напруги, за отриманими даними побудуйте графік залежності вихідної напруги від вхідної.

Таблиця 2

$U_{вх}, \text{В}$	0	1	...	10
$U_{вих}, \text{В}$				

Таблиця 1

$U_{вх}, \text{В}$	0	0,2	...	4
$U_{вих}, \text{В}$				

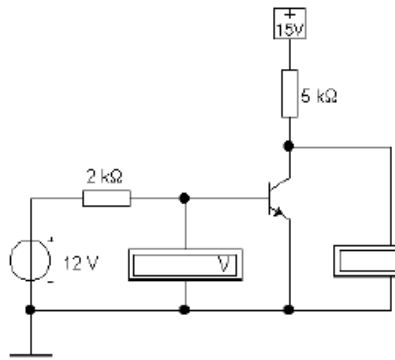


Рис.5. Схема с загальним емітером

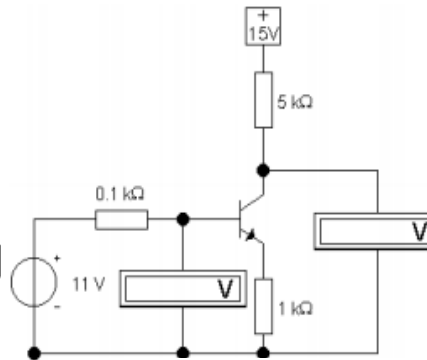
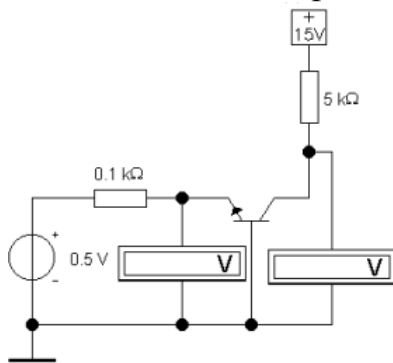


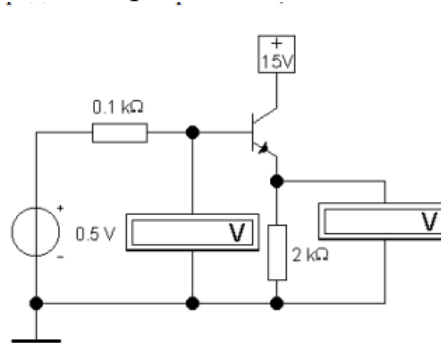
Рис. 6. Схема с загальним емітером та зв. зв'язком

4. Визначення залежності вихідної напруги від вхідної для схем із загальною базою і загальним колектором

Складіть схеми, представлені на рис. 7 а, б.



а



б

Рис. 7. Схем із загальною базою і загальним колектором

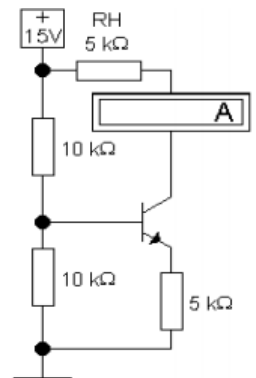


Рис. 8 Джерело струму на транзисторі

Повторіть операції з дослідів 2 для схем із загальною базою і загальним колектором, за результатами досліджень, побудуйте графіки $U_{вих} = f(U_{вх})$.

5. Транзистор як джерело стабільного струму

Ідеальне джерело струму забезпечує в навантаженні струм, що не залежить від опору навантаження. Схема джерела струму зображеного на рис. 8, побудована на основі схеми із загальним емітером і негативним зворотним зв'язком по струму. Навантаження включається послідовно із транзистором. Вихідний струм залишається незмінним поки транзистор насичений. Складіть схему представлену на рис. 8, змінюючи опір навантаження від 10 Ом до 5 кОм, переконайтеся в незмінності показань амперметра.

6. Визначення залежності вихідної напруги від вхідної для схеми із загальним истоком та истокового повторювача

Схема із загальним истоком відповідає схемі із загальним емітером для біполярного транзистора. Схема із загальним стоком (истоковий повторювач) має значно більший опір, чим схема із загальним истоком, крім того, вона істотно зменшує вхідну ємність каскаду (рис. 9).

Змінюючи вхідну напругу від 0 до 14 В із кроком 1. За отриманим даними побудуйте залежність $U_{вих} = f(U_{вх})$.

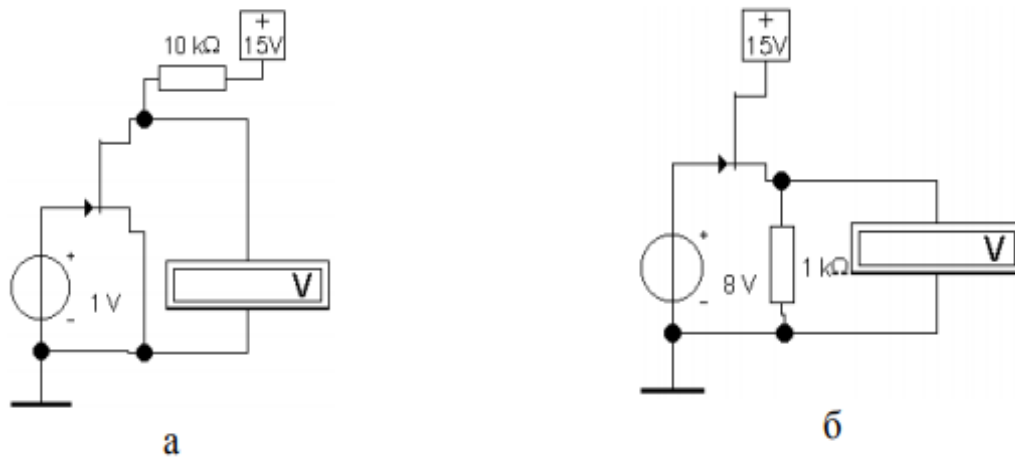


Рис. 9. Схема із загальним істком (а) та істоковий повторювач (б)

7. Керований дільник напруги

При малих напругах стік – істок вольтамперна характеристика така ж, як і в омичного опору, величину якого можна міняти в широких межах шляхом зміни напруги затвор – істок. Мінімальна напруга буде при $U_{ЗС} = 0$. Складіть схему, представлену на рис. 10. Змінюючи напругу, що подається на затвор транзистора в межах від 0 до 12 В виміряйте вихідну напругу і побудуйте графік отриманої залежності.

8. Контрольні питання

1. Назвіть основні типи напівпровідникових приладів та укажіть кількість р-п-переходів у них.

2. Що таке напівпровідниковий діод? Наведіть класифікацію та умовні позначення напівпровідникових діодів.

3. Що таке випрямний діод? Накресліть його умовне позначення і вольт-амперну характеристику. Де застосовуються випрямні діоди?

4. Що таке фото- і світлодіоди? Накресліть їх умовні позначення. Де використовують фото- і світлодіоди?

5. Що таке транзистор? Наведіть класифікацію та умовні позначення транзисторів.

6. Накресліть структуру та умовне позначення біполярних транзисторів р-п-р і п-р-п-типів. Позначте основні елементи.

7. Назвіть і наведіть схеми включення біполярного транзистора.

8. Зобразіть вхідну і вихідну вольт-амперні характеристики біполярного транзистора, включеного за схемою зі спільним емітером.

9. Накресліть структуру, умовне позначення і схему включення польового транзистора з керованими р-п-переходами.

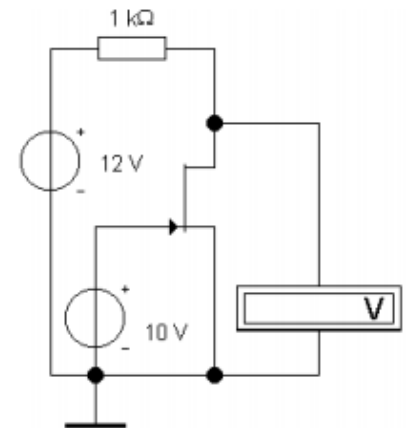


Рис. 10. Керований дільник напруги

Тема 5. Схемотехніка цифрових елементів

Лабораторне заняття №5. Дослідження функціонування тригерів

Навчальна мета заняття:

1. Закріплення теоретичного матеріалу, набуття навиків аналізу і синтезу основних схем тригерів.

2. Набуття навиків створення і моделювання схем цифрової схемотехніки, робота з різними вимірювальними приладами.

Кількість годин: 4 год.

Література:

1. Рябенський В.М. Жуйков В.Я. Ямненко Ю.С. Заграничний А.В. Схемотехніка: Пристрої цифрової електроніки. Київ: 2016, КПІ. 399 с.

2. Кравець В.О., Сокол Є.І., Рисований О.М. Комп'ютерна схемотехніка. Підручник. Харків: НТУ «ХПІ», 2007. 480 с.

Матеріально-технічне забезпечення: комп'ютерна мережа із системою модулювання електронних схем Electronics Workbench.

Навчальні питання

1. Аналіз функціонування RS-тригерів
2. Аналіз функціонування JK-тригерів
3. Аналіз функціонування T-тригерів
4. Дослідження роботи тригерів у статичному режимі
5. Контрольні питання

1. Аналіз функціонування RS-тригерів

Завдання 1. Побудувати структурну схему, таблицю станів, характеристичні рівняння тригера для асинхронного RS-тригера, виготовленого на логічних елементах 2АБО-НІ.

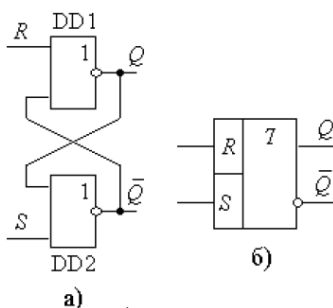


Рис. 1

Схема RS-тригера, зібраного на логічних елементах 2АБО-НІ, приведена на рис. 1,а, а його умовне позначення на рис. 1,б. Стан тригера часто ототожнюється з сигналом на прямому виході Q .

Повна таблиця станів RS-тригера (табл. 1) описує особливості його роботи. Звернемо увагу на те, що при $S_n=R_n=1$, незалежно від Q_n , стан виходів тригера є невизначеним.

Таблиця 1

R_n	S_n	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	x
1	1	1	x

Дійсно, якщо подати на обидва входи таку комбінацію сигналів, то на обох виходах з'являться логічні нулі ($Q_{n+1}=\bar{Q}_{n+1}=0$). Але якщо ці сигнали одночасно зняти, задавши $Q_{n+1}=\bar{Q}_{n+1}=0$, то стан виходів буде невизначеним. Це пов'язано з тим, що після зняття сигналів у кожному з логічних елементів буде протікати перехідний процес зміни станів, і результат його залежатиме від швидкодії елементів. Остання є величиною невизначеною. Тому розглянута комбінація входних сигналів називається *невизначеною комбінацією*, а перехід від такої комбінації до нульової – *забороненим переходом*. Фактично це означає, що при проектуванні цифрових пристроїв необхідно приймати міри для

$R_n S_n$	0	0	1	1
Q_n	0	1	x	0
1	1	1	x	0

Рис. 2

Таблиця 2

R_n	S_n	Q_{n+1}
0	0	Q_n
0	1	1
1	0	0
1	1	x

виключення подібних ситуацій.

Представивши табл. 1 у формі карти Карно (рис. 2), помічаємо наступну особливість логічної функції: при значеннях $S_n = R_n = 0$ маємо $Q_{n+1} = Q_n$, а для решти комбінацій входів значення виходу Q_{n+1} не залежить від значення Q_n . Це дає можливість мінімізувати табл. 1 (див. табл. 2) та одержати характеристичне рівняння тригера. Перетворюючи ці рівняння в базиси логічних функцій І-НІ. Цим рівнянням відповідає схема рис. 3. Перше з них виконане на DD1, а друге на DD2. Цей тригер працює в інверсних кодах.

$$Q_{n+1} = S_n + Q_n \cdot \bar{R}_n;$$

$$\bar{Q}_{n+1} = R_n + \bar{Q}_n \cdot \bar{S}_n;$$

$$Q_{n+1} = \overline{\bar{S}_n \cdot (Q_n \cdot \bar{R}_n)};$$

$$\bar{Q}_{n+1} = \overline{R_n \cdot (\bar{Q}_n \cdot \bar{S}_n)}.$$

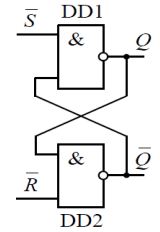


Рис. 3

Завдання 2. Побудувати таблицю станів для асинхронного RS-тригера, виготовленого на логічних елементах 2І-НІ (рис. 3), а також часові діаграми сигналів на входах і виходах при різних їх співвідношеннях.

Таблиця 3

\bar{S}_n	\bar{R}_n	Q_{n+1}
0	0	X
0	1	1
1	0	0
1	1	Q_n

Таблиця станів будується або на основі табл. 2, або на основі характеристичних рівнянь тригера і має вигляд табл. 3. Високі рівні входних сигналів не змінюють стану тригера, а низькі рівні одночасно на двох входах є забороненими для тригера, виготовленого на логічних елементах 2І-НІ. Установка (запис інформації) забезпечується низьким рівнем по входу \bar{S} , а обнуління – відповідно, низьким рівнем по входу \bar{R} . Часові діаграми, що пояснюють роботу тригера, приведені на рис. 4.

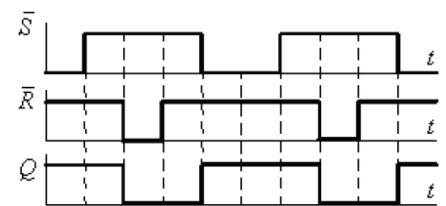


Рис. 4. Часові діаграми

Зміна станів тригерних схем при подачі різних послідовностей входних сигналів зображується за допомогою графа переходів (рис. 5).

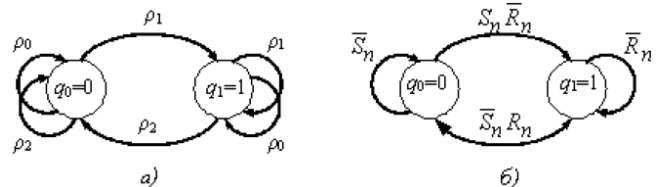


Рис. 5. Граф переходів

Розглянемо детальніше особливості побудови графа переходів тригера. Виходячи з таблиці станів, можемо

розглядати тригер як пристрій з одним виходом Q (Q_{n+1}), оскільки другий вихід \bar{Q} (\bar{Q}_{n+1}) є лише інверсією першого.

Вихід Q може приймати два значення. Позначимо їх $q_0 = 0$ і $q_1 = 1$ як дві вершини графа. Вектор впливаючих сигналів позначимо буквою ρ . Він може приймати, у відповідності до табл. 2, значення $\rho_0 = \bar{S}_n \bar{R}_n$, $\rho_1 = S_n \bar{R}_n$, $\rho_2 = \bar{S}_n R_n$, вектор ρ_0 не може змінювати стани тригера. Таку особливість на граф-схемі (рис. 5,а) зобразимо у вигляді дуги, що виходить з вершин і замикається на них. Вектор ρ_1 переводить тригер у стан $q_1 = 1$ і на граф-схемі зображується у вигляді дуги, що виходить з вершини q_0 і закінчується в q_1 . Якщо тригер знаходиться в стані q_1 , то впливаючий сигнал ρ_1 не змінить стану тригера. На граф-схемі це дуга, що замикається на стані q_1 . Аналогічно, впливаючий сигнал ρ_2 переводить тригер зі стану q_1 в q_0 , а в стані q_0 не впливає на нього. Оскільки перехід з q_1 в q_1 забезпечується двома сигналами з однаковим результатом, то поєднаємо їх як:

$$\rho_{01} = \rho_1 + \rho_0 = S_n \overline{R_n} + \overline{S_n} \overline{R_n} = \overline{R_n}.$$

Аналогічно маємо:

$$\rho_{02} = \rho_0 + \rho_2 = \overline{S_n} \overline{R_n} + \overline{S_n} R_n = \overline{S_n}.$$

В результаті граф-схема переходів RS-тригера прийме вигляд рис. 5,б.

Графи переходів відповідають часовим діаграмам, що ілюструють роботу тригера у часі. Крім того, часові діаграми дають більш детальну характеристику перехідним процесам у схемі.

На рис. 6 наведені часові діаграми роботи тригера, схему якого зображено на рис. 1,а. Елементи тригера перемикаються послідовно. Запуск тригера відбувається за фронтом вхідного сигналу S у момент часу, коли його значення досягне порогового рівня спрацьовування логічного елемента DD2. Вихідний стан $\bar{Q} = 1$ логічного елемента DD2 змінюється, і в момент часу, коли потенційний рівень спаду сигналу \bar{Q} зменшиться до рівня порогової напруги елемента DD1, останній починає перемикатись. На рис. 6 послідовність перемикання показана стрілками.

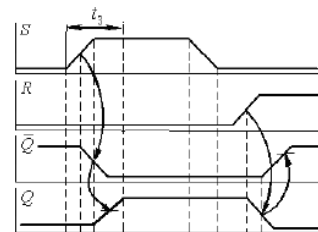


Рис. 6

Інтервал часу перемикання визначається з моменту t_1 початку дії сигналу S до моменту t_2 – завершення перемикання логічного елемента DD1. Цей інтервал часу t_3 , що зветься *часом затримки на перемикання*, дає можливість оцінити мінімальну тривалість вхідних сигналів, при якій гарантовано будуть змінюватися стани тригера.

З опису процесу перемикання витікає, що t_3 тригера визначається сумою часових затримок двох базових логічних елементів, на яких він виготовлений. У довідковій літературі для кожного тригера приводиться таблиця станів, аналіз якої дозволяє визначити всі режими роботи пристрою.

Завдання 3. Пояснити особливості роботи RS-тригера K564TP2 (аналоги провідних західних фірм-виробників – 4043BDC, CD4043AD, MC14043BAL, MN4043B, MSM4043B, SCB4043B, TC4043BP), таблиця станів якого додається (табл. 5.).

Таблиця 5

S_n	R_n	V	Q_{n+1}
0	0	1	Q_n
0	1	1	0
1	x	1	1
x	x	0	Z

У мікросхемі K564TP2 дозволяючий вхід V повинен мати високий рівень потенціалу для забезпечення робочого режиму схеми. При $V = 0$ вихід Q переходить у високоомний стан. Тригер виготовлений з використанням логічних елементів 2АБО-НІ і має лише один прямий вихід, оскільки при $S_n=R_n=V=1$ значення $Q_{n+1} = 1$ перебуває у незабороненому стані. При наявності двох виходів сигнали на їх виходах були б однаковими, що оцінювалося б як невизначеність.

RS-тригери у багатьох випадках використовуються як самостійні пристрої в тих ситуаціях, коли одним сигналом необхідно встановити якусь умову, а іншим – її зняти. Таке їх використання передбачається у контролерах і мікроконтролерах у складі регістрів ознак. Здебільшого RS-тригери використовуються у складі більш складних схем тригерів, модулів пам'яті. Вони знаходять широке використання в пристроях електронної автоматики.

Таблиця 6

S_n	R_n	C	D_n	Q_{n+1}
0	0	1	0	0
0	0	1	1	1
0	0	1	x	Q_n
0	1	x	x	0
1	0	x	x	1
1	1	x	x	$Q = \bar{Q}$

Завдання 4. Дати пояснення особливості роботи D-тригера K561TM2 (аналоги західних фірм-виробників – 4013BDM, CD4013AD) за допомогою таблиці станів (табл. 6).

Виходячи з перших двох рядків таблиці, бачимо, що маємо справу з динамічним тригером, який в синхронному режимі за фронтом синхроімпульсу забезпечує запис інформації з D-входу.

Третій рядок інформує про те, що за спадом синхроімпульсу тригер на значення D-входу не реагує. Четвертий і п'ятий рядки характеризують режим роботи асинхронного RS-тригера, виготовленого на ЛЕ 2АБО-НІ. Шостий рядок – заборонений асинхронний режим. Для роботи пристрою в режимі D-тригера асинхронні входи S і R необхідно заземлити.

2. Аналіз функціонування JK-тригерів

Завдання 5. Для JK-тригера, схема якого наведена на рис. 7, при початкових умовах $J = K = Q_1 = Q_2 = 0$ задається наступна послідовність входніх сигналів (рис. 8). Необхідно побудувати часові діаграми на виходах Q_1 , Q_2 тригера в інтервали часу, протягом яких були подані чотири синхросигнали C. У момент часу t_1 сигнал на вході J переходить з низького рівня в високий. Через інтервал часу t_2 , який повинен бути достатнім для усталення перехідних процесів на J-вході, подається сигнал C. Оскільки на вході DD1 в цей час маємо два сигнали високого рівня J і \bar{Q}_2 , то на виході DD1 сигнал \bar{S}_1 прийме низький рівень, в той час як вихід \bar{R}_1 ЛЕ DD2 матиме високий рівень логічного сигналу. Таке співвідношення сигналів \bar{S}_1 і \bar{R}_1 приведе до появи на виході Q_1 сигналу високого рівня.

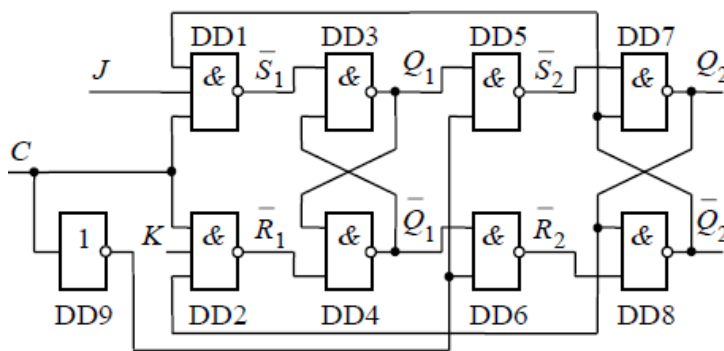


Рис. 7. Двотактний JK-тригер

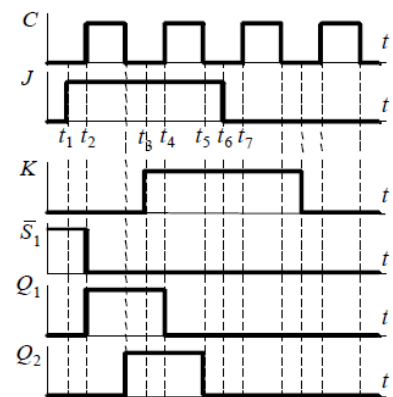


Рис. 8. Часові діаграми JK-тригера

Вказане розподілення рівнів напруг залишатиметься протягом часу тривалості синхроімпульсу. При спаді синхроімпульсу на виході DD9 з'явиться високий рівень сигналу, який призведе до зміни стану DD5 і, відповідно, перезапису сигналу Q_1 на вихід Q_2 . У момент t_3 з'являється сигнал високого рівня на вході K. На цей час $Q_2 = 1$ і, відповідно, при $K = 1$, $Q_2 = 1$ поява в t_4 другого синхроімпульсу приведе до зміни стану DD2, \bar{R}_1 стане рівним нулю і, відповідно, $\bar{Q}_1 = 1$, $Q_1 = 0$. За спадом другого синхроімпульсу низький рівень перезапишеться на вихід Q_2 . У подальшому, при дії синхроімпульсів рівні сигналів не змінюються.

