

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ» КОНСПЕКТ ЛЕКЦІЙ

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальністю 171 «Електроніка»,
спеціалізацією «Електронні компоненти і системи»*

Київ
КПІ ім. Ігоря Сікорського
2017

«Цифрові інформаційні системи»: конспект лекцій [Електронний ресурс]: навч. посіб. для студ. спеціальності 171 «Електроніка», спеціалізації «Електронні компоненти і системи» / КПІ ім. Ігоря Сікорського ; уклад.: К. С. Осипенко. – Електронні текстові данні (1 файл: 80,2 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2017. – 87 с.

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 3 від 23.11.2017 р.)
за поданням Вченої ради факультету електроніки (протокол № 10/2017 від 30.10.2017 р.)*

Електронне мережне навчальне видання

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

КОНСПЕКТ ЛЕКЦІЙ

Укладачі: *Осипенко Катерина Сергіївна*, канд. техн. наук.

Відповідальний редактор *Ямненко Ю. С.*, завідувач кафедри промислової електроніки,
д-р техн. наук, проф.

Рецензенти: *Попов А. О.*, доцент кафедри електронної інженерії, канд. техн. наук, доц.
Юрченко О. М., завідувач відділу транзисторних перетворювачів, д-р техн. наук, с.н.с.

При підготовці бакалаврів за спеціальністю 171 Електроніка, спеціалізацією «Електронні компоненти і системи» однією з важливих дисциплін, що є компонентом нормативної частини програми, є дисципліна «Цифрові інформаційні системи».

Метою розробки наочного конспекту лекцій є дати студентам ґрунтовні знання з питань теорії та практики побудови цифрових інформаційних систем. В результаті вивчення матеріалу конспекту лекцій студент повинен отримати знання з форм представлення інформації, Булевої алгебри, мови VHDL, комбінаційних пристроїв, аналізу та синтезу цифрових автоматів, лічильників, регістрів та ПЛІС.

Конспект лекцій містить презентації по 17 лекціям зі списком рекомендованої літератури.

ЗМІСТ

Вступ	4
Лекція №1. Форми представлення інформації. Моделі сигналів та цифрових пристроїв. Логічні елементи. Області застосування ЦІС	6
Лекція №2. Булева алгебра. Закони і тотожності алгебри логіки. Способи задання логічних функцій. Геометрична інтерпретація логічних функцій. Методи мінімізації логічних функцій: карти Карно (Вейча), метод Квайна, Квайна-Мак-Класкі	10
Лекція №3. Мова VHDL. Ідентифікатори в мові VHDL. Об'єкти, операції та оператори мови VHDL. Опис інтерфейсу пристрою	15
Лекція №4. Комутатори. Мультиплексори. Пристрої зсуву. Селектори-мультиплексори. Кодуючі та декодуєчі пристрої	22
Лекція №5. Суматори. Компаратори. Арифметично-логічні пристрої. Буферні елементи. Перехідні процеси в логічних схемах	27
Лекція № 6. Тригери. Тригерні схеми. Загальна характеристика тригерних схем. RS-, D-, JK-, T- та TV-тригери	31
Лекція №7. Синхронізація в цифрових схемах. Узгодження вхідних сигналів. Синхронізатори. Практичні проблеми проектування синхронних цифрових схем	35
Лекція №8. Загальна характеристика скінченних автоматів. Автомати Мілі і Мура. Закони функціонування автоматів. Способи опису роботи автоматів. Еквівалентні перетворення автоматів	39
Лекція №9. Основи аналізу цифрових автоматів. Аналіз автоматів з D-тригерами. Особливості аналізу скінченних автоматів з JK-тригерами	43
Лекція №10. Синтез скінченних автоматів	48
Лекція №11. Синтез асинхронних імпульсних автоматів. Особливості синтезу синхронних автоматів. Використання теореми Шенона при синтезі скінченних автоматів на основі JK-тригерів	53
Лекція №12. Колективна поведінка автоматів	56

Лекція №13. Теорія ігор	64
Лекція №14. Типи лічильників та особливості їх роботи. Асинхронні лічильники. Лічильники з довільним модулем рахунку. Синхронні лічильники. Скінченні автомати на основі лічильників	68
Лекція №15. Загальне поняття про регістри. Регістри пам'яті. Регістри зсуву. Забезпечення обміну інформацією у послідовному форматі. Регістрові лічильники імпульсів (розподілювачі). Лічильники Джонсона. Системи контролю цифрової апаратури	74
Лекція №16. Постійні запам'ятовуючі пристрої. Репрограмовані ПЗП	80
Основи побудови флеш пам'яті. Оперативні запам'ятовуючі пристрої	
Лекція №17. Основи побудови структур простих програмованих логічних матриць. Сучасні ПЛІС. Основні параметри ПЛІС	84