3. Аналіз функціонування Т-тригерів

Завдання 6. Провести аналіз роботи Т-тригерів відповідно до таблиці істинності.

Таблиця 7

Т-тригер		\bar{T} -тригер	
T_n	Q_n	T_n	Q_n
0	Q_n	0	Q_n
1	\bar{Q}_n	1	\bar{Q}_n
1	Q_n	1	Q_n

До тригерів Т-типу, як вказувалось раніше, відносяться такі схеми, які за сигналом на Т-вході переключаються у протилежний стан. Це зазвичай тригери з динамічним Т-входом або з динамічним С-входом і статичним Т-входом.

У зв'язку з тим, що вони легко можуть бути організовані на базі динамічних D- або JK-тригерів, як самостійні мікросхеми Т-тригери не виготовляються.

У зв'язку з їх широким використанням в лічильниках імпульсів, Т-тригери з динамічним Т-входом часто називаються тригерами з лічильним входом, або лічильними тригерами. В залежності від характеру дії Т-динамічного входу, вони поділяються на Т-тригери, які спрацьовують за фронтом Т-імпульсу, та \bar{T} -тригери, що спрацьовують за зрізом Т-імпульсу.

У табл. 6 приведений перелік можливих станів обох типів тригерів. З таблиці витікає, що у Т-тригерах, на відміну від раніше розглянутих схем, стан виходу Q_{n+1} залежить не від значень інформаційних сигналів, а визначається тільки їх станом у попередньому такті. Рівняння роботи асинхронного та синхронного Т-тригерів має вигляд:

$$Q_{n+1} = Q_n \bar{T}_n + \bar{Q}_n T_n. \quad Q_{n+1} = (Q_n \bar{T} + \bar{Q}_n T) C + \bar{C} Q_n.$$

Т-тригер легко можна одержати з JK- або D-тригерів. На рис. 9 приводяться приклади створення схем Т-тригерів. Варіанти взаємних перетворень тригерів, приведені на рис. 9 розкривають велику гнучкість і широкі можливості різноманітних тригерних схем.

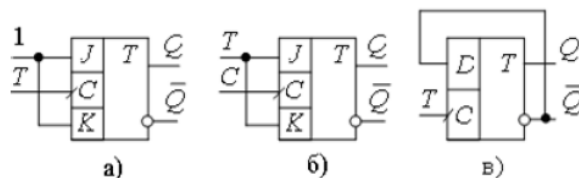


Рис. 9. Схеми Т-тригерів

4. Дослідження роботи тригерів у статичному режимі

Зібрати схему тригера на логічних елементах 2І-НІ згідно з варіантом (табл. 8) у пакеті EWB 5.12 та провести його дослідження.

Таблиця 8

№	Завдання	№	Завдання
1.	RS-тригер асинхронний	8.	TV-тригер синхронний
2.	Т-тригер асинхронний	9.	TV-тригер асинхронний
3.	TV-тригер асинхронний	10.	JK-тригер асинхронний
4.	RS-тригер синхронний	11.	JK-тригер синхронний
5.	Т-тригер синхронний	12.	DV-тригер синхронний
6.	D-тригер синхронний	13.	D-тригер синхронний
7.	DV-тригер синхронний	14.	RS-тригер асинхронний

У зв'язку з тим, що програма EWB програмно емулює функціонування дискретних логічних елементів, то при дослідженні тригерів вона вносить деякі помилки – робить самозбудження тригерів. Тому доцільно використати дискретні логічні елементи, які виконані у вигляді закінчених мікросхем, наприклад, на МС 7400 (4 елементи 2І-НІ).

Наприклад, структурна схема D-тригера на основі 4 логічних елементів 2І-НІ наведено на рис. 10, а її реалізація на основі МС 7400 наведена на рис. 11. При дослідженні такого тригера доцільно не використовувати початковий стан генератора слів, який дорівнює 0000, тому що програма EWB також

1. Рябенський В.М. Жуйков В.Я. Ямненко Ю.С. Заграничний А.В. Схемотехніка: Пристрої цифрової електроніки. Київ: 2016, КПІ. 399 с.

2. Кравець В.О., Сокол Є.І., Рисований О.М. Комп'ютерна схемотехніка. Підручник. Харків: НТУ «ХПІ», 2007. 480 с.

Матеріально-технічне забезпечення: комп'ютерна мережа із системою модулювання електронних схем Electronics Workbench.

Навчальні питання

1 Побудова дешифраторів та шифраторів

2 Дослідження функціонування мультиплексорів, дешифраторів

3 Контрольні питання

1. Побудова дешифраторів та шифраторів

Завдання 1. Побудувати структурну схему повного шифратора на 4x2 у базисі І-НІ.

Побудувати структурну схему повного шифратора на 4x2 у базисі І-НІ.

Повний двійковий шифратор має n виходів і 2^n входів, де n - розрядність двійкового входу. збудженого розряду унітарного коду. З метою розв'язання цієї задачі складемо таблицю істинності (відповідності) між вхідним 4-розрядним унітарним кодом $\{x_3, x_2, x_1, x_0\}$ та вихідним 2-розрядним кодом $\{y_1, y_0\}$ (табл. 1).

Таблиця 1

x_3	x_2	x_1	x_0	y_1	y_0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Кожен з розрядів вихідного двійкового коду y_1, y_0 будемо розглядати як перемикальну функцію, яка залежить від вхідних змінних, тобто $y_i = f_i(x_3, x_2, x_1, x_0)$. Записуючи її в ВДНФ, одержимо

$$y_1 = x_3 x_2 x_1 x_0 \vee x_3 x_2 x_1 \overline{x_0} = K_2(v) \vee K_3(v),$$

$$y_0 = \overline{x_3} \overline{x_2} \overline{x_1} \overline{x_0} \vee \overline{x_3} \overline{x_2} x_1 \overline{x_0} = K_1(v) \vee K_3(v).$$

Тому що відповідно до умови задачі водночас може бути збудженим тільки один із входів, наприклад $x_3 = 1$, то достовірно відомо, що $\overline{x_2} = \overline{x_1} = \overline{x_0} = 1$. Це означає, що $K_3(v) = x_3 \overline{x_2} \overline{x_1} \overline{x_0} = x_3$. Розповсюджуючи ці твердження і до інших конститuent 1, що входять у попередні вирази, можемо записати:

$$y_1 = x_3 \vee x_2 = \overline{\overline{x_3} \overline{x_2}},$$

$$y_0 = x_3 \vee x_1 = \overline{\overline{x_3} \overline{x_1}}.$$

Схема шифратора, що реалізує ці вирази, та її умовне графічне позначення зображені на рис. 1.

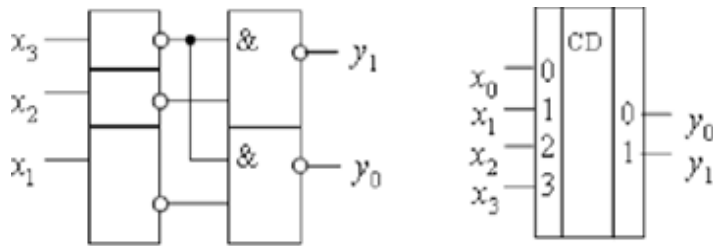


Рис. 1. Шифратор 4x2 та його умовне графічне позначення

Завдання 2. Побудувати двохступінчатий п'ятірозрядний дешифратор на базі трьохрозрядних дешифраторів.

П'ятірозрядний дешифратор повинен мати $2^5 = 32$ виходів. Розділимо п'ять розрядів на молодші x_2, x_1, x_0 і старші x_4, x_3 . Тоді молодші можна подати на входи чотирьох 3-розрядних дешифраторів другого каскаду і сформувати $8 \cdot 4 = 32$ виходу. Використовуючи входи дозволу EN, можна вибирати один з чотирьох дешифраторів другого каскаду, на якому повинен сформуватися одиничний сигнал. Для цього старші два розряди подаємо на входи керуючого дешифратора першого каскаду, а його виходи підключимо до входів дозволу EN дешифраторів першого каскаду (рис.2).

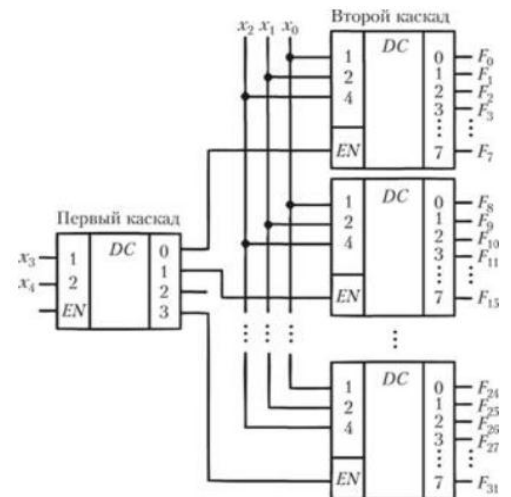


Рис. 2. Схема нарощування розмірності дешифратора

Нехай, наприклад, вхідний код дорівнює $11011 = 27_{10}$. Так як старші розряди – "11", то керуючий дешифратор дозволить роботу 4-го дешифратора другого каскаду. При цьому на виходах перших трьох дешифраторів будуть нулі, а на виході "3" четвертого дешифратора, тобто F_{27} буде логічна одиниця.

Дешифратор широко застосовуються в системах управління технологічними процесами. Багато виконавчі пристрої, такі, як електродвигун, виконавчий механізм на основі електромагніту, можуть управлятися всього двома командами: "включити" і "вимкнути". При цьому команді "включити" зручно зіставити логічну "1", а команді "вимкнути" – логічний "0". Для управління такими пристроями використовують унітарні коди, в яких кожен розряд жорстко пов'язаний з певним пристроєм. Кількість керованих пристроїв може складати кілька десятків, і дешифратор повинен мати відповідне число виходів.

Завдання 3. Побудувати мультиплексор 8 в 1, якій передає інформацію з одного із восьми входів на один вихід.

У реальні схеми мультиплексорів вводять керуючі дешифратори, число виходів яких дорівнює числу інформаційних входів. Вихідний сигнал дешифратора (логічна 1) включає відповідний

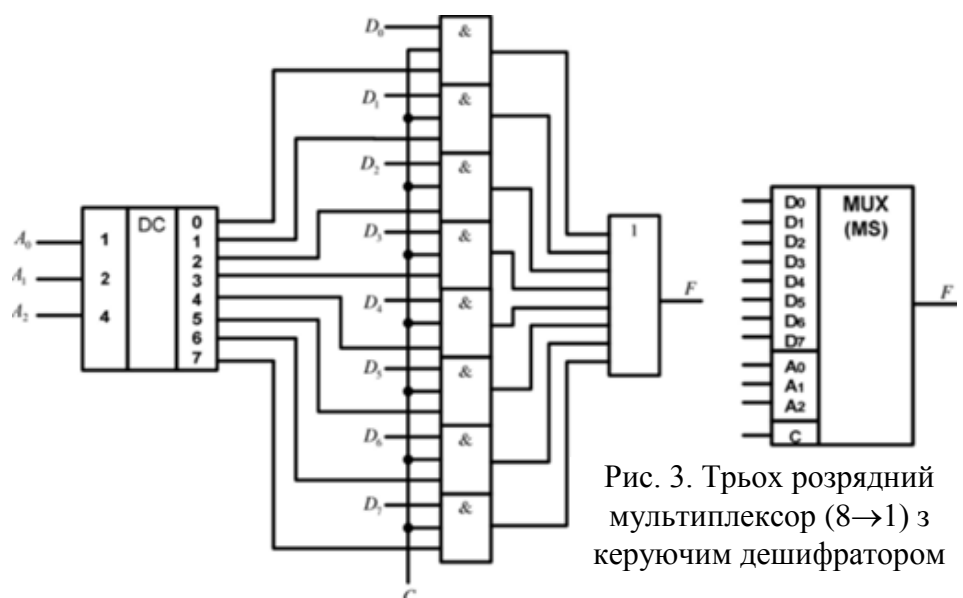


Рис. 3. Трьох розрядний мультиплексор (8→1) з керуючим дешифратором

логічний елемент І, причому в будь-який момент часу може бути відкритий тільки один логічний елемент.

При необхідності в схему мультиплексора може бути введений тактовий сигнал С. Варіант схеми 3-х розрядного мультиплексора (8 в 1) з керуючим дешифратором і його УГО наведені на рис. 3.

Завдання 4. Побудувати демультиплексор для передавання з одного вхідного каналу в один з чотирьох каналів-приймачів «із 1 в 4».

Демультиплексором називається функціональний вузол комп'ютера, призначений для комутації (перемикання) сигналу з одного інформаційного входу D на один з n інформаційних виходів. Номер виходу, на який в кожний такт машинного часу передається значення вхідного сигналу, визначається адресним кодом $A_0, A_1, A_2, \dots, A_{m-1}$. Адресні входи m та інформаційні входи n пов'язані співвідношенням $n=2^m$ або $m=\log_2 n$. На рис. 4 наведені функціональна схема (а) та умовне позначення (б) демультиплексора «із 1 в 4».

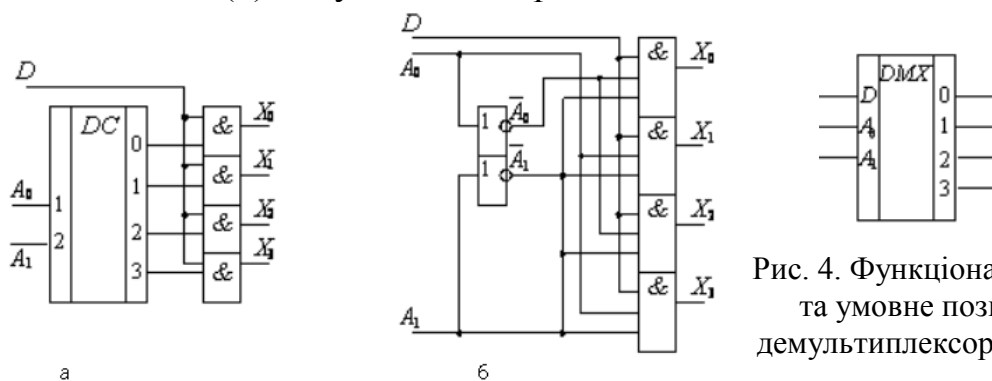


Рис. 4. Функціональна схема та умовне позначення демультиплексора «із 1 в 4»

2. Дослідження функціонування мультиплексорів, дешифраторів

2.1. Дослідження функціонування селектора-мультиплексора.

У відповідності з варіантом (табл. 2) вибрати селектор-мультиплексор.

Таблиця 2

№ з/п	Серія SN74	Вітчизняні МС	Функціональне призначення
1	7400	K155ЛА3	Побудувати мультиплексор (2→1) на лог. елементах
2	74150	155КП1	Селектор-мультиплексор 16×1
3	74151	155КП7	Селектор-мультиплексор 8×1
4	74153	155КП2	2 селектора-мультиплексора 4×2
5	74157	533КП16	4-розряд. селектор-мультиплексор 2×1
6	74158	1533КП18	4-розряд. селектор-мультиплексор 2×1 з інвер.
7	74251	155КП15	Селектор-мультиплексор 8×1 з 3-ма станами
8	74253	155КП12	2 селектора-мультиплексора 4×1 з 3-ма станами
9	74257	155КП11	4 селектора-мультиплексора 2×1 з 3-ма станами
10	74258	155КП14	4 селект.-мультиплек. 2×1 з 3-ма станами та інвер.
11	74298	155КП13	4 2-входових мультиплексора з запам'ятанням
12	74352		2 Здвоєний MUX 4×1 з прямими виходами
13	74353	555КП17	2 Здвоєний MUX 4×1 з інвер. та 3-ма станами вих.
14	74157	533КП16	Побудувати мультиплексор (4 → 1) на 74157
15	7400	K155ЛА3	Побудувати мультиплексор $8 \rightarrow 1$ на лог. елементах
16	74151	155КП7	Побудувати мультиплексор $16 \rightarrow 1$ на 74151

У програмі схемотехнічного моделювання аналогових та цифрових радіоелектронних пристроїв Electronics Workbench зібрати електричну схему проведення дослідження з цифровим пристроєм з використанням генератора слів, багатоканального осцилографа та підключити до виходу пристрою світлодіод, як показано, наприклад, на рис. 5.

Вивід G' позначає інверсний вхід дозволу вибору елемента. Вибір ІМС можна провести двома шляхами: або натиснути на піктограму з назвою Digital, а потім перетягнути на робоче поле піктограму з ім'ям MUX, вибрати згідно з варіантом свою ІМС та натиснути кнопку Ассерт, або вибрати піктограму з ім'ям Digit, натиснути на піктограму класу ІМС, перетягнути його на робоче поле, вибрати згідно з варіантом свою ІМС та підтвердити свій вибір кнопкою Ассерт.

На схемі для візуального контролю генератора слів паралельно йому підключений 7-сегментний індикатор з внутрішнім дешифратором.

Підключити до всіх входів даних всі 0 (рис. 5). Зняти осцилограму функціонування ІМС.

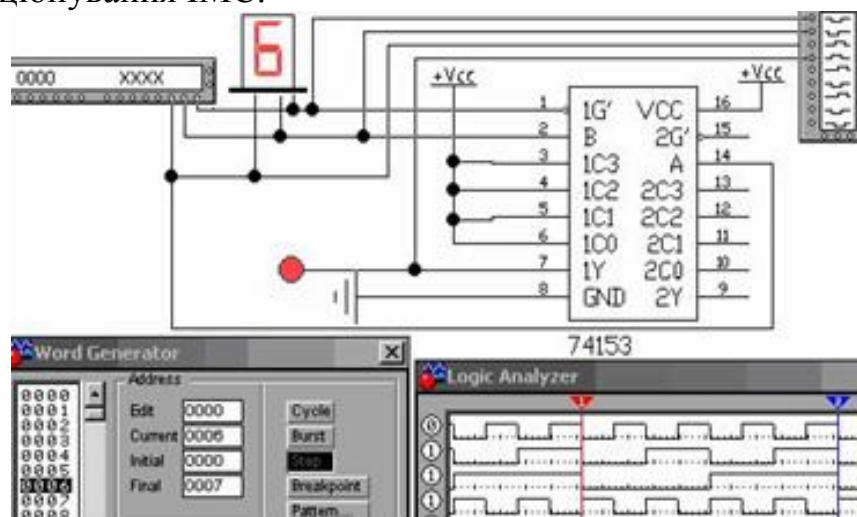


Рис. 5. Схема дослідження ІМС 74153

Підключити до всіх входів даних всі 1. По осцилограмі функціонування ІМС зробити висновки щодо змін, які з'явилися. Також виміряти час затримки вихідного сигналу відносно вхідних. Додати цю характеристику до звіту.

Виводи ІМС мають таке призначення:

- VCC – напруга живлення +5 В для ТТЛ;
- Vdd – напруга живлення для КМОП-логіки;
- GND – загальний для ТТЛ;
- USS – загальний для КМОП-логіки;
- I, A, B, C, ... – входи;
- Y, O – виходи;
- W – інверсний вихід;
- G' – вхід дозволу (активний рівень – низький).

Для складніших ІМС визначення функціонального призначення їх виводів доцільно проводити шляхом зіставлення з вітчизняними аналогами. Для більш оперативної орієнтації при роботі з цифровими ІМС приводиться перелік найбільш

поширених мнемонічних позначень на їх функціональних схемах і в таблицях станів:

- A = B (Parity) – вихід рівності операндів A і B;
- A/S (Asynchro/Synchro) – вхід асинхронного і синхронного режимів;
- B/D (Binary/Decimal) – вхід перемикавання рахунку з двійкового на десятковий;
- (Clock input) – вхід тактових імпульсів;
- CD (Count down) – вхід тактових імпульсів на зменшення рахунку (у реверсивних лічильниках);
- Ci (Count up) – вхід тактових імпульсів на збільшення рахунку;
- CE (Clock enable) – вхід дозволу для тактових імпульсів;
- CEP (Count enable parallel) – вхід паралельного нарощування розрядів лічильника;
- CET (Count enable trickle) – вхід дозволу рахунку при нарощуванні розрядів лічильника;
- CLR (Clear) – вхід скидання;
- C (Carry in) – вхід для розряду перенесення;
- CS (Chip select) – вибір кристала; визначає доступ до однієї з ІМС пристрою;
- D (Data input) – вхід даних тригера, лічильника, регістра;
- DSI (Data serial input) – вхід послідовних даних;
- DS (Data select) – вхід вибору даних;
- DL, DR (Data left, Data right) – входи для послідовного завантаження (регістра) зліва, справа;
- DSL, DSR (Data shift left, Data shift right) – входи для зрушення даних вліво, вправо;
- E (Enable) – вхід сигналу дозволу;
- EC (Enable count) – вхід сигналу дозволу рахунку;
- EE (Enable even) – вхід сигналу дозволу, рахунковий;
- EI (Enable input) – вивід ІМС, за яким дається дозвіл на прийом даних;
- EIO (Enable input/output) – вивід для одночасного дозволу по входу і виходу;
- EO (Enable output) – вивід для дозволу по виходу;
- LSB (Least significant bit) – молодший значущий розряд (МЗР). М (Mode control) – вибір режиму “Арифметики-логіка” в АЛП;
- PE (Parallel enable load) – вхід дозволу паралельного завантаження;
- P/S (Parallel/serial) – вхід перемикавання режимів паралельного або послідовного завантаження;
- R (Reset) – асинхронне скидання даних;
- RE (Read enable) – вхід дозволу читання;
- S (Set) – установка тригера, лічильника, регістра;
- S (Set enable) – дозвіл попереднього паралельного запису;
- SI (Serial input) – вхід послідовний;
- SIR, SIL (Serial input right, SI left) – вхід послідовний справа, зліва;
- SR (Synchro reset) – вхід скидання синхронно з тактовим імпульсом;
- TC (Terminal count) – вихід закінчення рахунку;

- TCD (Terminal count down) – те ж на зменшення рахунку;
- TCU (Terminal count up) – те ж на збільшення рахунку.