ВСТУП

Метою вивчення дисципліни «Цифрові інформаційні системи» є отримання студентами ґрунтовних знань з питань теорії та практики побудови цифрових інформаційних систем (ЦІС). Розглядається важливий клас ЦІС – автомати Мілі та Мура синхронного та асинхронного типів, синтез та аналіз їх структур, використання постійних запам'ятовуючих пристроїв (ПЗП), а також програмованих логічних інтегральних схем (ПЛІС) та побудова на їх основі цифрових пристроїв контролю, перетворення та відображення інформації.

Згідно з вимогами програми навчальної дисципліни студенти після засвоєння дисципліни мають продемонструвати такі результати навчання:

- знання з теорії представлення та мінімізації цифрових пристроїв, синтезу та аналізу цифрових автоматів за методикою Мілі та Мура синхронного та асинхронного типів із застосуванням сучасної інтегральної елементної бази;
- вміння використовувати теоретичні знання для розробки різноманітних структур ЦІС завадостійкого типу, проводити тестовий та функціональний контроль;
- навички застосувати набуті знання при виконанні курсових та випускної роботи, у практичній діяльності та наукових дослідженнях за фахом.

Поставлені цілі досягаються за рахунок широкого застосування в навчальному процесі обчислювальної техніки та сучасного програмного забезпечення, підвищення пізнавальної та творчої активності студентів.

Конспект лекцій містить інформацію по 16 лекціям. По кожній з лекцій наведено список рекомендованої літератури.

Лекція №1

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №1. Форми представлення інформації. Моделі сигналів та цифрових пристроїв. Логічні елементи. Области застосування ЦІС

Слайд 2

Зміст

- Форми представлення інформації
- Моделі цифрових пристроїв
- Аналітичне, графічне, програмне представлення цифрових пристроїв
- Моделі сигналів
- Логічні елементи
- Области застосування ЦІС

2

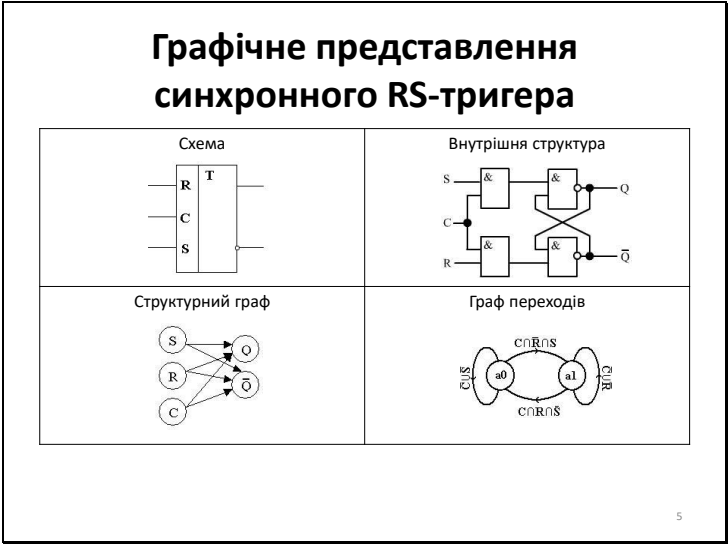
Слайд 3

Форми представлення інформації

Інформація – 1) сукупність відомостей про всілякі об'єкти, явища, процеси; 2) вибір одного варіанта (або декількох) із багатьох можливих і рівноправних.