Наприклад, для позначення мультиплексорів найбільш часто використовуються символи:

– для 4-канального мультиплексора 74153 (K155КП2): А, В – адресні входи, 1G, 2G – інверсні входи дозволу першого і другого мультиплексорів, 1C0...1C3 і 2C0...2C3, 1Y і 2Y – входи і виходи першого і другого мультиплексорів відповідно;

– для 4-канальних і 2-канальних мультиплексорів 74298 (K555КП13): А1, А2; В1, В2; С1, С2; D1, D2 – входи однойменних 2- канальних мультиплексорів, QА, QВ, QС, QD – виходи відповідних мультиплексорів, CLK – синхросигнал запам'ятовування результату, WS – сигнал вибору напрямку прийому інформації з першого або другого каналу.

Занести до звіту схему та результати досліджень згідно з варіантом та схему внутрішньої структури цифрового пристрою, яку теоретично може мати пристрій, що досліджується.

2.2. Дослідження функціонування дешифратора.

У відповідності з варіантом табл. 3, вибрати дешифратор.

Таблиця 3

№ з/п	Серія	Вітчизняні МС	Функціональне призначення
1	Generic	–	2-to-4 DEMUX
2	Generic	–	3-to-8 DEMUX
3	7442	555ИД6	DC 4 × 10
4	7445		
5	7447		
6	74138	155ИД7	Дешифратор-демультиплексор 3 × 8
7	74139	155ИД14	2 дешифратори-мультиплексора 2 × 4
8	74145	155ИД10	Двійково-десятковий DC з відкр. колектором
9	74154	155ИД3	Дешифратор-демультиплексор 4 × 16
10	74155	155ИД4	2 дешифратори-мультиплексора 2 × 4
11	74156	555ИД5	2 DC-DMUX 2 4 з відкритим колектором
12	74159		
13	74445		
14	4028	561ИД1	Двійково-десятковий DC
15	4514		
16	4515		

Зібрати схему дослідження дешифратора, під'єднати до нього генератор слів, світлодіод та 7-сегментний індикатор з дешифрацією адреси вхідних слів згідно з рис. 6. Разом зі світлодіодами можна підключити й багатоканальний осцилограф та прослідити зміну станів у динаміці.

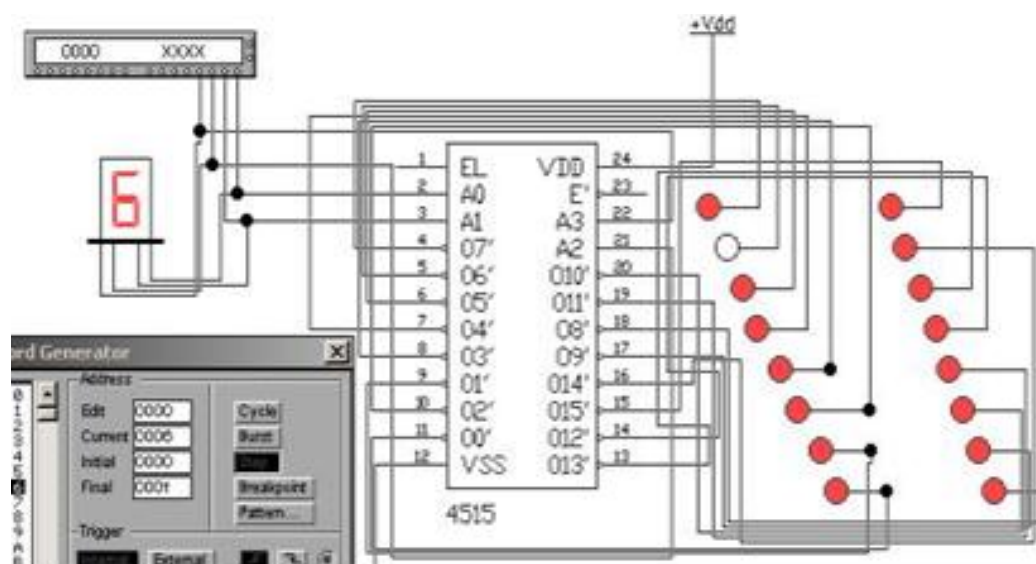


Рис. 6. Схема включення ІМС 4515

3. Контрольні питання

1. Які типи дешифраторів та шифраторів ви знаєте та чим обумовлено їх різноманіття?
2. Наведіть функціональне призначення дешифраторів та шифраторів.
3. Приведіть таблиці істинності повного дешифратора, шифратора та рівняння їх функціонування.
4. Обґрунтуйте використання неповних дешифраторів да дешифраторів з інверсними виходами.
5. Поясніть, практичне використання дешифраторів та шифраторів
6. Що таке мультиплексор та демультимплексор, та яке їх призначення?
7. Які функції виконує цифровий компаратор та в яких пристроях він може використовуватись?
8. Яке значення мають формувачі парності та де вони можуть використовуватись?

Тема 11. Комбінаційні цифрові вузли

Лабораторне заняття №7. Дослідження функціонування кодоперетворювачів та комбінаційних суматорів

Навчальна мета заняття:

1. Закріплення теоретичного матеріалу, набуття навиків аналізу і синтезу основних типових комбінаційних цифрових вузлів.
2. Набуття навиків створення і моделювання схем цифрової схемотехніки, робота з різними вимірювальними приладами.

Кількість годин: 4 год.

Література:

1. Рябенкий В.М. Жуйков В.Я. Ямненко Ю.С. Заграничний А.В. Схемотехніка: Пристрої цифрової електроніки. Київ: 2016, КПІ. 399 с.
2. Кравець В.О., Сокол Є.І., Рисований О.М. Комп'ютерна схемотехніка. Підручник. Харків: НТУ «ХПІ», 2007. 480 с.

Матеріально-технічне забезпечення: комп'ютерна мережа із системою модулювання електронних схем Electronics Workbench.

Навчальні питання

- 1 Побудова кодоперетворювачів двійкового коду
- 2 Дослідження функціонування комбінаційних суматорів
- 3 Контрольні питання

1. Побудова кодоперетворювачів двійкового коду

Завдання 1. Побудувати схему перетворювача 4-х розрядного двійкового коду X_1, X_2, X_3, X_4 з розрахунком значення знакового розряду X_{zn} в обернений $Y_{zn}, Y_1, Y_2, Y_3, Y_4$.

В процесі перетворення прямого коду в обернений значення знакового розряду X_{zn} використовується як керуючий сигнал, який забезпечує отримання наступного виразу:

$$Y_i = \overline{X_{zn}} X_i \vee X_{zn} \overline{X_i} = X_{zn} \oplus X_i$$

Схема перетворювача прямого коду в обернений, побудовану на елементах "Виключальне АБО", показано на рис. 1

Завдання 2. Побудувати схему перетворювача 4-х розрядного двійкового коду X_1, X_2, X_3, X_4 з розрахунком значення знакового розряду X_{zn} в доповняльний $Y_{zn}, Y_1, Y_2, Y_3, Y_4$.

Доповняльний код додатного двійкового числа збігається з його прямим і оберненим кодами. Доповняльний код від'ємного двійкового числа утворюється з його оберненого коду додаванням до молодшого розряду одиниці. Таким чином, операція перетворення прямого коду в доповняльний код не є порозрядною і виконується значно складніше, ніж отримання оберненого коду.

Відповідність між прямим і доповняльним кодами на прикладі чотирьох цифрових розрядів представлено в табл. 1.

Таблиця 1

Прямий код				Доповняльний код				Прямий код				Доповняльний код			
X_4	X_3	X_2	X_1	Y_4	Y_3	Y_2	Y_1	X_4	X_3	X_2	X_1	Y_4	Y_3	Y_2	Y_1
0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	1	1	1	1	1	1	0	0	1	0	1	1	1
0	0	1	0	1	1	1	0	1	0	1	0	0	1	1	0
0	0	1	1	1	1	0	1	1	0	1	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0
0	1	0	1	1	0	1	1	1	1	0	1	0	0	1	1
0	1	1	0	1	0	1	0	1	1	1	0	0	0	1	0
0	1	1	1	1	0	0	1	1	1	1	1	0	0	0	1

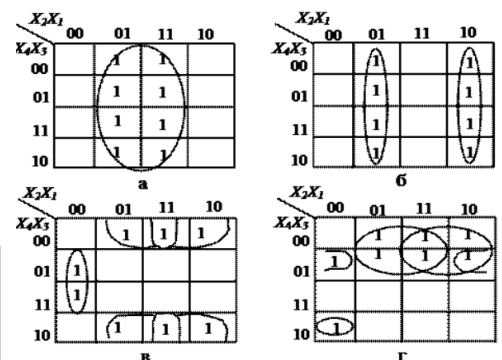


Рис. 1. Карта Карно для функцій а) – Y_1 ; б) – Y_2 ; в) – Y_3 ; г) – Y_4

На основі карт Карно (рис. 1) з урахуванням знакового розряду X_{zn} прямого коду для функцій $Y_1 Y_2 Y_3 Y_4$, що представляють виходи перетворювача, отримуємо:

$$Y_{3H} = X_{3H}; Y_1 = X_1; Y_2 = X_2 \oplus X_1 X_{3H}; Y_3 = X_3 \oplus (X_2 \vee X_1) X_{3H}$$

$$Y_4 = X_4 \oplus (X_3 \vee X_2 \vee X_1) X_{3H}$$

Схему перетворювача прямого коду в доповняльний код на основі виразів показано на рис. 2. Даний перетворювач характеризується високою швидкодією. Час встановлення вихідного коду визначається трьома затримками поширення сигналу, проте зі зростанням номера розряду лінійно зростає і необхідне число входів елементів АБО, що використовуються у схемі.

Завдання 3. Побудувати перетворювачі кодів дешифратор – шифратор, в якому кількість входів дорівнює кількості входів дешифратора (x_i), а кількість виходів (f_i). Схема перетворювача кодів дешифратор – шифратор подано на рисунку нижче відповідно до даних таблиці.

x_1	x_2	f_1	f_2	f_3
0	0	0	0	0
0	1	0	0	1
1	0	0	1	1
1	1	1	1	1

дешифратор – шифратор	
0	0
1	1
2	3
3	7

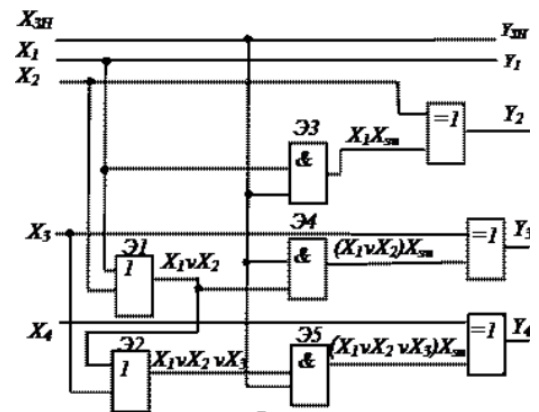
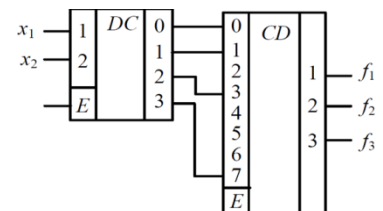


Рис. 2. Перетворювач прямого коду в доповняльний



Додаткові завдання для побудови перетворювача кодів дешифратор – шифратор.

№	Входи дешифратора	Виходи шифратора
1	2	000,001,011,101, 111
2	2	000,010,100,110
3	4	0000,0001,0011,0101,...,1111
4	3	0000,0010,0100,0110,...,1110
5	3	Парні 4-х розрядного коду
6	3	Непарні 4-х розрядного коду
7	3	Непарні 4-х розрядного коду
8	3	Парні 4-х розрядного коду

2. Дослідження функціонування комбінаційних суматорів

Провести дослідження суматорів згідно з варіантами (табл. 2).

Таблиця 2

№	Завдання на дослідження
1	3-розрядний СМ на повних суматорах
2	2-розрядний СМ на напівсуматорах
3	2-розрядний СМ з використанням елементів М2
4	4-розрядний СМ на повних суматорах
5	3-розрядний СМ на напівсуматорах
6	3-розрядний СМ з використанням елементів М2
7	5-розрядний СМ на повних суматорах
8	4-розрядний СМ на напівсуматорах
9	4-розрядний СМ з використанням елементів М2
10	6-розрядний СМ на повних суматорах

11	5-розрядний СМ на напівсуматорах
12	5-розрядний СМ з використанням елементів M2
13	7-розрядний СМ на повних суматорах
14	6-розрядний СМ на напівсуматорах
15	6-розрядний з використанням елементів M2

У програмі EWB арифметичні суматори наведені в бібліотеці Comt I двома базовими пристроями, наведеними на рис. 4, напівсуматором і повним суматором.

Вони мають наступні призначення виводів: А, В – входи операндів; Σ – результат додавання; Co – вихід переносу; Ci – вхід переносу. Багаторозрядний суматор створюється на базі одного напівсуматора і n повних суматорів. Як прикладна рис. 5 наведена структурна схема 2-розрядного суматора. На входи A1, A2 і B1, B2, подаються перші і другі доданки відповідно, а з виходів S1 та S2 знімається результат додавання.



Рис. 4. Зображення напівсуматора та повного суматора

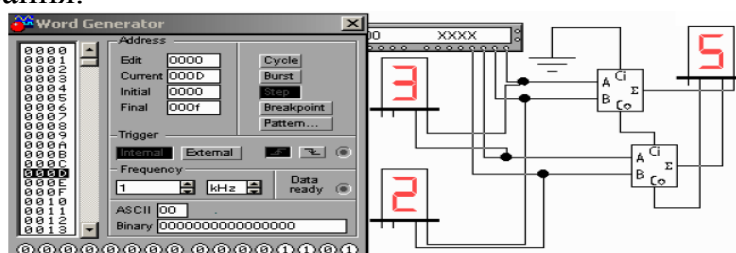


Рис. 5. Схема дослідження 2-розрядного суматора

Схема повного суматора однорозрядного на напівсуматорах наведена на рис.6, а схема повного однорозрядного суматора на елементах mod2 наведена на рис. 7.

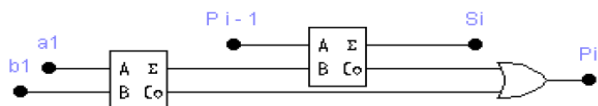


Рис. 6. Схема повного суматора однорозрядного на напівсуматорах

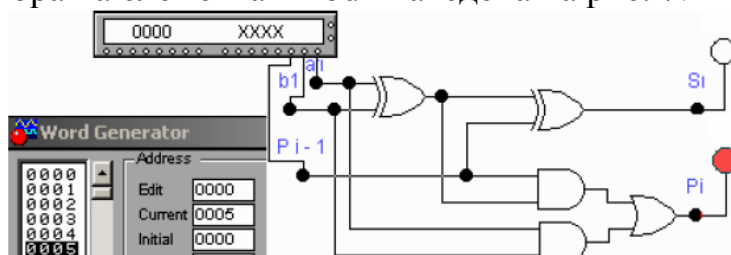


Рис. 7. Схема повного однорозрядного суматора на елементах mod2

Напівсуматори та логічні елементи mod2 завжди спрощують складність схем, які виконані з використанням комбінаторної логіки.

3. Контрольні питання

1. Які функції виконує кодоперетворювач та області його використання?
2. Чим відрізняється напівсуматор від повного суматора?
3. Кодування чисел: прямий, зворотний, додатковий двійкові коди.
4. Додавання і вирахування двійкових чисел у додатковому, зворотному кодах.
5. Вкажіть переваги і недоліки двійкових суматорів з послідовним переносом.

Тема 7. Схемотехніка цифрових вузлів

Лабораторне заняття №8. Дослідження функціонування регістрів, лічильників

Навчальна мета заняття:

1. Закріплення теоретичного матеріалу, набуття навиків аналізу і синтезу основних типових цифрових вузлів.

2. Набуття навиків створення і моделювання схем цифрової схемотехніки, робота з різними вимірювальними приладами.

Кількість годин: 4 год.

Література:

1. Рябенський В.М. Жуйков В.Я. Ямненко Ю.С. Заграничний А.В. Схемотехніка: Пристрої цифрової електроніки. Київ: 2016, КПІ. 399 с.

2. Кравець В.О., Сокол Є.І., Рисований О.М. Комп'ютерна схемотехніка. Підручник. Харків: НТУ «ХПІ», 2007. 480 с.

Матеріально-технічне забезпечення: комп'ютерна мережа із системою модулювання електронних схем Electronics Workbench.

Навчальні питання

1 Побудова типових цифрових вузлів: регістрів, лічильників

2 Дослідження функціонування регістра

3 Дослідження функціонування лічильників

4 Контрольні питання

1. Побудова типових цифрових вузлів: регістрів, лічильників

Завдання 1. Побудувати 3-розрядний паралельний регістр на RS тригерах з вхідними та вихідними ланцюгами записи та читання інформації.

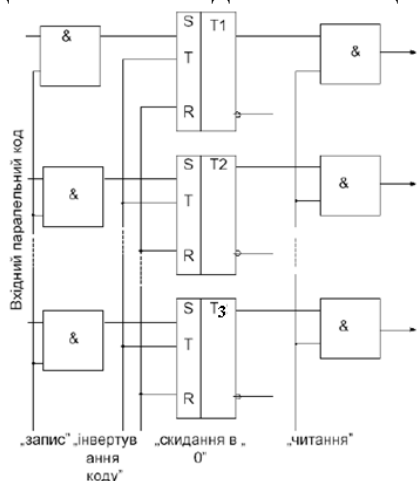


Рис. 1. Загальна схема паралельного регістру

В паралельних регістрах без зсуву інформація (двійкові числа слова) записуються одночасно до всіх розрядів (паралельний код). Загальна схема паралельного регістру зображена на рис.1.

Завдання 2. Побудувати схему 3-розрядного регістра, який створено на D-тригерах та виконує операції: установка в "0"; зсув слова на один розряд вліво; зсув слова на один розряд вправо; паралельний прийом інформації.

Поставимо у відповідність типам операцій значення управляючих сигналів: 00 – установка в "0"; 01 – зсув управо на один розряд; 10 – зсув уліво на один розряд; 11 – паралельний прийом інформації. Для виконання послідовного управління регістром можна використати як дешифратор, так і мультиплексор. Схема багатофункціонального регістра, призначеного для виконання необхідних операцій з використанням дешифратора наведена на рис. 2.

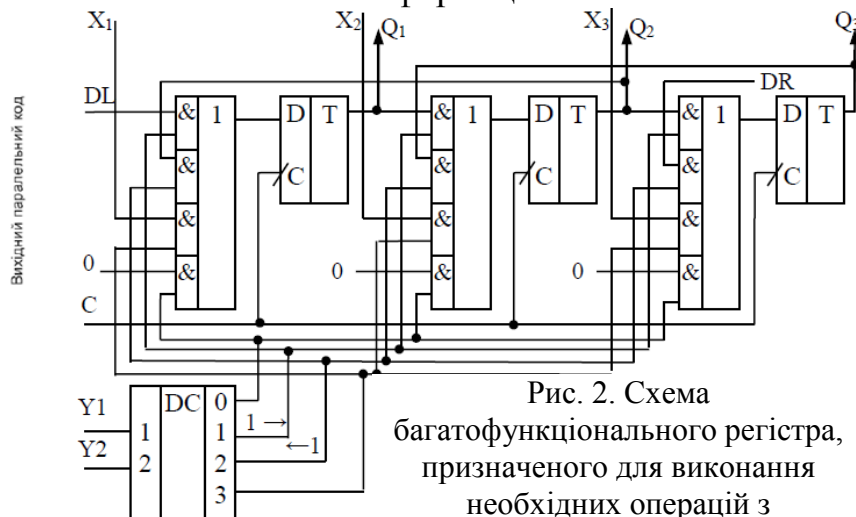


Рис. 2. Схема багатофункціонального регістра, призначеного для виконання необхідних операцій з використанням дешифратора

Завдання 3. Побудувати реверсивний чотирьохрозрядний лічильник на Т-тригерах.

Реверсивні лічильники можуть працювати в режимах як додавання, так і віднімання. Прикладом реверсивного лічильника може служити лічильник К155ІЕ7, який наведено на рис.3. В його схемі передбачені два входи для лічильних імпульсів, які позначаються С+ та С-. В режимі додавання імпульси повинні надходити на вхід С+, а в режимі віднімання - на вхід С-, причому на незадіяному вході в цей момент повинен бути сигнал високого рівня. Даний лічильник має два виходи для сигналів перенесення: вихід >15 для сигналу прямого перенесення, який використовується при роботі лічильника в режимі додавання та вихід <0 для сигналу зворотнього перенесення, який використовується при роботі лічильника в режимі віднімання. Обидва виходи перенесення використовують при нарощуванні розрядності лічильників.

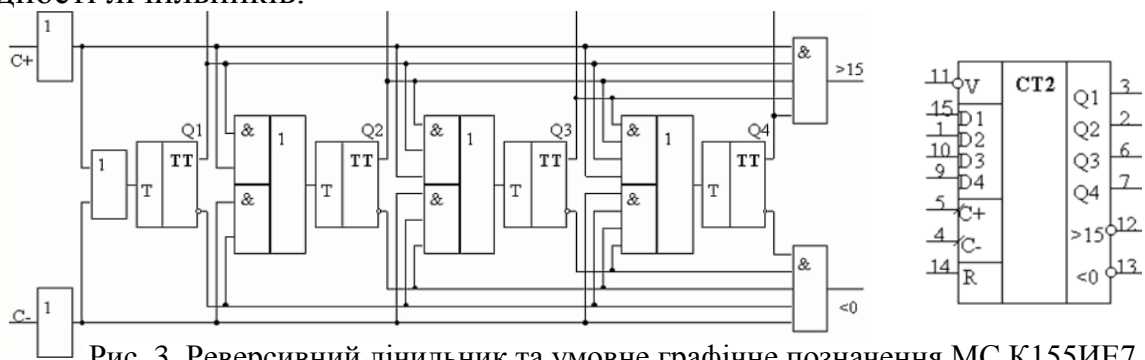


Рис. 3. Реверсивний лічильник та умовне графічне позначення МС К155ІЕ7

2. Дослідження функціонування регістра

Дослідження функціонування регістра, який побудовано на конкретних мікросхемах. У відповідності з варіантом з табл. 1 вибрати мікросхему та побудувати регістр. Навести функціональну схему та часові діаграми його функціонування.