3



Табличне представлення синхронного RS-тригера

Повна таблиця переходів <table><tr><th colspan="5">t</th><th>t+1</th></tr><tr><th>C</th><th>R</th><th>S</th><th>Q</th><th>Q</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>x</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>x</td></tr></table>	t					t+1	C	R	S	Q	Q	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	1	1	0	1	0	0	0	0	1	0	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	0	0	1	0	0	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1	0	x	1	1	1	1	x	Скорочена таблиця переходів <table><tr><th colspan="4">t</th><th>t+1</th></tr><tr><th>C</th><th>R</th><th>S</th><th>Q</th></tr><tr><td>0</td><td>x</td><td>x</td><td>Q(t)</td></tr><tr><td>1</td><td>0</td><td>0</td><td>Q(t)</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>x</td></tr></table>	t				t+1	C	R	S	Q	0	x	x	Q(t)	1	0	0	Q(t)	1	0	1	1	1	1	0	0	1	1	1	x
t					t+1																																																																																																																				
C	R	S	Q	Q																																																																																																																					
0	0	0	0	0																																																																																																																					
0	0	0	1	1																																																																																																																					
0	0	1	0	0																																																																																																																					
0	0	1	1	1																																																																																																																					
0	1	0	0	0																																																																																																																					
0	1	0	1	1																																																																																																																					
0	1	1	0	0																																																																																																																					
0	1	1	1	1																																																																																																																					
1	0	0	0	0																																																																																																																					
1	0	0	1	1																																																																																																																					
1	0	1	0	1																																																																																																																					
1	0	1	1	1																																																																																																																					
1	1	0	0	0																																																																																																																					
1	1	0	1	0																																																																																																																					
1	1	1	0	x																																																																																																																					
1	1	1	1	x																																																																																																																					
t				t+1																																																																																																																					
C	R	S	Q																																																																																																																						
0	x	x	Q(t)																																																																																																																						
1	0	0	Q(t)																																																																																																																						
1	0	1	1																																																																																																																						
1	1	0	0																																																																																																																						
1	1	1	x																																																																																																																						
Матриця переходів <table><tr><th>Q(t) - Q(t+1)</th><th>C</th><th>R</th><th>S</th></tr><tr><td>0 - 0</td><td>a</td><td>b</td><td>a & c</td></tr><tr><td>0 - 1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1 - 0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1 - 1</td><td>d</td><td>d & e</td><td>f</td></tr></table>	Q(t) - Q(t+1)	C	R	S	0 - 0	a	b	a & c	0 - 1	1	0	1	1 - 0	1	1	0	1 - 1	d	d & e	f	Карта Карно <table><tr><th>SQ</th><th>00</th><th>01</th><th>11</th><th>10</th></tr><tr><th>CR</th><td></td><td></td><td></td><td></td></tr><tr><td>00</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>01</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>11</td><td>0</td><td>0</td><td>x</td><td>x</td></tr><tr><td>10</td><td>0</td><td>1</td><td>1</td><td>1</td></tr></table>	SQ	00	01	11	10	CR					00	0	1	1	0	01	0	1	1	0	11	0	0	x	x	10	0	1	1	1																																																																						
Q(t) - Q(t+1)	C	R	S																																																																																																																						
0 - 0	a	b	a & c																																																																																																																						
0 - 1	1	0	1																																																																																																																						
1 - 0	1	1	0																																																																																																																						
1 - 1	d	d & e	f																																																																																																																						
SQ	00	01	11	10																																																																																																																					
CR																																																																																																																									
00	0	1	1	0																																																																																																																					
01	0	1	1	0																																																																																																																					
11	0	0	x	x																																																																																																																					
10	0	1	1	1																																																																																																																					

6

Аналітичне представлення синхронного RS-тригера

Рівняння функціонування тригера, отримане шляхом мінімізації карти Карно

$$Q(t+1) = \bar{C}Q \cup \bar{R}Q \cup CS$$

7

Алгоритмічне представлення синхронного RS-тригера

```
Мовою VHDL
architecture JK_tr_arc of JK_tr is
begin
  process(clk, reset, set) is
    variable q1 :std_logic;
    variable p :std_logic_vector(0 to 1);
  begin
    p :=set & reset;
    case p is
      when "00"=>q1 :='X';
      when "10"=>q1 :='0';
      when "01"=>q1 :='1';
      when "11"=>q1 := Q;
      when others => q1 :='X';
    end case;
    Q<=q1; Not_Q<=not q1;
  end process;
end JK_tr_arch;
```

8

Моделі сигналів

«U» – (Uninitialized) невизначене значення. Значення, яким ініціалізуються сигнали та змінні типу std_logic за замовчуванням. При нормальній роботі проекту, воно зазвичай не з'являється в моменти часу після першого призначення сигналу;

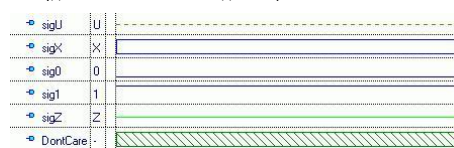
«X» – (Forcing Unknown) стан, що відповідає невизначеному значенню (0,4 - 2,4 В);

«0» – (Forcing 0) стан, що відповідає значенню логічного нуля (<0,4 В);

«1» – (Forcing 1) стан, що відповідає значенню логічної одиниці (>2,4 В);

«Z» – (High Impedance) стан високого імпедансу.

«-» – (Don't care) байдуже стан (значення). Використовується для завдання маски, оскільки при порівнянні з будь-яким значенням завжди повертає true.

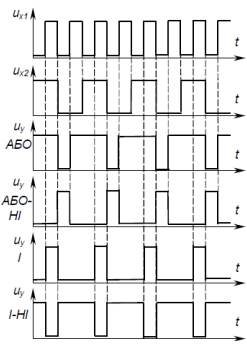


9

Логічні елементи					
Вхідні змінні		Функція у			
x_1	x_2	АБО	АБО-НІ	І	І-НІ
0	0	0	1	0	1
0	1	1	0	0	1
1	0	1	0	0	1
1	1	1	0	1	0
Математичний запис (формула)		$y = x_1 + x_2 = x_1 \vee x_2$	$y = \overline{x_1 + x_2}$	$y = x_1 \cdot x_2 = x_1 \wedge x_2$	$y = \overline{x_1 \cdot x_2}$
Назва функції		Логічне додавання (диз'юнкція) – функція АБО	Заперечення логічного додавання (стрілка Пірса) – функція АБО-НІ	Логічне множення (кон'юнкція) – функція І	Заперечення логічного множення (штрих Шеффера) – функція І-НІ
Графічне позначення елемента, що реалізує функцію					

10

Часові діаграми роботи двовходових логічних елементів



11

Області застосування ЦІС

- додатки, де необхідна велика кількість портів вводу-виводу
- цифрова обробка сигналу
- цифрова відеоаудіоапаратура
- високошвидкісна передача даних
- криптографія
- проектування і прототипування ASIC, як мостів (комутаторів) між системами з різною логікою і напругою живлення
- реалізація нейронів
- моделювання квантових обчислень

12

Список використаних джерел

1. Уилкинсон Б. Основы проектирования цифровых схем: Пер. с англ. – М.: Изд. дом «Вильямс», 2004. – 320 с.
2. Рябенский В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.

13

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

14

Лекція №2

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №2. Булева алгебра. Закони і тотожності алгебри логіки. Способи задання логічних функцій. Геометрична інтерпретація логічних функцій. Методи мінімізації логічних функцій: карти Карно (Вейча), метод Квайна, Квайна-Мак-Класкі

Слайд 2

Зміст

- Булева алгебра
- Закони і тотожності алгебри логіки
- Способи задання логічних функцій
- Теорема Шенона
- Розкладання Ріда
- Геометрична інтерпретація логічних функцій
- Методи мінімізації логічних функцій: карти Карно (Вейча), метод Квайна, Квайна-Мак-Класкі

2

Слайд 3

Булева алгебра

- Математичний апарат, що оперує з аргументами та функціями, які набувають тільки двох значень – “0” та “1” – називається *двійковою (булевою) алгеброю* або *алгеброю логіки*.
- Усі можливі логічні функції n змінних можна створити за допомогою трьох основних операцій:
 - а) логічне заперечення (інверсія, операція **НІ**); позначається рискою над відповідною функцією або аргументом – такий зв’язок між аргументом x та функцією y , при якому y істинна тоді і тільки тоді, коли значення x хибне, і навпаки;
 - б) логічне додавання (диз’юнкція, операція **АБО**), яке позначається символами $(\vee)$, $(+)$ – така функція, яка істинна тоді і тільки тоді, коли одночасно істинні всі логічні змінні;
 - в) логічне множення (кон’юнкція, операція **І**), яке позначається символами $(\wedge)$, $(\cdot)$, $(\&)$ – така функція, яка хибна тоді і тільки тоді, коли одночасно хибні всі додавані змінні. Для позначення еквівалентності логічних виразів використовується знак $(=)$.

3

Назва аксіоми чи закону	Вирази
Аксіоми (тотожності)	$0 \cdot x = 0; 1 + x = 1; 0 + x = x$ $x \cdot x = x; x + x = x$
Закони комутативності	$x_1 + x_2 = x_2 + x_1$ $x_1 \cdot x_2 = x_2 \cdot x_1$
Закони асоціативності	$x_1 + x_2 + x_3 = x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3 =$ $= (x_1 + x_3) + x_2$ $x_1 \cdot x_2 \cdot x_3 = x_1 \cdot (x_2 \cdot x_3) =$ $= x_2 \cdot (x_1 \cdot x_3) = x_3 \cdot (x_1 \cdot x_2)$
Закони дистрибутивності	$x_1 \cdot (x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3$ $x_1 + x_2 \cdot x_3 = (x_1 + x_2) \cdot (x_1 + x_3)$
Закони інверсії (теорема де Моргана, принцип подвійності)	$x_1 \cdot x_2 = \overline{x_1 + x_2}$ $x_1 + x_2 = \overline{x_1 \cdot x_2}$
Закони поглинання	$x_1 + x_1 \cdot x_2 = x_1$ $x_1 \cdot (x_1 + x_2) = x_1$