Таблиця 1

№	Серія SN74	Вітчизняні МС	Функціональне призначення
1	7472	155ТВ1	JK-тригер з елементом 3І на виходах
2	7474	155ТМ2	2 D-тригери
3	7475	155ТМ7	D-тригери із прямими та інверсними виходами
4	7476	155ТВ7	JK-тригери
5	7477	155ТМ5	D-тригери із прямими виходами
6	7478	133ТВ14	JK-тригери
7	74107	155ТВ6	JK-тригери із роздільною установкою нуля
8	74109	155ТВ15	JK-тригери
9	74112	155ТВ9	JK-тригери
10	74113	155ТВ10	JK-тригери із поперед. установкою нуля та одиниці
11	74114	155ТВ11	JK-тригери із поперед. установ. 0 або 1 та "обнулінням -
12	74175	155ТМ8	D-тригери
13	74173	155ІР15	4-розр. регістр із трьома станами
14	74194	155ІР11	4-розр. універсальний регістр
15	74195	155ІР12	4-розр. регістр зсуву із паралельним введенням
16	74395	533ІР25	4-розр. паралельний регістр зсуву

Приклад заповнення генератора слів та схема дослідження паралельного регістра на JK-тригерах по входах попередньої установки наведено на рис. 4.

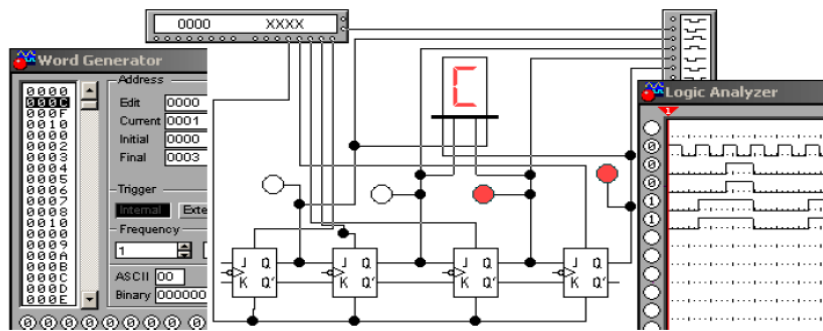


Рис. 4. Схема дослідження регістра на JK-тригерах по входах попередньої установки

Приклад дослідження схеми послідовного регістра з послідовним записом коду в регістр зсуву по інформаційних входах наведено на рис. 5.

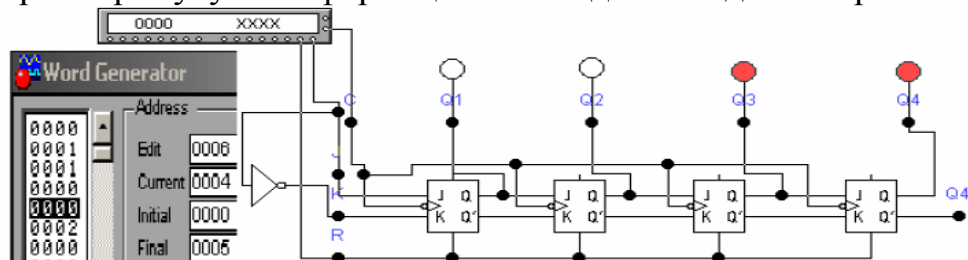


Рис. 5. Схема дослідження регістра зсуву на JK-тригерах з послідовним записом

3. Дослідження функціонування лічильників

Дослідження лічильників в інтегральному виконанні. Зібрати схему дослідження 4-розрядного асинхронного двійкового лічильника МС 7493 (155ІЕ5) як наведено на рис. 3. Змінити схему рис. 6 згідно з варіантом (табл. 2) та навести часові діаграми. Логічна структура мікросхеми К155ІЕ5 наведена на рис. 7.

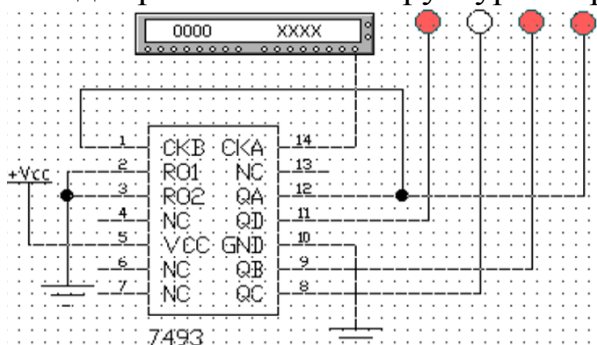


Рис. 6. Схема дослідження МС 7493

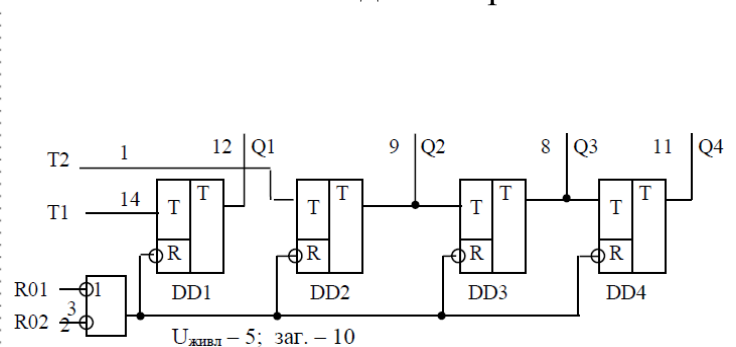


Рис. 7. Логічна структура мікросхеми К155ІЕ5

Таблиця 2

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
К _{лч}	3	4	5	6	7	8	9	10	11	12	13	14	15	6	9	11

4. Контрольні питання

1. Що таке регістр та які функції він може виконувати?
2. Наведіть типи регістрів та їх можливе використання.
3. Побудувати багатфункціональний регістр, який виконує операції: паралельний прийом інформації; зсув ліворуч на 1 розряд; скидання регістра.
4. Побудувати 3-розрядний лічильник на додавання на синхронних Т-тригерах та навести часові діаграми.

5. Побудувати 3-розрядний лічильник на віднімання на синхронних Т-тригерах та навести часові діаграми.

6. Побудувати 3-розрядний лічильник на додавання на D-тригерах та навести часові діаграми.

7. Побудувати 3-розрядний лічильник на віднімання на D-тригерах та навести часові діаграми.

Тема 8. Напівпровідникові запам'ятовуючі пристрої

Лабораторне заняття №9. Дослідження постійної та оперативної пам'яті комп'ютера

Навчальна мета заняття:

1. Закріплення теоретичного матеріалу, набуття навиків побудови та функціонування модулів пам'яті.

2. Набуття навиків створення і моделювання схем цифрової схемотехніки, робота з різними вимірювальними приладами.

Кількість годин: 4 год.

Література:

1. Рябенський В.М. Жуйков В.Я. Ямненко Ю.С. Заграничний А.В. Схемотехніка: Пристрої цифрової електроніки. Київ: 2016, КПІ. 399 с.

2. Кравець В.О., Сокол Є.І., Рисований О.М. Комп'ютерна схемотехніка. Підручник. Харків: НТУ «ХПІ», 2007. 480 с.

Матеріально-технічне забезпечення: комп'ютерна мережа із системою модулювання електронних схем Electronics Workbench.

Навчальні питання

1 Дослідження функціонування модуля оперативної пам'яті

2 Дослідження функціонування модуля постійної пам'яті

3 Контрольні питання

1. Дослідження функціонування модуля оперативної пам'яті

Створити комірку оперативної пам'яті. Підключити на вхід схеми генератор слова, а на вихід – світлодіод чи лампочку. Оформити її у вигляді підсхеми (субблока).

Лабораторна робота виконується по групах. Кожна група повинна побудувати комірки пам'яті статичного та постійного ЗП, а на основі цих субблоків зібрати та дослідити модулі пам'яті згідно з варіантами за табл. 1.

Таблиця 2

№ з/п	ОЗП		ПЗП	
	Ряд.	Стовп.	Ряд.	Стовп.
1	2	2	5	2
2	2	3	4	4
3	2	4	4	3
4	3	2	4	2
5	3	3	3	4
6	3	4	3	3
7	4	2	3	2
8	4	3	2	4
9	4	4	2	3
10	5	2	2	2

Схема комірки пам'яті з елементами керування наведена на рис. 1. Для зображення підписів елементів треба виділити елемент (клацнути лівою клавішею миші). Виділений елемент повинен стати червоного кольору. Навести стрілку курсора на елемент та натиснути праву клавішу миші. Вибрати

підменю Component Properties. В ньому вибрати піктограму з назвою Lebel. Ввести назву елемента. Вікно програми EWB 5.12 зі схемою комірки статичної пам'яті зображене на рис. 2.

Для запису в комірку пам'яті на відповідній розрядній шині даних встановлюється 1 або 0, на вході WR/RD – сигнал 1 та після стробування лічильника або дешифратора адреси сигналом CS включаються елементи 2І: U1, U2. Перепад сигналу з елемента U2 поступає на тактовий вхід D-тригера U4, в результаті чого в нього записується 1 або 0 в залежності від рівня сигналу на його D-вході.



Рис. 1. Схема комірки пам'яті з елементами керування

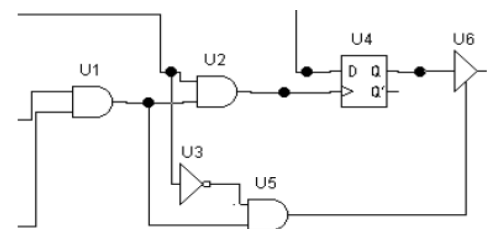


Рис. 2. Вікно програми EWB 5.12 зі схемою комірки статичної пам'яті

При читанні з комірки пам'яті на вході WR/RD встановлюється 0, при цьому спрацьовують елементи U1, U3, U5 та на вхід ДОЗВІЛ ВИХОДУ буферного елемента U6 поступає сигнал дозволу, в результаті чого сигнал Q-виходу D-тригера передається на розрядну шину даних DO0 ... DO3. Для перевірки функціонування комірки пам'яті використовується генератор слова.

Створити модуля оперативної пам'яті. Створити, в залежності від варіанта, матрицю n-бітного ОЗП. Для дослідження функціонування матриці підключити відповідні елементи.

Відповідно до варіанта завдання побудуйте модуль n-бітного ОЗП (табл. 1). Для цього потрібно виділити з допомогою лівої клавіші миші таку область схеми, яку б не потрапили елементи, які не належать до неї. В результаті виконання команди викликається діалогове вікно, в стрічці Name якого треба ввести ім'я підсхеми. Після цього можливі варіанти:

Copy from Circuit – підсхема копіюється з назвою в бібліотеку Custom;

Move from Circuit – виділена частина вирізається з схеми та у вигляді підсхеми копіюється в бібліотеку Custom;

Replace in Circuit – виділена частина замінюється в схемі підсхемою та копіюється в бібліотеку Custom.

На рис. 3 наведена комірка статичної пам'яті з використанням підсхеми, а на рис. 4 показана матриця 16-бітового статичного ОЗП: вона

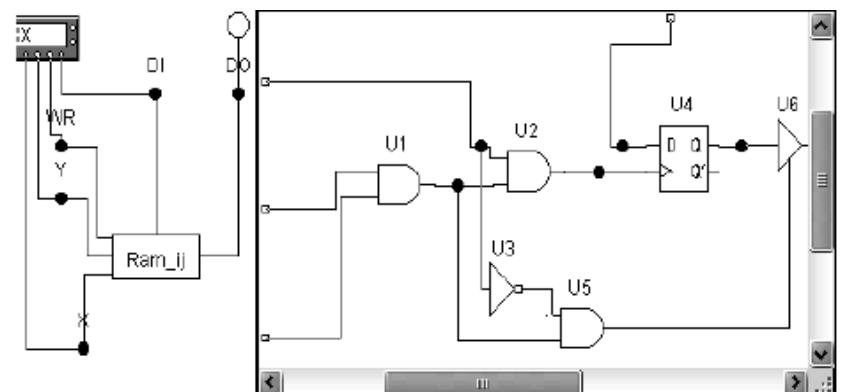


Рис. 3. Комірка статичної пам'яті з використанням підсхеми

складається з 16 комірок пам'яті.

Кожна комірка пам'яті адресується по входах X, Y шляхом вибору дешифраторами адресних ліній за рядками $Ax0 - Ax3$ і за стовпцями $Ay0 - Ay3$ і подачі по обраних лініях сигналу логічної одиниці.

При цьому в обраній комірці пам'яті спрацьовує двовходовий елемент І (U1), який підготовлює ланцюг читання-запису інформації на вхідних $DI0...DI3$ чи вихідних $DO0...DO3$ розрядних шинах. Сигналом для видачі адреси є CS (chip select — вибір кристала), який подається на вхід дозволу лічильника адреси (Addr_cnt) або такий же вхід дешифраторів, підключених до виходів лічильника.

Для перегляду підсхеми редагування потрібно двічі клацнути мишею по її значку. Редагування підсхеми здійснюється за загальними правилами редагування схем. При створенні додаткового виводу необхідно з відповідної точки підсхеми курсором миші протягнути провідник до краю її вікна до появи незафарбованої прямокутної контактної площадки, після чого відпустити ліву кнопку миші. Для видалення виводу необхідно курсором миші схопитися за його прямокутну площадку в краю вікна підсхеми і винести її за межі вікна.

Конструктивно будь-яке ОЗП складається з двох блоків — матриці запам'ятовуючих елементів і дешифратора адреси. З технологічних міркувань матриця найчастіше має двокоординатну дешифрацію адреси — за рядками і за стовпцями.

2. Дослідження функціонування модуля постійної пам'яті

Створити, в залежності від варіанта, матрицю запам'ятовуючого масиву n-бітного ПЗП. Для дослідження функціонування матриці підключити відповідні елементи.

Порядок побудови модуля ПЗП аналогічний до порядку побудови модуля ОЗП. Схема нагромаджувача ПЗП наведена на рис. 5.

На першому етапі створення схеми ПЗП необхідно створити схему

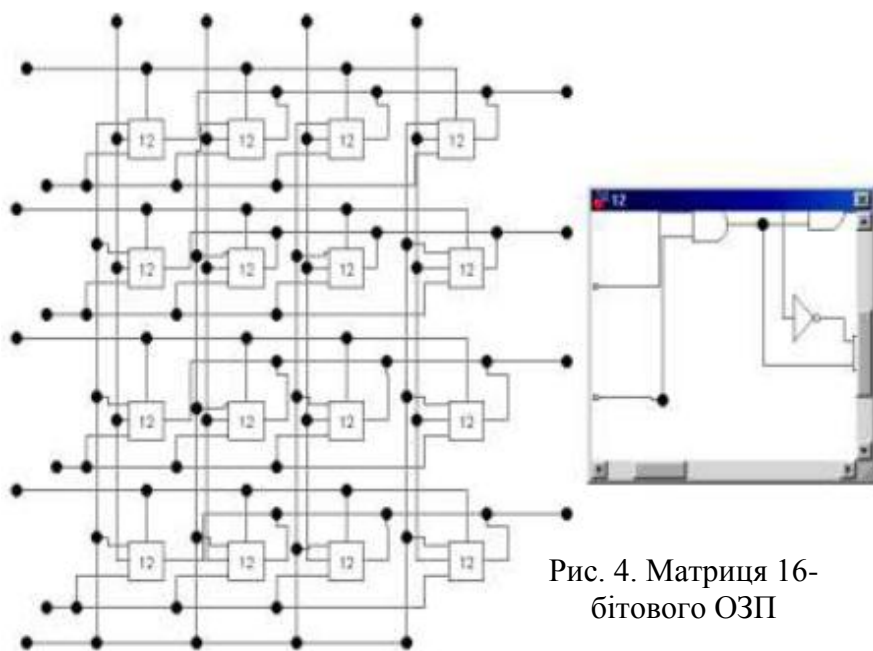


Рис. 4. Матриця 16-бітового ОЗП

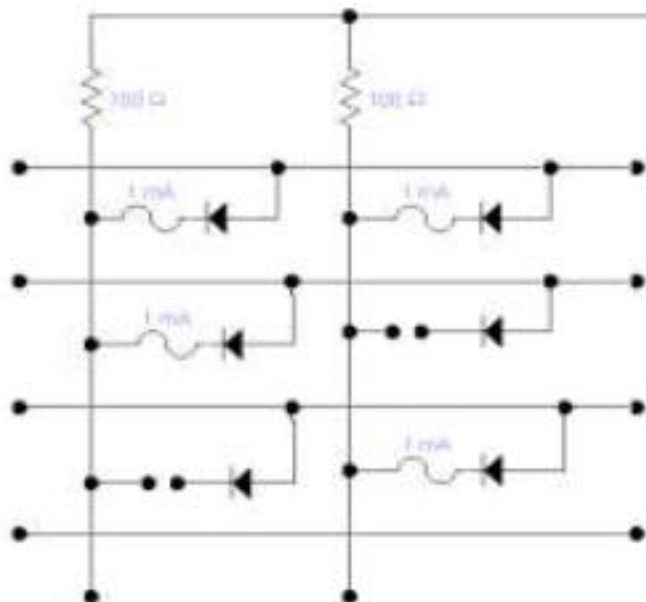


Рис. 5. Схема нагромаджувача ПЗП

дешифратора, та налагодити його роботу. Один із варіантів реалізації схеми дешифратора наведено на рис. 6.

Створення субблока дешифратора аналогічне до створення субблока комірки статичної пам'яті та наведене на рис. 7.

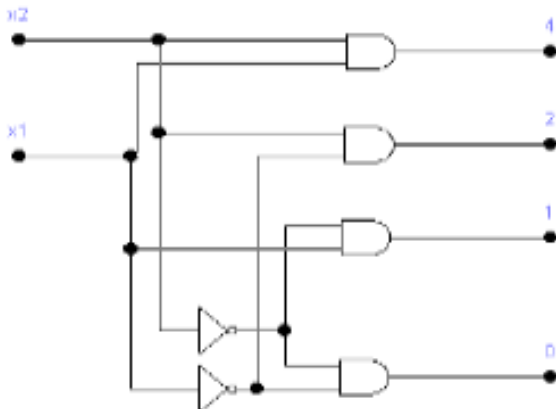


Рис. 6. Схема дешифратора з організацією 2×4

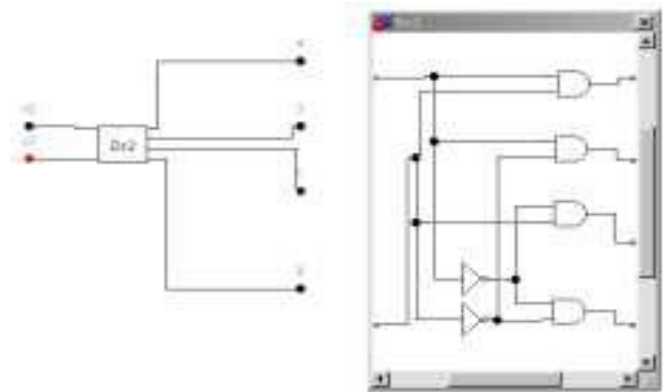


Рис. 7. Схема субблока дешифратора з організацією 2×4

3. Контрольні питання

1. Які бувають типи пам'яті?
2. Де в сучасних комп'ютерах використовується пам'ять статичного типу?
3. Чим відрізняється динамічна пам'ять від статичної?
4. Які типи динамічної пам'яті використовуються в сучасних комп'ютерах?
5. Що таке відеопам'ять та як вона пов'язана з характеристиками відображуваної на дисплеї інформації?

Тема 9. Основи мікропроцесорної техніки

Лабораторне заняття № 10 Дослідження засобів індикації МПС

Навчальна мета заняття:

1. Закріплення теоретичного матеріалу, набуття навиків програмування мікроконтролерів.
2. Набуття навиків моделювання функціонування мікроконтролерів на симуляторі EdSim5DI.

Кількість годин: 4 год.

Література:

1. Керівництво користувача EdSim51. URL: <https://www.edsim51.com/simInstructions.html#LDAndDI>
2. Круліковський Б.Б., Николайчук Я.М., Шатний С.В. Мікропроцесорні системи. Практикум. Навчальний посібник. Рівне : НУВГП, 2016. 191 с.
3. Проектування мікропроцесорних систем керування: навчальний посібник/ І.Р. Козбур, П.О. Марущак, В.Р. Медвідь, В.Б. Савків, В.П. Пісьціо. Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2022. 324с.
4. Методичні вказівки до лабораторних робіт з дисципліни «Мікропроцесорна техніка» Укл.: В.І. Рева. Запоріжжя: ЗНТУ, 2019. 114 с.
5. Плахтеев А.П., Бабешко Є.В, Ткаченко В.А., Здоровець Ю.В. Архітектури та розроблення систем Інтернету / Вебу Речей на основі вбудованих платформ.

Лабораторні роботи / За ред. В.С. Харченка. Міністерство освіти і науки України, Національний аерокосмічний університет, ХАІ, 2019. 147 с.

6. Мікроконтролери сімейства MCS-51 в задачах обробки інформації та керування. Методичні вказівки до виконання лабораторних робіт з дисципліни «Мікропроцесорні пристрої керування та обробки інформації». Укл.: Войтенко В.П., Хоменко М.А. Чернігів: ЧНТУ, 2014. 71 с.

Матеріально-технічне забезпечення: комп'ютерна мережа з інтегрованим середовищем EdSim51 Simulator.

Навчальні питання

- 1 Характеристика симулятора EdSim51 Simulator
- 2 Способи формування часових інтервалів
- 3 Програмування операцій часової затримки
- 4 Порядок виконання роботи
- 5 Контрольні питання

1 Характеристика симулятора EdSim51 Simulator

Симулятор призначений для моделювання роботи мікроконтролера I8051, розроблення та відлагодження програмного забезпечення для мікроконтролерних систем із периферійними пристроями.

Симулятор EdSim51DI TM Simulator працює під управлінням ОС Windows або Ubuntu в середовище Java, яке можливо завантажити за посиланням <http://www.java.com> та попередньо встановити на свій пристрій.

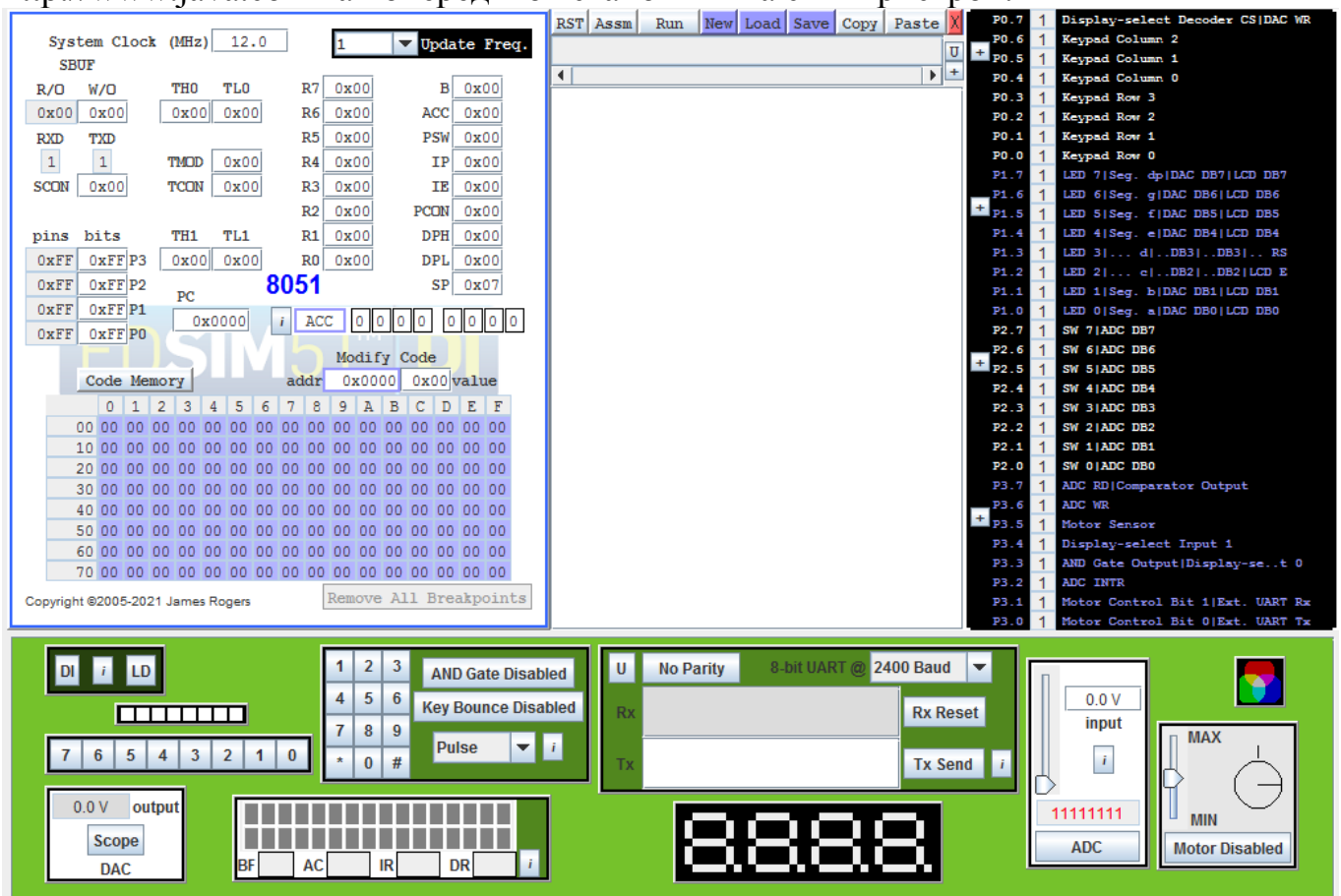
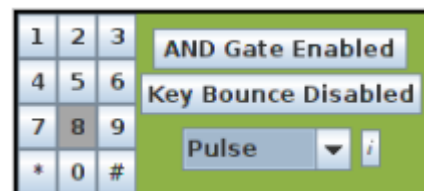


Рис. 1. Робоче вікно симулятора «edsim51»

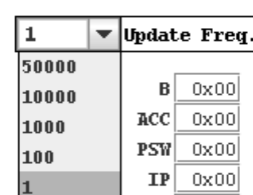
Отримати на EdSim51DI TM Simulator можливо на сайті розробника за посиланням <https://www.edsim51.com/index.html> безкоштовно. Після того, як ви завантажили файл, розпакуйте його у своїй папці. Для того, щоб запустити в емулятор потрібно двічі клацнути на файлі з ім'ям edsim51di.jar, після цього можна побачити екранну форму, вид який представлений на рис.1.

Симулятор «edsim51» призначений для написання і налагодження програм для мікропроцесорів сімейства Intel MCS-51. Програмування процесора здійснюється за допомогою побудови програми на мові асемблер.

Як видно, робоче вікно симулятора складається з чотирьох частин:



1. Область пам'яті мікропроцесора, в якій відображаються значення регістрів загального призначення, регістрів спеціальних функцій. Показаний область пам'яті програм і пам'яті даних. В цій області відображається елементи управління при налагодження програми.

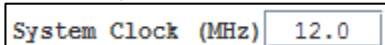


2. Вікно користувача. У цьому вікні відображається налагоджувальний код, який виконується в цій мікропроцесорному середовищі.

3. Вікно портів введення / виводу призначений для детального відображення стану значень портів P0-P3 в бінарному вигляді. Текстом підписані пристрої, які підключені до відповідних портів введення / виводу.

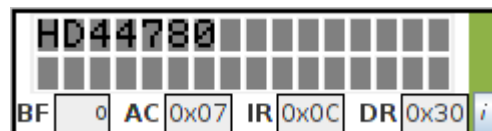
4. Панель зовнішніх пристроїв введення-виведення, підключених до мікропроцесора.

Зміна тактової частоти системи. Користувач може задавати значення тактової частоти в МГц. Допустимі значення знаходяться у діапазоні від 0,001 МГц до 999 МГц включно.



Зміна масштабу. Для моніторів високої роздільної здатності необхідно натиснути на кнопку +, яка знаходиться трохи нижче кнопки Exit.

Режими клавіатури. Користувач може вибрати один з трьох режимів роботи: Standard – одночасно можна натиснути кілька клавіш. Pulse – після відпускання кнопки миші клавіша повертається у вихідне положення. Radio – може бути зафіксована лише одна клавіша.



Оновлення вікна емулятора. EdSim51 дозволяє користувачеві або виконувати програму покроково (виконання однієї команди за один крок), або запустити програму у безперервне виконання. Паузи у проходженні дозволяють користувачу спостерігати за змінами в апаратній частині й регістрах. Користувач може обирати (показано на рисунку), як часто буде перезавантажуватися екран. Варіанти: оновлення екрану після кожного виконання команди (за замовчуванням), після 100, 1000, 10000 або 50000 виконаних команд.



Панель мікроконтролера. Надає користувачеві доступ до регістрів і пам'яті даних мікроконтролера. Поля, зафарбовані у білий колір, можуть редагуватися в

процесі роботи. В сірій – не можуть редагуватися. Наприклад, біти портів можуть бути змінені користувачем, але контакти порту керуються зовнішніми пристроями, тому не піддаються редагуванню. Крім того, можна змінювати значення програмного лічильника PC.

Бітове поле. В попередньому пункті відображені окремі біти акумулятора. Користувач може ввести будь-яку адресу або ім'я у полі синього кольору (замінивши ACC) і працювати з бітами за вказаною адресою. Крім того, якщо навести курсор на один з бітів, буде відображено опис біта. Бітове поле може бути використане також, щоб побачити біти за будь-якою адресою в оперативній пам'яті (від 0 до 7Fh), ввівши цю адресу в рядку синього кольору. Якщо ця область дозволяє бітову адресацію, поле є білим, і користувач може змінювати його значення.

Таким чином, якщо область не адресується за бітами, поле буде мати сіре забарвлення, як на цьому рисунку.

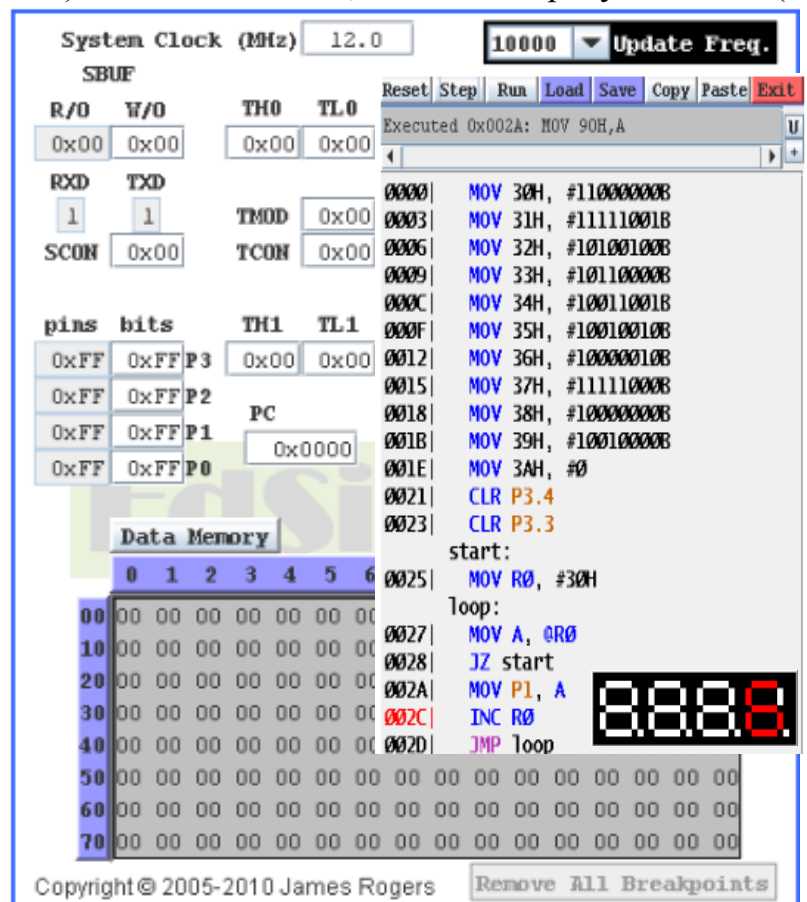
Пам'ять даних та програмна пам'ять. За замовчуванням відображається область пам'яті даних. Будь-яка адреса в оперативній пам'яті (від 00H до 7Fh) може бути змінена шляхом введення адреси в рядку синього кольору (addr), а далі – введенням потрібного значення в полі праворуч (value). Пам'ять програми теж може бути відображена та відредагована, як показано на рисунку нижче. Для перемикання між даними та програмою треба натиснути кнопку, позначену як «DataMemory/CodeMemory».

Пам'ять даних і кодів. За замовчуванням відображається пам'ять даних. Будь-яку адресу в ОЗУ (від 00H до 7FH) можна змінити, ввівши адресу в поле (з позначкою addr), а потім увівши потрібне значення в поле праворуч (з позначкою value).

Панель складання коду. Просту програма складання показано на панелі коду складання праворуч. Ця програма працює в безперервному циклі, відображаючи числа від 0 до 9 і назад до 0 на першому 7-сегментному дисплеї. Знімок його виконання показано праворуч.

Якщо фон текстової області коду складання білий, її можна редагувати. Програміст може написати код безпосередньо тут або завантажити програму з файлу за допомогою кнопки Завантажити (про це йдеться в наступному розділі).

Коли програма готова до



тестування, користувач може натиснути кнопку *Assm* , щоб виконати інструкції по черзі, або кнопку *Run*, щоб запустити програму безперервно. У будь-якому випадку програма спочатку буде зібрана. Якщо в коді виявлено помилку, у вікні повідомлення над кодом складання відображається повідомлення (з червоним фоном), а рядок із помилкою виділяється в коді червоним кольором.

Якщо код збирається без помилок, *Assm* замінюється на *Step* , фон текстової області змінюється на світло-сірий. Наразі код не можна редагувати. Якщо ви хочете повернутися до редагування свого коду, просто натисніть кнопку *RST*.

Симулятор *EdSim51* був розроблений таким чином, щоб усе було видно на екрані одночасно - внутрішні регістри, вихідний код, периферійні пристрої тощо. Однак під час написання самого коду панель вихідного коду може здатися трохи маленькою та обмеженою. Тому було додано нову функцію: натиснувши кнопку з позначкою *U* , користувач може розблокувати панель вихідного коду в головному вікні. Щоб заблокувати панель вихідного коду на місце в головному вікні, просто закрийте вікно вихідного коду, як показано.

Завантажити та зберегти. Користувач може писати код безпосередньо в текстове поле, коли воно знаходиться в режимі редагування, або наявну програму можна завантажити з файлу за допомогою кнопки *Load*. Так само код у текстовому полі можна зберегти у файл за допомогою кнопки *Seve*. Щоб зробити симулятор більш зручним для користувача, останній каталог, до якого було зроблено доступ, буде запам'ятовуватися. Таким чином, наступного разу, коли користувач відкриє діалогове вікно файлу, натиснувши ці кнопки, діалогове вікно автоматично відкриється в останньому відвіданому каталозі.

Копіювати і вставляти. Ви можете вибрати код у текстовій області збірки та скопіювати його до системного буфера обміну за допомогою кнопки *Copy* так само, як ви це робите у своєму пакеті обробки текстів. Потім це можна вставити в інше місце текстової області збірки за допомогою кнопки *Paste* (якщо текстову область можна редагувати – білий фон – якщо ні, натисніть *RST*). Або ви можете вставити виділений текст у іншу програму (наприклад, текстовий процесор). Подібним чином ви можете скопіювати текст з іншої програми та вставити його в текстову область збірки.

Кілька зауважень про асемблер. Асемблер із симулятором *EdSim51* не є повноцінним асемблером. Він не пов'язує кілька файлів і реалізовано лише деякі директиви, які ви могли очікувати. Нижче наведено список його функцій:

- Реалізовано всі інструкції 8051, за винятком інструкцій *MOVX* , оскільки симулятор не обробляє зовнішню пам'ять.
- *JMP rel* дорівнює або *SJMP rel* , або *AJMP rel* . *LJMP rel* має бути запрограмовано явно.
- Так само *CALL* дорівнює *ACALL* . *LCALL* має бути запрограмований явно.
- Реалізовано директиви *SET* і *EQU* .
- Впроваджено *ORG* .
- Реалізовано директиву *USING* (вказує, який банк-реєстратор використовується).

- ARn дорівнює адресі реєстру, як зазначено в USING (якщо банк реєстру не вказано до використання ARn, передбачається, що банк реєстру 0).
- Імена SFR та імена бітів SFR відповідають відповідній адресі.
- HIGH, за яким іде операнд у дужках, дорівнює старшому байту операнда.
- LOW, за яким іде операнд у дужках, дорівнює молодшому байту операнда.
- Після міток ставиться двокрапка.

– За замовчуванням для числових значень є десятковий. Шістнадцяткові значення можна ввести, додавши H після числа або поставивши перед ним 0x. Якщо використовується H, число не може починатися з літери (приклад: F5H має бути написано як 0F5H). Двійкові значення вводяться шляхом додавання B після числа (як показано на зображенні нижче).

- Асемблер не чутливий до регістру.

Налагодження. Незалежно від того, чи виконується код, чи виконується поетапний процес, коли код збирається без помилок, адреса кожної інструкції відображається ліворуч. Під час покрокового проходження коду інструкція, яку щойно було виконано, відображається в сірому полі вгорі разом із адресою інструкції (наведений приклад: Виконано 0x2A: MOV 90H, A). Наступна інструкція для виконання, її адреса підсвічується. (002CH).

Точку зупину можна встановити, двічі клацнувши адресу інструкції, як показано на зображеннях нижче. Встановлення точки зупину: під час покрокового проходження коду наведіть вказівник миші на адресу інструкції та двічі клацніть. Коли встановлено точку розриву, вертикальна смуга (|) праворуч від адреси замінюється зірочкою (*), як показано на зображенні праворуч. Видалення точки зупину: щоб видалити точку зупину, наведіть вказівник миші на адресу інструкції та двічі клацніть. * замінюється на |. Крім того, ви можете видалити всі контрольні точки одним клацанням: [Remove All Breakpoints](#).

```

0000| MOV 30H, #11000000B
0003| MOV 31H, #11111001B
0006| MOV 32H, #10100100B
0009| MOV 33H, #10110000B
000C| MOV 34H, #10011001B
000F| MOV 35H, #10010010B
0012| MOV 36H, #10000010B
0015| MOV 37H, #11111000B
0018| MOV 38H, #10000000B
001B| MOV 39H, #10010000B
001E* MOV 3AH, #0
0021| CLR P3.4
0023| CLR P3.3

start:
0025| MOV R0, #30H
loop:
0027| MOV A, @R0
0028| JZ start
002A| MOV P1, A
002C| INC R0
002D| JMP loop

```

Периферійні пристрої. Панель зовнішніх пристроїв введення-виведення, підключених до мікропроцесора, складається з наступних пристроїв:

- Аналого-цифровий перетворювач (АЦП), який перетворює аналогове напруга в цифровий вигляд (ADC);
- Дисплей, який складається з чотирьох семи сегментних індикаторів, підключених за схемою динамічної індикації (7 Segment LED Displays);
- Приймо-передавач послідовного порту (8-bit UART);
- Клавіатурний модуль (Keypad);
- Модуль світлодіодної індикації (LEDs);
- Імітатор обертання двигуна постійного струму (DC Motor);
- Модуль перемикачів (Switch bank);

– Цифро-аналоговий перетворювач (ЦАП), який формує аналогове напругу з цифрового коду. Вихід ЦАП підключений до імітатора осцилографа, який динамічно відображає форму і рівень сформованої напруги. (DAC output on scope).

Приклад 1. Записати в резидентну пам'ять даних за адресами 41 та 42 число 1C3Fh:

```
MOV R0,#41h ;завантаження в R0 покажчика даних
MOV @R0,#1Ch ;завантаження в пам'ять числа 1CH
INC R0 ;інкремент покажчика
MOV @R0,#3Fh ;записати в пам'ять число 3FH
```

Приклад 2. Завантажити в покажчик даних початкову адресу 7F69H масиву даних, розташованого у ВПД:

```
MOV DPTR, #7F69H ;завантаження початкового значення покажчика даних
```

Приклад 3. Завантажити керуюче слово в регістр управління таймером:

```
MOV TCON, #00000101B TCON 0x05
```

Приклад 4. Скинути всі прапорці користувача (область РПД з адресами 20H – 2FH):

```
MOV R0, #20H ; завдання початкової адреси області прапорців
MOV R1, #10H ; лічильник (довжина області прапорців)
LOOP: MOV @R0, #0 ; скидання одного байта (8 прапорців)
INC R0 ; перехід до наступного байта
DJNZ R1, LOOP ; цикл
```

Приклад 5. Вибрати нульовий регістровий банк:

```
ANL P2, #10111010B ; скидання бітів 0, 2, 6 порту 2
```

Приклад 6. Проінвертувати біти 7,6, 5 порту 0:

```
XRL P0, #11100000B ; виключне АБО порту 0 та константи
```

Приклад 7. Скласти програму відображення чисел від 0 до 9 і назад до 0 на першому 7-сегментному дисплеї, яка працює в безперервному циклі. Надати опис програми.

2 Способи формування часових інтервалів

При розробці програм керування об'єктами часто виникає необхідність виконувати певні дії з певними часовими параметрами. До таких задач відносяться формування часових інтервалів для керованих об'єктів, підрахунок кількості зовнішніх подій за певний інтервал часу, визначення часових параметрів зовнішніх сигналів, формування часових інтервалів для власних потреб тощо.

Для реалізації таких задач використовується два способи формування часових інтервалів: програмний та апаратно-програмний. При програмному способі формування часових інтервалів мікроконтролер виконує певну програму підрахунку кількості періодичних імпульсів власної тактової частоти протягом певного часу, тобто витрачає власні обчислювальні ресурси, не виконуючи при цьому іншої

Data Memory																addr		0x21	0x00	value
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F				
00	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				

ображає форму і рівень (ре).

пам'ять даних за адресами

P2.7	1	P0.7	0
P2.6	0	P0.6	0
P2.5	1	P0.5	0
P2.4	1	P0.4	1
P2.3	1	P0.3	1
P2.2	0	P0.2	1
P2.1	1	P0.1	1
P2.0	0	P0.0	1

показчика даних

пам'ять числа 1CH

Data Memory				
	0	1	2	3
00	42	00	00	00
10	00	00	00	00
20	06	03	00	00
30	C0	F9	A4	B0
40	00	1C	3F	00

DPH	0x7F
DPL	0x69

обчислювальної роботи. При апаратно-програмному способі для формування часових інтервалів використовуються вбудовані в МК програмно керовані апаратні засоби (таймери-лічильники T/COTaT/CI). Після запуску таймерів у роботу паралельно з роботою таймерів мікроконтролер може виконувати іншу обчислювальну роботу або керувати технологічним обладнанням.

При використанні обох способів формування часових інтервалів здійснюється наближено з певною похибкою: чим менша за обсягом програма, тим меншої точності формується часовий інтервал, і навпаки.

Програмна реалізація часових затримок може бути реалізована на основі програмного циклу. В один з регістрів завантажується число, яке потім в кожному проході циклу зменшується на 1. Так продовжується до тих пір, поки вміст використаного в якості лічильника регістра не стане дорівнювати 0. В такий момент виконання циклу завершується. Час затримки при цьому залежить від числа, що завантажено в регістр, а також часу виконання команд, що утворюють цикл. Оскільки необхідність у формуванні затримок може виникати в різних частинах програми і неоднократно, то відповідні програмні цикли доцільно оформляти у вигляді підпрограм, щоб можна було до них звернутися з будь-якого місця програми. Перша команда підпрограми позначається її назвою, що завершується двокрапкою, тобто міткою. Останньою командою підпрограми повинна бути RET, що забезпечує автоматичне повернення в основну програму.

У мікроконтролері MCS-51 тривалість циклу 1 мкс при тактовій частоті 12 МГц. Команда занесення числа в регістр виконується за 1 мкс, а команда DJNZ Rn, rel (декремент регістра і перехід, якщо не нуль) - за 2 мкс., а команда часу повернення з підпрограми RET - 2 мкс. Тому наведений фрагмент програми

```
DEL513: MOV R1, #n
C1:DJNZ R1, C1;
RET
```

виконується протягом часу T, який обчислюється за формулою: $T=12(3 + 2n)/f$, де f - частота синхронізації мікроконтролера; n - число, що записане в регістр R1.

Якщо $f = 12$ МГц і $n_{\max} = FFH$, то $T_{\max} = 3 + 2 \cdot 255 = 513$ мкс. Затримку більшої тривалості при тій же частоті синхронізації можна отримати вкладенням циклів.

3 Програмування операцій часової затримки

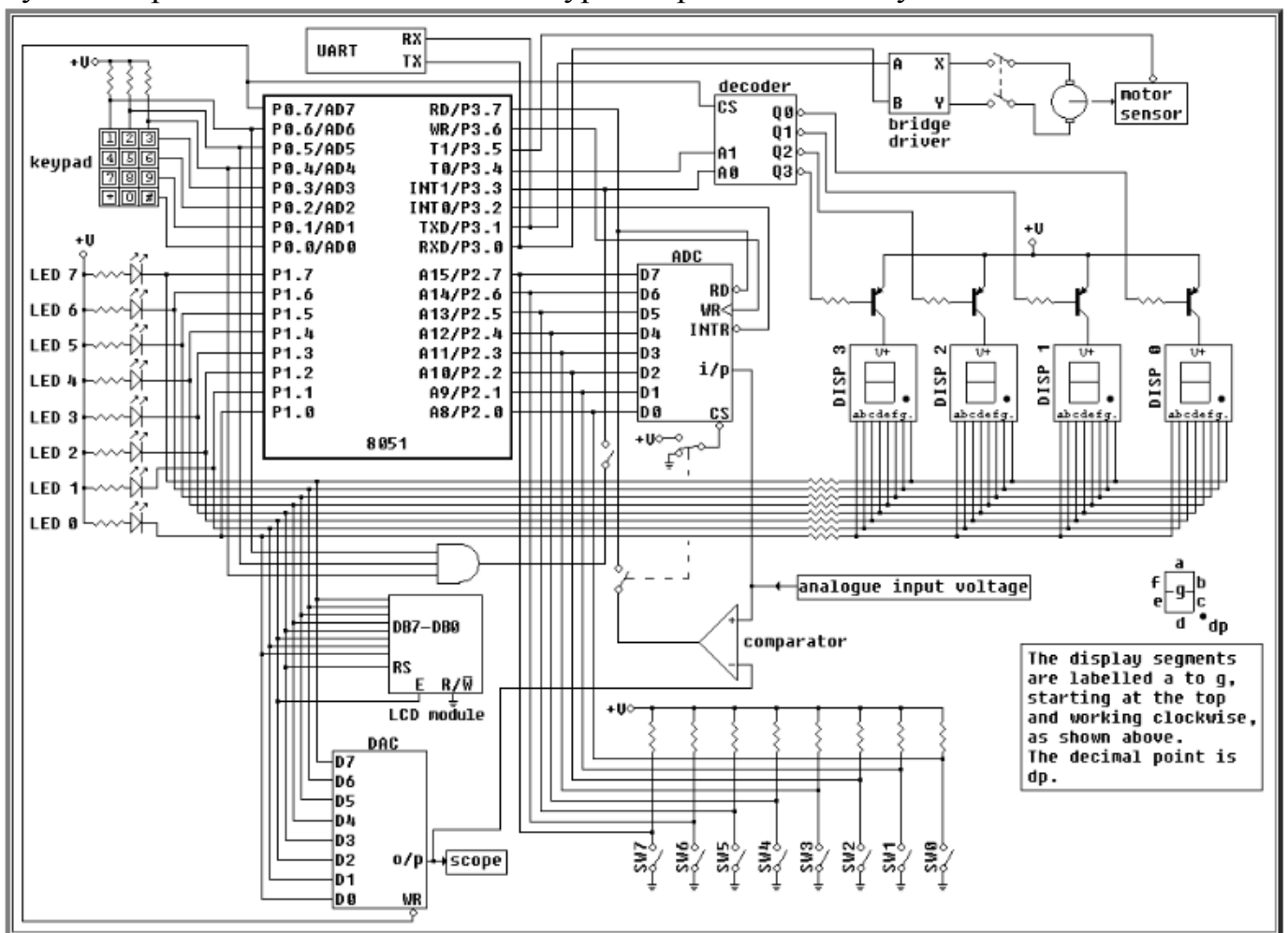
Нижче наведена схема, роботу якої моделює EDSim51DI, яка надає уявлення про периферійні зв'язки та взаємодію всіх пристроїв цієї схеми.

Як видно зі схеми, блок світлодіодів, входи ЦАП і лінії даних 7-сегментного дисплея мають спільний порт P1. Вибір того, який із чотирьох дисплеїв увімкнено, здійснюється за допомогою P3.3 і P3.4. Ці контакти порту застосовані до лінійного декодера 2-4, вихідні сигнали якого застосовані до бази транзисторів, які вмикають/вимикають дисплеї.

Декодер вмикається через логіку 1 на P0.7. Цей контакт також застосований до входу DAC WR, який має активний низький рівень. Таким чином, щоб записати дані в ЦАП, програміст вимикає дисплеї, що також призводить до ввімкнення ліній даних ЦАП. Потім дані можна записати в ЦАП. Все, що є на входах, коли лінія WR

переходить у низький рівень, зберігається (або записується) у внутрішній регістр ЦАП і залишається там після того, як лінія WR переходить у високий рівень. Аналоговий вихід для цих даних буде відображатися на осциллографі до наступного разу, коли WR буде прийнято на низький рівень, і нове значення буде розміщено у внутрішньому регістрі.

Клавіатура може бути реалізована за допомогою лінії зовнішнього переривання 1 (див. клавіатуру), але не одночасно мультиплексує 7-сегментні дисплеї. Це пояснюється тим, що лінія зовнішнього переривання 1 (P3.3) також використовується декодером дисплея (див. логічну схему вище). Таким чином, переривання клавіатури та мультиплексування 7-сегментного дисплея не можуть бути використані одночасно - клавіатура і ворота повинні бути вимкнені.



Відповідно до цього, нижче приведена програма видачі чисел на індикатори симулятора з затримкою

Del513:

org 00h ;директива розміщення програми з комірки 0

mov a,#11000000b;завантажити в акумулятор номер світного стану

mov p1,a ;вивести число 0 в лівий індикатор

call del ;виклик підпрограми затримки

clr p3.4 ;скидання біта 4 розряду 3 порту(управління вибором індикатора)

clr p3.3 ;скидання біта 3 розряду 3 порту (управління вибором індикатора)

mov a,#11111001b ;завантажити в акумулятор дозвіл світного стану

```

mov r1,a ;вивести число 1 в правий індикатор
del;підпрограма затримки
mov r1,#0ffh ;встановлення кількості циклів в регістру r1
c2:djnz r1,c2 ;декремент регістра r1 і перехід на c2, якщо не нуль
ret ;вихід з підпрограми
end ;директива завершення програми

```

4 Порядок виконання роботи

Завдання. Розробити алгоритм та програму формування сигналів керування індикаторами з часовими параметрами до заданого варіанту. Варіанти завдань для кожної бригади наведені в таблиці, де вказані тривалості імпульсів затримки $T_{\text{зат}}$ для індикаторів І0-І3, управління відбором яких реалізується двох розрядним кодом порту 3: P3.4,P33.

Варіант	1	2	3	4	5	6	7	8	9
$T_{\text{зат}}, \text{мкс}$	40	50	70	90	170	210	300	340	400
Перший	І0	І0	І0	ІІ	ІІ	ІІ	І2	І2	І2
Другий	ІІ	І2	І3	І0	І2	І3	І0	ІІ	І3

1. Розрахувати кількість циклів n для програми затримки.
2. Розпланувати ресурси мікроконтролера для операції затримки і вказати розміщення операндів та результату в шапці програми.
3. Записати мовою асемблера x51 програму управління станом індикаторів і підпрограму затримки. Для цього редагувати програму управління станами до потрібних індикаторів та завантаження розрахованого операнду кількості циклів вибрати для нього розміщення регістр або комірку пам'яті.
4. Для перевірки працездатності виконати розроблену програму на симуляторі. При отриманні хибного результату внести в програму корективи.
5. Розробити загальну блок-схему алгоритму затримки.
6. Вказати перелік команд, що призначені для виконання кожного блоку алгоритму і записати коментарі до алгоритму.
7. Створити скрін-шоти вікон симулятора за результатами виконання програми.

Надати висновки щодо налагодження програм на симуляторі.

5 Контрольні запитання

1. Основні характеристики однокристального МК сімейства MCS-51.
2. Організація резидентної пам'яті даних та пам'ять програм.
3. Система переривань однокристальних мікроЕОМ сімейства MCS-51.
4. Надати характеристику портів вводу-виводу.
5. Використання послідовного інтерфейсу.
6. Під'єднання до MCS-51 зовнішньої пам'яті даних та програм.
7. Вкажіть призначення бітів регістра PSW .
8. До якої групи команд відноситься команда LCALL?
9. Якою командою встановлюються окремі біти регістра PSW?
10. Якою командою скидаються окремі біти регістра PSW?
11. Яким чином визначається час виконання команд мікропроцесором?
12. В чому полягає принцип статичної індикації?

13. Вкажіть переваги та недоліки статичної індикації.
14. Які апаратні засоби симулятора виконують функцію статичної індикації?
15. Вкажіть призначення регістрів статичного індикатора.
16. Вкажіть принцип роботи семисегментного індикатора.
17. Вкажіть призначення та розташування сегментів індикатора.
18. Вкажіть призначення та принцип дії дешифратора семисегментного індикатора.
19. Якими командами виводиться інформація на статичний індикатор симулятора?
20. Яким чином здійснити керування світним станом статичного індикатора?

Тема № 9. Основи мікропроцесорної техніки

Лабораторне заняття №11. Програмування мікроконтролерів

Навчальна мета заняття:

1. Закріплення теоретичного матеріалу, набуття навиків програмування мікроконтролерів.
2. Набуття навиків моделювання функціонування мікроконтролерів на симуляторі EdSim5DI.

Кількість годин: 4 год.

Література:

1. Керівництво користувача EdSim51. URL: <https://www.edsim51.com/simInstructions.html#LDAndDI>
2. Круліковський Б.Б., Николайчук Я.М., Шатний С.В. Мікропроцесорні системи. Практикум. Навчальний посібник. Рівне : НУВГП, 2016. 191 с.
3. Проектування мікропроцесорних систем керування: навчальний посібник/ І.Р. Козбур, П.О. Марущак, В.Р. Медвідь, В.Б. Савків, В.П. Пісьціо. Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2022. 324с.
4. Методичні вказівки до лабораторних робіт з дисципліни «Мікропроцесорна техніка» Укл.: В.І. Рева. Запоріжжя: ЗНТУ, 2019. 114 с.
5. Плахтеев А.П., Бабешко Є.В, Ткаченко В.А., Здоровець Ю.В. Архітектури та розроблення систем Інтернету / Вебу Речей на основі вбудованих платформ. Лабораторні роботи / За ред. В.С. Харченка. Міністерство освіти і науки України, Національний аерокосмічний університет, ХАІ, 2019. 147 с.
6. Мікроконтролери сімейства MCS-51 в задачах обробки інформації та керування. Методичні вказівки до виконання лабораторних робіт з дисципліни «Мікропроцесорні пристрої керування та обробки інформації». Укл.: Войтенко В.П., Хоменко М.А. Чернігів: ЧНТУ, 2014. 71 с.

Матеріально-технічне забезпечення: комп'ютерна мережа з інтегрованим середовищем EdSim51 Simulator.

Навчальні питання

- 1 Алгоритм додавання багатобайтових чисел
- 2 Програмування операцій додавання операндів
- 3 Алгоритм віднімання багатобайтових чисел
- 4 Програмування операцій віднімання операндів

5 Порядок виконання роботи

6 Контрольні питання

1 Алгоритм додавання багатобайтових чисел

Усі команди арифметичних операцій призначені для роботи з додатними цілими числами байтового формату, хоча команди додавання і віднімання у випадку відсутності переповнення забезпечують одержання коректного результату при спеціальному способі кодування від'ємних чисел. При необхідності роботи з числами, що не можуть бути подані в форматі байта, необхідно розробляти відповідні підпрограми. Виконання операцій множення і ділення з числами, що мають довільний знак, можливе, якщо подати їх через знак і модуль та при використанні відповідних підпрограм власної розробки чи запозичені з бібліотек інших розробників.

Команда додавання *ADD* працює з даними у форматі байта, при цьому як приймач завжди використовується тільки акумулятор:

ADD A, #dat

ADD A, Rn

ADD A, @Ri

ADD A, src

Мнемоніка цієї команди відповідає слову *ADDition* (додавання). Для роботи з числами, що не можуть бути подані у вигляді одного байта, використовується команда додавання, яка враховує біт переносу, отриманий при додаванні попередньої пари байтів – *ADDC*:

ADDC A, #dat

ADDC A, Rn

ADDC A, @Ri

ADDC A, src

Літера *C* у позначенні команди вказує на використання біта переносу (*ADDition with Carrier*).

Додавання двійкових чисел здійснюється у відповідності з табл. 1, де для кожної комбінації вхідних сигналів доданку *X*, доданку *Y* та переносу з молодшого розряду *C_{in}* вказана відповідна комбінація вихідних сигналів суми *S* та переносу з поточного в наступний розряд *C_{out}*.

MCS-51 має команду однобайтового додавання акумулятора *A* з вказаним операндом з врахуванням переносу *ADDC*. *ADDC* використовує для сигналу *C_{in}* переносу в нульовий розряд поточне значення прапора переносу *CY* (або *C*) з регістру *PSW* мікроконтролера. Після виконання інструкції *ADDC* в *CY* записується *C_{out7}* - прапор переносу з старшого 7-го розряду, який може бути використаний в якості *C_{in0}* для наступного додавання старших байтів і фактично рахуватись *C_{in8}*. Таким чином, повторенням *ADDC* можна додавати числа з одного, двох і т.д. байтів. Для цього необхідно виконати *ADDC* для кожних 8 біт доданків починаючи з молодшого, не втрачаючи при цьому значень біту *CY* між сусідніми командами *ADDC*. Слід пам'ятати, що

Таблиця 1

<i>C_{in}</i>	<i>X</i>	<i>Y</i>	<i>C_{out}</i>	<i>S</i>
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

перед додаванням молодших байтів чисел, біт CY повинен бути встановленим в 0 за допомогою команди CLR C. Але є більш зручне рішення: команда ADD, в якій значення CY завжди береться рівним 0. Тому при додаванні молодших байтів багатобайтових чисел слід використовувати саме команду ADD.

Необхідно також пам'ятати наступне. При додаванні двох n -розрядних чисел сума може виявитись $n+1$ -розрядною. Тому, якщо при додаванні двох беззнакових чисел встановлюється прапор пере носу CY, це означає, що для збереження правильної суми необхідний ще один додатковий біт для зберігання цього прапора переносу, що дорівнює 1.

Таким чином, алгоритм add16 додавання двох цілих 16-розрядних чисел можна записати так:

- 1) Завантажити в акумулятор молодший байт доданку A.
- 2) Додати вміст акумулятора до молодшого байту доданку B.
- 3) Зберегти молодший байт суми доданків.
- 4) Завантажити старший байт доданку A в акумулятор.
- 5) Додати до вмісту акумулятора старший байт доданку B з врахуванням переносу із попередньої суми молодших байтів.
- 6) Зберегти старший байт суми.
- 7) В разі потреби зберегти $(n+1)$ -й розряд суми.

Для запису підпрограми необхідно вивчити призначення та порядок виконання команд ADD A, Rn; MOV A, Rn; ADDC A, Rn; JNC Out; RET, а також їх вплив на стан прапорів акумулятора.

Приклад виконання завдання. Визначаємо перелік необхідних ресурсів МК:

Вхід: R3, R2 - перший доданок A (в R3 - старший байт, в R2 - молодший байт), R5, R4 - другий доданок B (R5 - старший байт, R4 - молодший байт).

Вихід: R5, R4 - розміщення суми (R5 - старший байт, R4 - молодший байт). Регістр R6 - для можливого збереження $n+1$ -го біту суми. Прапори: CY - ознака переповнення при додаванні беззнакових чисел; OV - ознака переповнення при додавання чисел із знаком. Використані регістри: A. Вимагає вільних байт в стеку: 2 (для збереження адреси повернення з підпрограми).

Блок-схема алгоритму можна зобразити так (рис. 1), де блоки 1, 2, 3 призначені для додавання молодших байтів операндів та збереження суми в регістрі R4, блоки 4, 5, 6 виконують додавання старших байтів операндів та збереження їх суми в РЗП R5. Блоки 7 та 8 здійснюють збереження $n+1$ -го біту результату в разі потреби.

Підпрограму add16 з врахуванням необхідних коментарів можна записати таким чином, як показано у табл. 2.



Рис. 1. Блок-схема алгоритму add16 додавання

двохбайтових чисел

Підпрограма add16 додавання двох 16-ти розрядних чисел

Таблиця 2

Підпрограма add16 додавання двох 16-ти розрядних чисел		
;Вхід: R3R2 – перший доданок, R5R4 – другий доданок; Вихід: R5R4 – сума; Прапори: CY – прапор переповнення при додаванні чисел ; OV – прапор переповнення при додаванні чисел зі знаком ; Використані регістри: A. Вимагає вільних байт в стеку: 2		
Мітка	Команда	Коментарі
	MOV A, R2	;завантажити в акумулятор молодший байт першого доданку
	ADD A, R4	;додати до акумулятора молодший байт другого доданку
	MOV R4, A	;молодший байт суми помістити в регістр R4
	MOV A, R3	;завантажити в акумулятор старший байт першого доданку
	ADDC A, R5	;додати до акумулятора старший байт другого доданку
	MOV R5, A	;старший байт суми помістити в регістр R5
	JNC Out	;вийти з підпрограми, якщо прапор переносу не встановлено
	MOV R6, #01	;записати в R6 1, якщо розмірність результату більше 16
Out:	RET	;вихід на підпрограму, що викликала Add16

2 Програмування операцій додавання операндів

Приклад 1. Для МК51 складемо програму віднімання двох двобайтових чисел:

CLR C; ; очищення біта переносу

MOV A, R0 ; мол. байт першого числа в акумуляторі

SUBB A, R2 ; віднімання мол. байта другого числа

MOV R0, A ; запам'ятовування мол. байта різниці

MOV A, R1 ; ст. байт першого числа в акумуляторі

SUBB A, R3 ; віднімання ст. байта другого числа і позики

MOV R1, A ; запам'ятовування ст. байта різниці

Різниця між діями додавання і віднімання полягає ще і в тому, що при додаванні потрібно враховувати перенос, а при відніманні – позику. Але для збереження позики при відніманні використовується той же самий біт переносу. В обох прикладах використані реєстрова адресації операндів і результату.

За допомогою непрямої адресації можна написати більш компактну програму додавання чи віднімання чисел, що складаються з будь-якої кількості байтів (аби вистачило пам'яті).

Приклад 2. Для МК51 складемо програму додавання чисел. Число $A = 428_{10} = 1AC_{16}$ знаходиться в R5, R6. Число $B = 461_{10} = 1CD_{16}$ знаходиться в РПД у комірках пам'яті $12_{10} = C16$, $13_{10} = D16$ із запам'ятовуванням результату за адресою другого операнда:

CLR C

MOV R5, #ACh

MOV R6, #01

MOV 0Ch, #CDh

MOV 0Dh, #01

MOV A, R5

ADD A, 0Ch

```
MOV 0Ch, A
MOV A, R6
ADDC A, 0Dh
MOV 0Dh, A
```

Слід відзначити, що при виконанні команд пересилання кодів стан біта переносу не змінюється. Для віднімання потрібно здійснити аналогічні дії, але необхідно враховувати, що в системі команд i8051 відсутнім є звичайне віднімання, тому програма віднімання повинна починатися з очищення біта переносу.

Приклад 3. Написати програму додавання двох масивів чисел, кожен з яких містить 5 байтів. Результат розмістити за адресою першого масиву.

Нехай молодший байт першого операнда записаний в комірці з ім'ям A1, а всі наступні – у наступних комірках. Аналогічним чином байти другого операнда повинні бути записані в масив, перша комірка якого має ім'я A2. Припустимо, що ці два числа складаються з 5 байтів. Для визначення адрес пам'яті запишемо

A1 .EQU 50 ;перший масив починається з адреси 50 у ОЗП

A2 .EQU 55 ;другий масив починається з адреси 55 у ОЗП

Вважаючи, що запис операндів у ці комірки здійснений в іншій частині програми, напишемо програму додавання першого числа до другого із записом суми на місце першого числа:

MOV R0, #A1 ; запис адреси першого операнда в регістр

MOV R1, #A2 ; запис адреси другого операнда в регістр

MOV R3, #5 ; запис кількості байтів у регістр

CLR C ; очищення біта переносу

addm: MOV A, @R0 ; байт першого числа в акумуляторі

ADDC A, @R1 ; додавання байта другого операнда

MOV @R0, A ; запам'ятовування байта суми

INC R0 ; обчислення адреси байта першого числа

INC R1 ; обчислення адреси байта другого числа

DJNZ R3, addm ; підрахування кількості неопрацьованих байтів

Три команди на самому початку програми використовують безпосередню адресацію джерела. При цьому для двох з них транслятор здійснює підстановку фактичних значень адрес молодших байтів операндів. Команда очищення біта переносу потрібна тому, що в циклічній частині програми використовується команда додавання з урахуванням переносу. За рахунок використання непрямої адресації вдається обробити всі байти першого і другого операндів тими самими командами. А для того, щоб у наступному циклі звернутися до наступної комірки ОЗП, вміст регістрів R0 і R1 треба збільшувати на 1.

Рахунок кількості неопрацьованих байтів здійснюється останньою командою. При її виконанні число в регістрі R3 зменшується на 1, і якщо результат не дорівнює 0, то управління передається на початок циклу. При показаному раніше способі для додавання 5 пар байтів довелося б використовувати 15 команд, у наведеному прикладі їх тільки 10. Крім економії пам'яті програм, цей фрагмент більш універсальний. З іншого боку, тривалість роботи цієї програми більша, тому що для

її завершення потрібно виконати 34 команди замість 15. Таким чином, економія одного ресурсу, як правило, здійснюється за рахунок витрат інших.

Переходячи до двох інших арифметичних дій, слід зазначити, що множення чи ділення двійкового числа, що складається з декількох байтів, на цілий ступінь двійки здійснюється за допомогою зсуву всіх байтів цього числа вліво чи вправо. Оскільки вага двійкової цифри 1 зростає вдвічі при переході в сусідній лівий розряд, то для множення на 2 потрібний зсув уліво на 1 розряд, для множення на 4 — на 2 розряди і так далі. Для передачі бітів коду числа з одного байта в інший потрібно використовувати операцію циклічного зсуву вліво за участю біта переносу. Оскільки перед черговим зсувом крайнього байта значення біта переносу може бути довільним, треба встановлювати потрібне значення або після завершення зсувів коректувати вміст відповідної кількості розрядів молодшого або старшого байта. Множення на попередньо задану константу, у кодї якої міститься невелика кількість одиниць, також може здійснюватись послідовністю операцій зсувів і додавання. Однак у тому випадку, коли значення множника заздалегідь невідоме, потрібно використовувати операції множення.

Приклад 4. Програма занесення у внутрішній ОЗП константи 5D3F за адресами 50, 51 (32I6 та 33I6) та переміщення цієї константи в зовнішній ОЗП за адресами 31,32 (1F16, 2016):

```
ORG 0 ; директива вказує транслятору: розташувати коди команди,
; яка надходить за нею, в комірках пам'яті, починаючи з адреси 0
ALMP START ;
ORG 100h ; директива розміщення кодів команди, яка надходить
; за нею, в комірках пам'яті, починаючи з адреси 100h
START: ; мітка початку програми
MOV 32h, #5Dh
MOV 33h, #3Fh
MOV DPTR, #1Fh
MOV A, 32h
MOVX @ DPTR, A
INC DPTR
MOV A, #33h
MOVX @ DPTR, A
```

Приклад 5. Програма переміщення вмісту комірок пам'яті 3 ...12 зовнішнього ОЗП в комірки пам'яті з адресами 5 ... 14 внутрішнього ОЗП:

```
MOV A, #0A ; A ← кількість комірок
MOV R0, #03 ; R0 ← номер першої комірки інформації
MOV R1, #05; R1 ← номер першої комірки, в яку переміщується інф.
m106: MOV R3, A
MOVX A, @R0
MOV @R1, A
MOV A, R3
INC R0
INC R1
```

```

DEC A
JNZ m106
RET

```

Приклад 6. Зчитування вмісту портів P1, P2 і занесення зчитаних даних у внутрішній ОЗП за адресами 31, 32:

```

MOV 31, P1
MOV 32, P2

```

3 Алгоритм віднімання багатобайтових чисел

Усі команди арифметичних операцій призначені для роботи з додатними цілими числами байтового формату, хоча команди додавання і віднімання у випадку відсутності переповнення забезпечують одержання коректного результату при спеціальному способі кодування від'ємних чисел. При необхідності роботи з числами, що не можуть бути подані в форматі байта, необхідно розробляти відповідні підпрограми. Виконання операцій множення і ділення з числами, що мають довільний знак, можливе, якщо подати їх через знак і модуль та при використанні відповідних підпрограм власної розробки чи запозичені з бібліотек інших розробників.

Набір команд для віднімання набагато менший. Команда обчислення різниці – SUBB існує тільки у варіанті з відніманням вмісту біта переносу:

```

SUBB A, #dat
SUBB A, Rn
SUBB A, @Ri
SUBB A, src

```

Мнемоніка цієї команди відповідає словам SUBtraction with Borrow (тобто віднімання з урахуванням позики, тому що при відніманні утворюється позика, а не перенос). З цієї причини перед обчисленням різниці молодших байтів потрібно обов'язково очищати біт переносу, якщо немає впевненості в його вмісті. При обчисленні різниці старших байтів цього робити не потрібно.

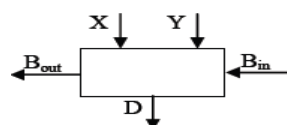


Рис. 1. Сигнали схеми віднімання

Таблиця істинності схеми двійкового віднімання

X	Y	B _{in}	D = X - Y	B _{out}
0	0	0	0	0
1	0	0	1	0
1	1	0	0	0
0	1	0	1	-1
0	0	-1	1	-1
1	0	-1	0	0
1	1	-1	1	-1
0	1	-1	0	-1

Повна однорозрядна схема віднімання для двійкової системи числення (рис. 1.) функціонує у відповідності до наступної таблиці істинності, де для кожної комбінації вхідних сигналів X, Y та B_{in} вказані значення вихідних сигналів D та B_o*. Щоб відрізнити позику від переносу перед одиницею проставлено знак (-).

Команда 8-ми бітового віднімання SUBB з врахуванням позики виконує віднімання від вмісту акумулятора X вміст вказаного операнда Y.

SUBB використовує для біту позики B нульовий розряд B_{in0} поточне значення прапора переносу CY з регістра PSW мікроконтролера. При завершенні виконання інструкції SUBB в CY записується B_{out7}, що є бітом позики з 7-го розряду. В свою чергу B_{out7}, збережений в CY, може стати позикою з нульового розряду B_{in0} при відніманні старших байтів чисел.

Таким чином, повторюючи SUBB, можна віднімати багатобайтові числа. Для цього треба виконати SUBB для кожного байта зменшуваного та від'ємника, не втративши при цьому значень CY між наступними командами SUBB. Перед відніманням молодших байтів операндів треба очистити біт CY командою CLR C, оскільки біт позики в нульовий розряд відсутній.

Для прикладу, алгоритм SUB16 віднімання двох цілих 16-розрядних чисел можна записати так:

- 1) Очистити прапор переносу CY.
- 2) Завантажити в акумулятор молодший байт зменшуваного.
- 3) Відняти від акумулятора молодший байт від'ємника.
- 4) Зберегти молодший байт різниці.
- 5) Завантажити старший байт зменшуваного в акумулятор.
- 6) Відняти від вмісту акумулятора старший байт від'ємника з врахуванням позики із молодшого байта.
- 7) Зберегти старший байт різниці.



Рис. 2. Блок-схема алгоритму SUB16 віднімання двохбайтових чисел

Таблиця 1

Підпрограма SUB16 віднімання двобайтових чисел		
;Вхід: R3 R2 – від'ємник, R5 R4 – зменшуване ;Вихід: R5 R4 – різниця. Прапори: CY – ознака позики(анти-переповнення) при відніманні беззнакових чисел ;OV – ознака переповнення при відніманні чисел із знаком ; Використані регістри: A; Вимагає вільних байт в стеку: 2		
Мітка	Команда	Коментарі
	CLR C	;Очистити прапор переносу для першого SUBB.
	MOV A, R4	;молодший байт зменшуваного записати в A
	SUBB A, R2	;Відняти молодший байт від'ємника.
	MOV R4, A	;Записати різницю молодших байтів на місце молодшого байта зменшуваного (R4).
	MOV A, R5	;Старший байт зменшуваного записати в A
	SUBB A, R3	;Відняти старший байт від'ємника і позику
	MOV R5, A	;Записати різницю старших байтів на місце старшого байта зменшуваного (R5).
	RET	;Повернення з підпрограми.

Блок-схема алгоритму SUB16 може мати такий вигляд (рис. 2). В алгоритмі SUB16 блок 1 призначений для очищення прапора позики перед відніманням молодших байтів операндів. Блоки 2, 3, 4 виконують віднімання молодших байтів операндів та збереження різниці в регістрі R4. Блоки 5, 6, 7 виконують віднімання старших байтів операндів та збереження їх різниці в R5. Для реалізації наведеного на рис. 3 алгоритму можливо використати наступну підпрограму SUB 16 (табл. 1).

Операцію віднімання зручно використовувати для порівняння двох чисел. Якщо після віднімання беззнакових чисел утворюється позика, то зменшуване менше від'ємника. Якщо позика не утворюється, то зменшуване більше від'ємника. Рівність має місце, якщо не утворюється позика і всі байти різниці дорівнюють 0.

Для виконання роботи необхідно вивчити призначення та порядок виконання мікроконтролером команд CLR C, MOV A, Rn та SUBB A, R2

4 Програмування операцій віднімання операндів

Приклад 1. Для МК51 складемо програму додавання двох двобайтових чисел $A = 1A25h$, $B = 2A36h$ із запам'ятовуванням результату за адресою першого операнда:

```
MOV R0, #25h ; запис молодшої частини першого числа в Rг0
MOV R1, #1Ah ; запис старшої частини першого числа в Rг1
MOV R2, #36h ; запис молодшої частини другого числа в Rг2
MOV R3, #2Ah ; запис старшої частини другого числа в Rг3
MOV A, R0 ; молодший байт першого числа в акумуляторі
ADD A, R2 ; додавання молодших байтів
MOV R0, A ; запам'ятовування молодшого байта суми
MOV A, R1 ; старший байт першого числа в акумуляторі
ADDC A, R3 ; додавання старшого байта другого числа та C
MOV R1, A ; запам'ятовування старшого байта суми
```

Приклад 2. Програма занесення у внутрішній ОЗП константи 5D3F за адресами 50, 51 (32I6 та 33I6) та переміщення цієї константи в зовнішній ОЗП за адресами 31,32 (1F16, 2016):

```
ORG 0 ; директива вказує транслятору: розташувати коди команди,
; яка надходить за нею, в комірках пам'яті, починаючи з адреси 0
ALMP START ;
ORG 100h ; директива розміщення кодів команди, яка надходить
; за нею, в комірках пам'яті, починаючи з адреси 100h
START: ; мітка початку програми
MOV 32h, #5Dh
MOV 33h, #3Fh
MOV DPTR, #1Fh
MOV A, 32h
MOVX @ DPTR, A
INC DPTR
MOV A, #33h
MOVX @ DPTR, A
```

Приклад 3. Програма переміщення вмісту комірок пам'яті 3 ...12 зовнішнього ОЗП в комірки пам'яті з адресами 5 ... 14 внутрішнього ОЗП:

```
MOV A, #0A ; A ← кількість комірок
MOV R0, #03 ; R0 ← номер першої комірки інформації
MOV R1, #05; R1 ← номер першої комірки, в яку переміщується інф.
m106: MOV R3, A
MOVX A, @R0
MOV @R1, A
MOV A, R3
INC R0
INC R1
DEC A
JNZ m106
```

RET

Приклад 4. Зчитування вмісту портів P1, P2 і занесення зчитаних даних у внутрішній ОЗП за адресами 31, 32:

MOV 31, P1

MOV 32, P2

5 Порядок виконання роботи

Розпланувати ресурси мікроконтролера для операції додавання/ віднімання і вказати розміщення операндів та результату в шапці програми.

8. Записати мовою асемблера x51 програму виконання операції додавання /віднімання. Для цього підпрограму (табл. 2) доповнити командами завантаження операндів у вибрані для їх розміщення регістри (комірки пам'яті).

9. Для перевірки працездатності виконати розроблену програму на симуляторі. При отриманні хибного результату внести в програму корективи.

10. Розробити загальну блок-схему алгоритму додавання/віднімання.

11. Вказати перелік команд, що призначені для виконання кожного блоку алгоритму і записати коментарі до алгоритму.

12. Створити скрін-шоти вікон симулятора за результатами виконання програми.

13. Надати висновки щодо відладки програм на симуляторі.

Варіанти завдань додавання

Варіант 1. Додати два числа. Число $A = 750_{10}$ знаходиться в R0, R1 число B – у R2, R3. Результат розташувати в резидентної пам'яті даних (РПД), починаючи з адреси 00h.

Варіант 2. Додати два числа. Число $A = 273_{10}$ знаходиться в R2, R3 число B – у РПД у комірках пам'яті: 31_{10} , 32_{10} . Результат розташувати в резидентної пам'яті даних (РПД), починаючи з адреси 02h.

Варіант 3. Додати два числа. Число $A = 275_{10}$ знаходиться в РПД у комірках пам'яті з адресами 33_{10} , 34_{10} число B – у РПД у комірках пам'яті з адресами 35_{10} , 36_{10} . Результат розташувати в резидентної пам'яті даних (РПД), починаючи з адреси 04h.

Варіант 4. Додати два числа. Число $A = 348_{10}$ знаходиться в РПД у комірках з адресами 35_{10} , 36_{10} число B – у РПД у комірках пам'яті з адресами 37_{10} , 38_{10} . Результат розташувати в резидентної пам'яті даних (РПД), починаючи з адреси 06h.

Варіант 5. Додати два числа. Число $A = 128_{10}$ знаходиться в РПД у комірках пам'яті з адресами 40_{10} , 41_{10} , число B – у резидентної пам'яті даних (РПД) у комірках пам'яті з адресами 05_{10} , 06_{10} Результат розташувати в РПД, починаючи з адреси 07h.

Варіант 6. Додати два числа. Число A знаходиться в РПД у комірках пам'яті з адресами 30h, 31h, число B – у зовнішньому ОЗП у комірках пам'яті з адресами 00h, 01h. Результат розташувати в РПД у комірках, починаючи з адреси 32h.

Варіант 7. Додати два числа. Число A знаходиться в РПД у комірках пам'яті з адресами 30h, 31h, число B – у РПД у комірках пам'яті з адресами 32h, 33h. Результат розташувати в РПД у комірках, починаючи з адреси 34h.

Додаткові варіанти завдань для програмування операції додавання

	Шістнадцяткове	Десяткове
1	$9F9F + 5FF = A59E$	$4863 + 1535 = 42398$
2	$75F6 + 1111 = 8707$	$3198 + 4369 = 34567$
3	$8D41 + 553F = E280$	$3161 + 4111 = 57984$
4	$5F22 + A131 = 10053$	$24354 + 41265 = 65619$
5	$7AA + 4450 = 10BFA$	$5114 + 7488 = 68602$
6	$3D77 + 3FF = 4176$	$15735 + 1023 = 16758$
7	$8811 + 2D0 = 8AE1$	$34833 + 720 = 35553$
8	$3388 + 78A = 3B12$	$13192 + 1930 = 15122$

Варіанти завдань віднімання

Варіант 1. Відняти два числа. Число А знаходиться у R4, R5 число В – у R6, R7. Результат розташувати у РПД, починаючи з адреси 21h.

Варіант 2. Відняти два числа. Число А = 373_{10} знаходиться в R0, R1 число В – у РПД у комітках пам'яті з адресами 42_{10} , 43_{10} . Результат розташувати в R4, R5.

Варіант 3. Відняти два числа. Число А = 375_{10} знаходиться в РПД у комітках пам'яті з адресами 21h, 22h, число В – у РПД у комітках пам'яті з адресами 40_{10} , 41_{10} . Результат розташувати в резидентної пам'яті даних (РПД) за адресою 10h.

Варіант 4. Відняти два числа. Число А = 448_{10} знаходиться в РПД у комітках пам'яті з адресами 42_{10} , 43_{10} , число В – у РПД у комітках пам'яті з адресами 2Ch, 2Dh. Результат розташувати в резидентної пам'яті даних (РПД) за адресою 0Ah.

Варіант 5. Відняти два числа. Число А = 148_{10} знаходиться в РПД у комітках пам'яті з адресами 44_{10} , 45_{10} , число В – у РПД у комітках пам'яті з адресами 40h, 41h. Результат розташувати в РПД в чарунках пам'яті з адресами 42h, 43h.

Варіант 6. Відняти два числа. Число А знаходиться в РПД у комітках пам'яті з адресами 40h, 41h, число В – у РПД у комітках пам'яті з адресами 42h, 43h. Результат розташувати в резидентної пам'яті даних (РПД), починаючи з адреси 00h.

Варіант 7. Відняти два числа. Число А знаходиться в резидентної пам'яті даних (РПД) у комітках пам'яті з адресами 00h, 01h, число В – у РПД у комітках пам'яті з адресами 50h, 51h. Результат розташувати в резидентної пам'яті даних (РПД), починаючи з адреси 02h.

Варіант 8. Відняти два числа. Число А знаходиться в РПД у комітках пам'яті з адресами 45h, 46h, число В – у R3, R4. Результат розташувати в резидентної пам'яті даних (РПД), починаючи з адреси 10h.

Додаткові варіанти завдань для програмування операції віднімання

Представлення чисел		
	Шістнадцяткове	Десяткове
1	$4444 - 5FF = 3E45$	$17476 - 1535 = 15941$
2	$A3FF - 1144 = 92BB$	$41983 - 4420 = 37563$
3	$7777 - A3F = 6D38$	$30583 - 2623 = 27960$
4	$DF2 - A20 = 3D2$	$3570 - 2592 = 978$

5	AAD5 – 555 = A580	43733 – 1365 = 42368
6	C077 – 7D7 = B8A0	49271 – 2007 = 47264
7	88EE – A88 = 7E66	35054 – 2696 = 32358
8	C350 – 7530 = 4E20	50000 – 30000 = 20000

5. Контрольні питання

1. Вкажіть формат і порядок виконання команди ADD додавання двох чисел мікроконтролером МК-51.
2. В яких випадках модифікується значення прапора CY PSW?
3. Чим відрізняється виконання команд ADD A, Rn та ADDC A, Rn?
4. Що треба забезпечити перед додаванням молодших байтів багатобайтових доданків?
5. Яку функцію виконує команда JNC Out в підпрограмі add16?
6. Вкажіть призначення біту CY PSW.
7. Яким чином можна скоротити підпрограму додавання константи та числа?
8. Яким чином можна скоротити підпрограму додавання числа та константи?
9. Яким чином можна зменшити обсяг пам'яті при виконанні операції додавання?
10. З якою метою виконується перевірка розрядності результату додавання?
11. Вкажіть формат і порядок виконання команди МК-51 для віднімання двох чисел.
12. В яких випадках модифікується значення прапора CY (C) PSW при виконанні операції віднімання?
13. Який стан регістра PSW треба забезпечити перед відніманням молодших байтів операндів?
14. Якою командою примусово встановлюються окремі біти регістра PSW?
15. Якою командою примусово скидаються окремі біти регістра PSW?
16. Яким чином можна скоротити підпрограму віднімання константи від числа?
17. Яким чином можна скоротити підпрограму віднімання числа від константи?
18. Яким чином можна зменшити обсяг пам'яті при виконанні операції віднімання?

Тема 10. Цифро-аналогові та аналого-цифрові перетворювачі

Лабораторне заняття №12. Дослідження цифро-аналогових та аналого-цифрових перетворювачів

Навчальна мета заняття:

1. Вивчення принципу дії цифро-аналогового перетворювача з ваговими двійково-зваженими опорам.
2. Практичне дослідження цифро-аналогового перетворення з використанням резистивної матриці.
3. Практичне дослідження принципу дії цифро-аналогових перетворювачів, побудованих на основі матриці з двома номіналами опорів R-2R.

4. Практичне дослідження роботи цифро-аналогового, аналого-цифрового перетворювачів, що представлені в бібліотеці програми Electronics Workbench.

Кількість годин: 4 год.

Література:

1. Онанченко Є.Л., Бражник І.Є. Методичні вказівки до виконання лабораторних робіт «Дослідження роботи ЦАП і АЦП» з дисципліни "Електронні системи". Суми : Сумський державний університет, 2012. 30 с.

2. Кравець В.О., Сокол Є.І., Рисований О.М. Комп'ютерна схемотехніка. Підручник. Харків: НТУ "ХПІ", 2007. 480 с.

3. Рева В.І. Методичні вказівки до лабораторних робіт з дисципліни «Мікропроцесорна техніка». Запоріжжя: ЗНТУ, 2019. 114 с.

Матеріально-технічне забезпечення: комп'ютерна мережа із системою модулювання електронних схем Electronics Workbench.

Навчальні питання

1 Основні характеристики ЦАП і АЦП

2 Дослідження цифро-аналогового перетворювача із двійково-зваженими опорам

3 Дослідження ЦАП на основі матриці R-2R

4 Дослідження роботи мікросхем АЦП

5 Контрольні питання

1. Основні характеристики ЦАП і АЦП

Характеристики ЦАП і АЦП визначаються видом характеристики перетворення, що встановлює відповідність між значеннями аналогової величини і цифрового коду, і відповідають за точність перетворення.

До *статичних параметрів* відносять:

– *розрядність n* – число розрядів цифрового коду, який формується на виході АЦП або подається на вхід ЦАП;

– *максимальна кількість кодових комбінацій* (рівнів квантування) на виході АЦП або вході ЦАП, що визначається числом розрядів цифрового коду і дорівнює: 2^n – для двійкових ЦАП (АЦП); 3^n – для трійкових ЦАП (АЦП);

– *максимальна напруга U_{\max}* (діапазон зміни вихідної напруги) – це максимальна вхідна напруга для АЦП і вихідна для ЦАП;

– *роздільна здатність* (абсолютна роздільна здатність) – найменше змінне значення вхідної величини, що розрізняється пристроєм і фіксується на виході; роздільна здатність для ЦАП - це мінімальне значення зміни вихідного сигналу, обумовлене зміною вхідного коду на одиницю у молодшому розряді; для АЦП – це мінімальна зміна величини аналогового сигналу, що викликає збільшення або зменшення вихідного коду на одиницю у молодшому розряді;

– *нелінійність δ_L* – це максимальне відхилення точки реальної характеристики перетворення від ідеальної;

– *диференціальна нелінійність* – це відхилення дійсного кроку квантування від його середнього значення;

– *абсолютна похибка* перетворення в кінцевій точці шкали – відхилення реальних максимальних значень вхідного для АЦП і вихідного для ЦАП аналогових

сигналів від значень, що відповідають кінцевій точці ідеальної характеристики перетворення;

– *напруга зсуву нуля* U_0 . Для АЦП це напруга, яку необхідно прикласти до його входу для отримання нульового вихідного коду. Для ЦАП - це напруга, наявна на його виході при подачі на вхід нульового коду.

Динамічні властивості ЦАП і АЦП характеризуються такими параметрами:

– *максимальна частота перетворення* – найбільша частота дискретизації, при якій задані параметри відповідають встановленим нормам;

– *час перетворення* – це інтервал часу від подачі цифрового коду на вхід ЦАП до появи вихідної напруги або інтервал часу від моменту зміни аналогового сигналу на виході АЦП до появи на його виході відповідного стійкого коду.

Процес аналого-цифрового перетворення показаний на рисунку 1 і полягає у послідовному виконанні таких дій:

– *процедура вибірки* – вибірка значень вхідної аналогової величини у деякий заданий момент часу, тобто відбувається дискретизація сигналу у часі (рис. 1а). Один із показників якості трансформації безперервного аналогового сигналу в цифровий сигнал – це *частота дискретизації*;

– *процес квантування* - округлення до деяких відомих величин (рівнів квантування) отриманих у дискретні моменти часу значень аналогової величини. У даному випадку на якість аналого-цифрового перетворення впливає кількість рівнів квантування, використовуваних для заміни безперервного аналогового сигналу на цифровий сигнал (рис. 1б);

– *кодування* - заміна знайдених окремих у часі значень вхідного сигналу на числові коди (рис. 1г).

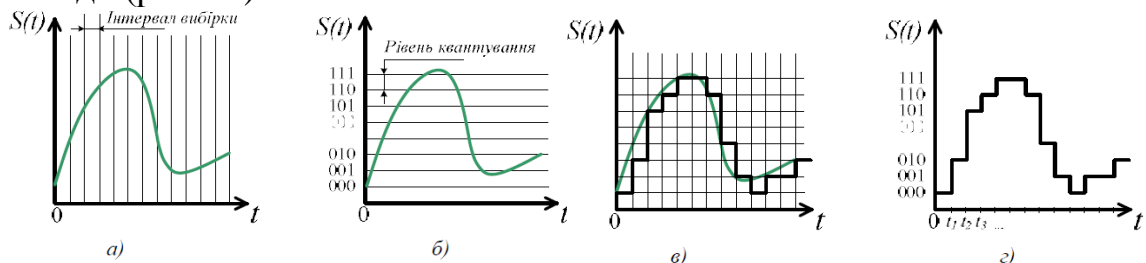


Рис.1. Перетворення аналогового сигналу у цифровий: а) процедура вибірки; б) процес квантування; в) вибірки і квантування; г) цифровий сигнал

Одна з основних проблем, з якою стикаються при перетворенні безперервного сигналу в дискретний, це проблема вибірки інтервалу дискретизації (частоти дискретизації). Відповідно до теорема Котельникова (теорема відліків, теорема Найквіста - Шенона), якщо аналоговий сигнал має обмежений по ширині спектр, то він може бути відновлений однозначно і без втрат за своїми дискретними відліками, узятими із частотою строго більше подвоєної верхньої (максимальної) частоти спектра.

2. Дослідження цифро-аналогового перетворювача із двійково-зваженими опорами

№	Опорна напруга	Вхідний код
---	----------------	-------------

	U_{on}, B	$a_3 a_2 a_1 a_0$
1	7	Таблиця 1
2	12	
3	13	
4	14	
5	16	
6	8	
7	9	
8	10	
9	11	
10	12	
11	14	
12	9	
13	18	
14	16	
15	8	
16	11	
17	10	
18	12	
19	13	
20	14	
21	15	
22	16	
23	17	
24	12	
25	14	

1. Зібрати схему ЦАП із двійково-зваженими опорамі (рис. 2). Схема складається з двох блоків: резистивної матриці (R_1 - R_4) і підсумувального підсилювача. Останній реалізований на операційному підсилювачі (ОП) з резистором R_{33} у зворотному зв'язку. U_{OP} – опорна напруга. Значення кожного розряду (вага) задається опором, зворотним зв'язку R_{33} і відповідними резисторами R_1 - R_4 .

Якщо всі перемикачі замкнуті на «землю», то напруга на виході ОП буде дорівнювати 0. При замиканні, наприклад, ключа K_4 в положення, що відповідає логічній одиниці, на вхід ОП через резистор R_4 подається напруга U_{OP} . Тоді коефіцієнт посилення вхідної напруги дорівнює $k=R_{33}/R_4$, а на виході маємо напругу $(U_{OP} R_{33})/R_4$.

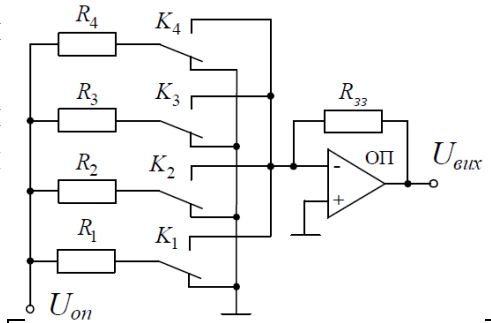


Рис. 2. ЦАП із двійково-зваженими опорамі

0000	Таблиця 2
0001	
...	
1111	

Отже, при подачі на вхід ЦАП цифрового коду $a_3 a_2 a_1 a_0$ вихідну напругу можна розрахувати за формулою:

$$U_{вих} = a_3 \cdot \underbrace{\frac{U_{on} \cdot R_{33}}{R_1}}_{\text{вага старшого розряду}} + a_2 \cdot \frac{U_{on} \cdot R_{33}}{R_2} + a_1 \cdot \frac{U_{on} \cdot R_{33}}{R_3} + a_0 \cdot \underbrace{\frac{U_{on} \cdot R_{33}}{R_4}}_{\text{вага молодшого розряду}},$$

2. Встановити напругу на джерелі U_{OP} згідно зі своїм варіантом відповідно з даними табл. 1.

3. Задати відповідні номінали опорів R_1 - R_4 : $R_1=R$, $R_2=2R$, $R_3=4R$, $R_4=8R$. Опір зворотного зв'язку дорівнює $R_{33}=R$ або $R_{33}=R/2$. Значення R може вибиратися будь-яке у розумних межах.

4. Комутуючи ключі K_1 , K_2 , K_3 і K_4 , тим самим набираючи послідовно вхідний код із табл. 1, виконати цифро-аналогове перетворення у покроковому режимі. Для кожного випадку зняти двома показання мультиметра(вольтметра) і заповнити колонки $U_{вих1}$ і $U_{вих2}$ у табл. 2.

На рис. 3 показаний момент подачі на ЦАП цифрового сигналу 1011 (K_1 - ключ старшого розряду). У даному випадку розрахункова вихідна напруга дорівнює

$$U_{вих} = 1 \cdot \frac{U_{on} \cdot R_{33}}{R_1} + 0 \cdot \frac{U_{on} \cdot R_{33}}{R_2} + 1 \cdot \frac{U_{on} \cdot R_{33}}{R_3} + 1 \cdot \frac{U_{on} \cdot R_{33}}{R_4}, \quad U_{вих} = 1 \cdot 12 + 0 \cdot 6 + 1 \cdot 3 + 1 \cdot 1,5 = 12 + 3 + 1,5 = 16,5 B$$

і, як бачимо, збігається з даними, які відображаються на мультиметрі (рис. 3).

5. Побудувати графіки залежності вихідної напруги від вхідного коду (характеристику перетворення) для обох випадків.

6. Визначити основні параметри ЦАП. Зробити висновки.

3. Дослідження ЦАП на основі матриці R-2R

1. Зібрати схему ЦАП на основі матриці R-2R з підсумовуванням напруги або струму з завдання викладача (рис. 4 або рис. 5). Перевага даних перетворювачів полягає в тому, що резистивна матриця містить тільки два номінали опорів на відміну від ЦАП із двійково-зваженими опорами, що має широкий діапазон номіналів.

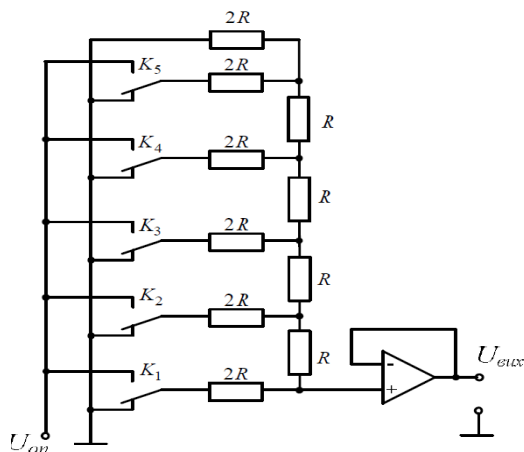


Рис. 4. ЦАП із матрицею R-2R з підсумовуванням напруги

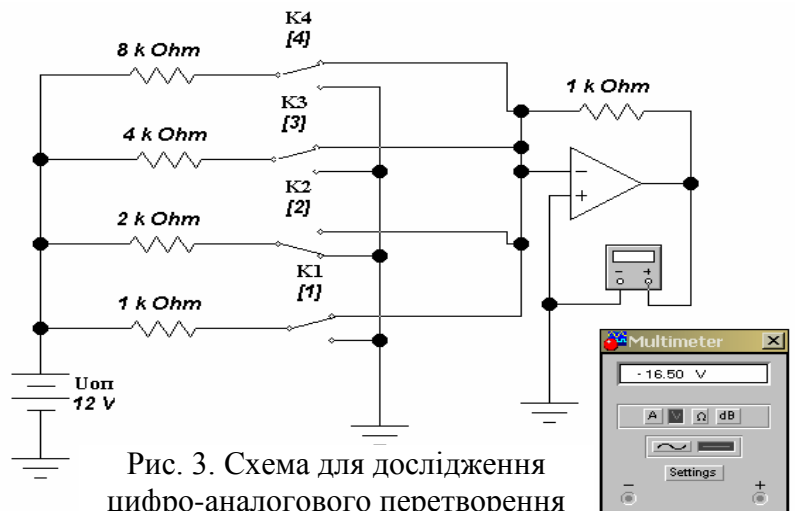


Рис. 3. Схема для дослідження цифро-аналогового перетворення

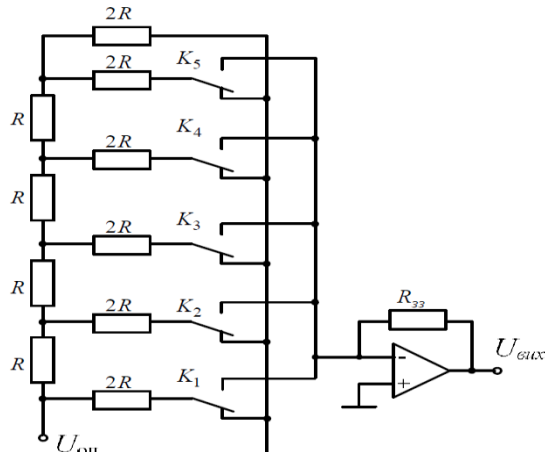


Рис. 5. ЦАП із матрицею R-2R із підсумовуванням струмів

2. Установити напругу джерела $U_{оп}$ згідно зі своїм варіантом (табл. 2, тут тільки вхідної код 5-ті розрядний). Задати відповідні номінали опорів. Значення опорів R та R_{33} вибрати самостійно.

3. Комутуючи ключі K_1-K_5 , послідовно подавати на вхід схеми цифровий код з урахуванням того, що ключ K_1 відповідає старшому розряду, а ключ K_5 - молодшому розряду цифрового коду. Кожного разу у колонку $U_{вих1}$ і $U_{вих2}$ табл. 2 записувати значення вихідної напруги, які відображаються на панелі мультиметра (вольтметра).

4. Побудувати графік залежності вихідної напруги від вхідного коду за отриманими результатами.

5. Зняти осцилограми. Для цього подати на вхід схеми цифровий код, який формується за допомогою генератора слів (Word Generator), а на виході операційного підсилювача підключити осцилограф (рис. 6).

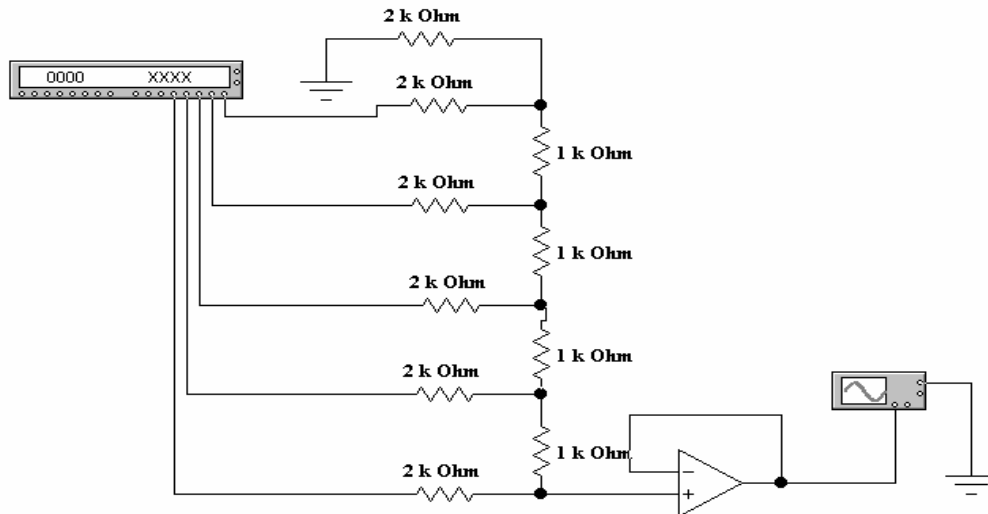


Рис. 6. Схема дослідження ЦАП із матрицею R-2R з підсумовуванням

4. Дослідження роботи мікросхем АЦП

1. Зібрати схему, наведену на рис. 7. Задати довільну частоту на генераторі (наприклад, 1 кГц), прямокутну форму коливань і амплітуду у межах 3 – 5 В. Установити опорну напругу, що дорівнює номеру варіанта.

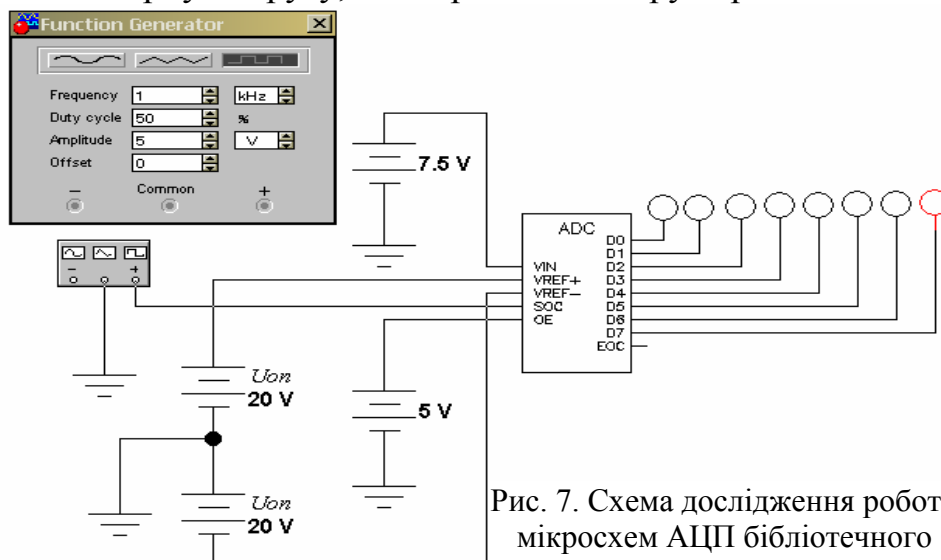


Рис. 7. Схема дослідження роботи мікросхем АЦП бібліотечного

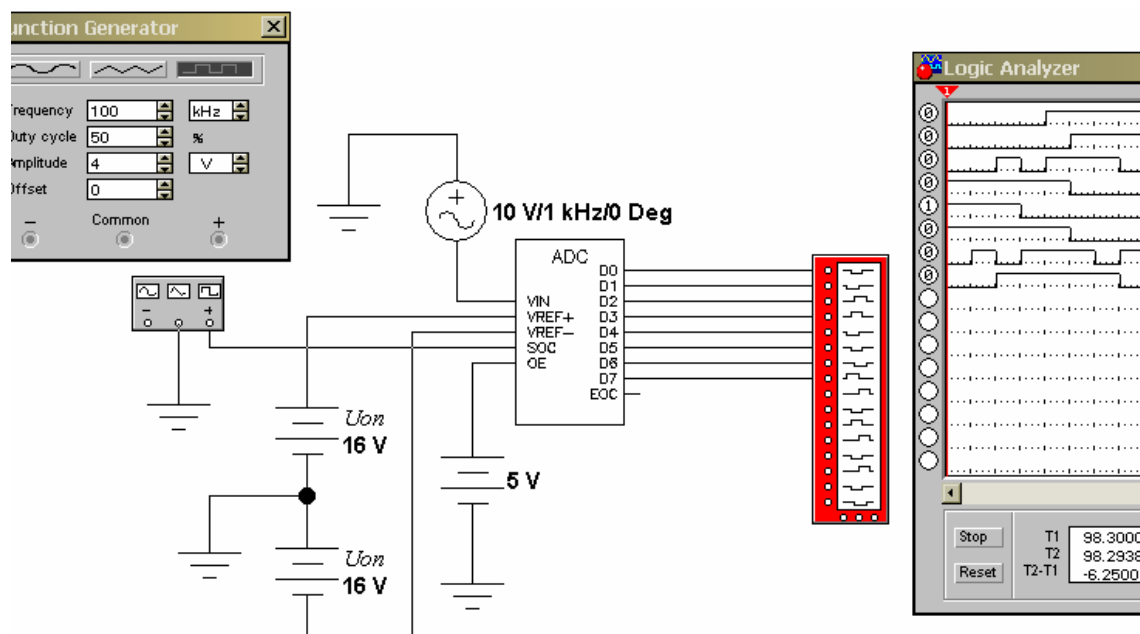
Призначення виводів мікросхеми АЦП: VIN - підключення джерела перетворюваного сигналу; VREF+, VREF- - підключення джерел опорної напруги; SOC - вхід синхронізації; OE - дозвіл на видачу вихідної двійкової комбінації на виходи D0...D7; EOC - сигнал готовності даних (наприклад, при видачі даних на комп'ютер).

Таблиця 3

Напруга на виході «VIN»	Значення індикаторів								D
	D7	D6	D5	D4	D3	D2	D1	D0	
$+U_{on}$									
$-U_{on}$									
$+U_{on}/2$									
$-U_{on}/2$									
$+U_{on}/10$									
$-U_{on}/10$									
0									

2. Змінюючи вхідну напругу на вході VIN, заповнити табл. 3. У останню колонку D таблиці занести розрахункове значення вихідного коду в десятковій системі числення.

3. Для схеми АЦП на рис. 8 побудувати часові діаграми. Установити опорну напругу, що дорівнює номеру варіанта. Зробити висновки.



5. Контрольні питання

1. Що розуміють під системою дискретного часу?
2. Яка головна вимога оптимальності методу дискретизації?
3. У чому полягає суть квантування?
4. У чому полягає сутність кодування інформації?
5. У чому сутність теореми Котельникова?
6. Назвіть та наведіть схеми найбільш поширених типів АЦП.
7. Назвіть та наведіть схеми найбільш поширених типів ЦАП.

3. Рекомендована література (основна, допоміжна), інформаційні ресурси в Інтернеті

Основна

1. Болюх В. Ф., Данько В. Г. Основи електроніки і мікропроцесорної техніки: Навч. посібник. Харків: НТУ «ХПІ», 2011. 257 с.
2. Зиков І.С., Межерицький С.Г., Подорожняк Л.О., Хавіна І.П. Програмування мікропроцесорів у зачищеному режимі: навч.-метод. Посібник. Харків : ТОВ «ДІСА ПЛЮС», 2018. 264 с.
3. Квітка С.О., Яковлев В.Ф., Нікітіна О.В. Електроніка та мікросхемотехніка: Навчальний посібник / За ред. проф. В.Ф. Яковлева. Київ: Аграрна освіта, 2010. 329 с.
4. Крилик Л.В., Селецька О.О. Матеріали електронної техніки: навчальний посібник. Вінниця: ВНТУ, 2017. 120 с
5. Круліковський Б.Б., Николайчук Я.М., Шатний С.В. Мікропроцесорні системи. Практикум. Навчальний посібник. Рівне : НУВГП, 2016. 191 с.
6. Методи перетворення сигналів. Навчально-методичний посібник до самостійної роботи і контролю знань студентів / Укл.: Ю.В. Головка. Запоріжжя: ЗДІА, 2011. 61 с.
7. Методичні вказівки до виконання лабораторних робіт «Дослідження роботи ЦАП і АЦП» з дисципліни "Електронні системи" / укладачі: Є. Л.

Онанченко, І. Є. Бражник. Суми : Сумський державний університет, 2012. 30 с.

8. Методичні вказівки до лабораторних робіт з дисципліни «Мікропроцесорна техніка» Укл.: В.І. Рева. Запоріжжя: ЗНТУ, 2019. 114 с.

9. Методичні вказівки до лабораторних робіт з курсу «Моделювання на ЕОМ» для студентів / Уклад. Харченко О. Л., Балєв В. М. Харків : НТУ «ХПІ», 2010. 68 с.

10. Чешко І.В. Вступ до спеціальності «Електроніка» : навчальний посібник. Суми : Сумський державний університет, 2017. 148 с.

Додаткова

1. Електроніка і мікропроцесорна техніка / Сенько В.І., Лисенко В.П., Юрченко О.М., Лукін В.Є., Руденський А.А. Київ: «Агроосвіта», 2015. 676 с.

2. Кирик В.В. Мікропроцесорна техніка: Навчальний посібник. Київ: ІВЦ «Видавництво «Політехніка», 2014. 183с.

3. Матвійків М.Д., Вус Б.С., Матвійків О.М. Елементи та компоненти електронних пристроїв: підруч. для студентів ВНЗ, які навчаються за напрямом «Радіоелектрон. апарати». Львів: Вид-во Львів. політехніки, 2015. 496 с.

4. Плахтєєв А.П., Бабешко Є.В., Ткаченко В.А., Здоровець Ю.В. Архітектури та розроблення систем Інтернету / Вебу Речей на основі вбудованих платформ. Лабораторні роботи / За ред. В.С. Харченка. Міністерство освіти і науки України, Національний аерокосмічний університет ХАІ, 2019. 147 с.

5. Теорія цифрових автоматів та формальних мов. Вступний курс : навч. посібник / Гавриленко С. Ю., Клименко А. М., Любченко Н.Ю. та ін. Харків : НТУ "ХПІ", 2011. 176 с.

6. Хіхловська І.В., Антонов О.С. Обчислювальна техніка та мікропроцесори. Підручник. Одеса: 2011. 440 с.

7. Цирульник С.М., Лисенко Г.Л. Проектування мікропроцесорних систем: навчальний посібник. Вінниця : ВНТУ, 2012. 201 с.

Інформаційні ресурси

1. KTechLab . URL: <https://sourceforge.net/projects/ktechlab/>

2. Курс електроніки. Лекції. URL: <http://vozom.ho.ua/index.html>

3. Михайлов С.Р. Основи мікропроцесорної техніки: Лабораторний практикум [Електронний ресурс] : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2019. 59 с.

4. Основи електроніки та мікропроцесорної техніки. URL: <http://repository.kpi.kharkov.ua/handle/KhPI-Press/18457>

5. Основи електроніки та схемотехніки. URL: <http://moodle.ipo.kpi.ua/moodle/mod/resource/view.php?id=12914>

6. Терещенко Т.О., Хоменко О.В. Мікропроцесорна техніка [Електронний ресурс]: конспект лекцій. Київ: КПІ ім. Ігоря Сікорського, 2017. 165с.