Способи задання логічних функцій
- Наведення функції у вигляді диз'юнкції кон'юнкцій називають <i>диз'юнктивною нормальною формою (ДНФ)</i>: $y = x_1x_2 + x_1x_3 + x_1x_2x_3$ - Запис функції у вигляді кон'юнкції диз'юнкцій – відповідно, <i>кон'юнктивною нормальною формою (КНФ)</i>: $y = (x_1 + x_2)(x_2 + x_3)(x_1 + x_2 + x_3)$ - Форми ДНФ та КНФ, в яких функція може бути записана єдиним чином називаються <i>досконалими диз'юнктивними (кон'юнктивними) нормальними формами (ДДНФ, ДКНФ)</i>. Вони характеризуються тим, що в ДДНФ кожна кон'юнкція, а в ДКНФ кожна диз'юнкція містять усі логічні змінні даної функції, з інверсіями або без них. - Оскільки кожна кон'юнкція функції, що наведена у ДДНФ, визначає її істинне значення, відповідає "1", то такі кон'юнкції називаються <i>конституєнтами одиниці (мінтермами)</i>. Аналогічно, диз'юнкції функції, що наведені у ДКНФ, називаються <i>конституєнтами нуля (макстермами)</i>. - Якщо замінити логічні змінні та їх заперечення одиницями та нулями, то кожна кон'юнкція буде представляти собою двійкове число.

Елементарні функції двох змінних

N	x_0	0	1	0	1	Назва функції	Позначення
	x_1	0	0	1	1		
0		0	0	0	0	Константа нуля	0
1		0	0	0	1	Кон'юнкція, І	$x_1 \cdot x_0$
2		0	0	1	0	Заборона по x_0	$x_1 \cdot \overline{x_0}$
3		0	0	1	1	Змінна x_1	x_1
4		0	1	0	0	Заборона по x_1	$\overline{x_1} \cdot x_0$
5		0	1	0	1	Змінна x_0	x_0
6		0	1	1	0	Виса, АБО, сума за mod 2	$x_1 \oplus x_0$
7		0	1	1	1	Диз'юнкція, АБО	$x_1 + x_0$
8		1	0	0	0	АБО-НІ, функція Парса	$\overline{x_0} \cdot \overline{x_1}$
9		1	0	0	1	Рівнозначність, еквівалентність	$x_1 \equiv x_0$
10		1	0	1	0	Заперечення x_0	$\overline{x_0}$
11		1	0	1	1	Імплікація по x_0	$x_1 + \overline{x_0}x_0 \rightarrow x_1$
12		1	1	0	0	Заперечення x_1	$\overline{x_1}$
13		1	1	0	1	Імплікація по x_1	$\overline{x_0}x_1 + x_0 \rightarrow x_0$
14		1	1	1	0	Функція Шеффера І-НІ	$\overline{x_1} \cdot \overline{x_0}$
15		1	1	1	1	Константа 1	1

6

Теорема Шенона

- Будь-яку функцію n змінних можна зобразити в формі

$$f(x_{n-1}, \dots, x_i, \dots, x_0) = \bar{x}_i \cdot f(x_{n-1}, \dots, 0, \dots, x_0) + x_i \cdot f(x_{n-1}, \dots, 1, \dots, x_0)$$

$$\bar{x}_i \cdot f(x_{n-1}, \dots, x_i, \dots, x_0) = \bar{x}_i \cdot f(x_{n-1}, \dots, 0, \dots, x_0);$$

$$x_i \cdot f(x_{n-1}, \dots, x_i, \dots, x_0) = x_i \cdot f(x_{n-1}, \dots, 1, \dots, x_0).$$

$$\bar{x}_i + f(x_{n-1}, \dots, x_i, \dots, x_0) = \bar{x}_i + f(x_{n-1}, \dots, 1, \dots, x_0);$$

$$x_i + f(x_{n-1}, \dots, x_i, \dots, x_0) = x_i + f(x_{n-1}, \dots, 0, \dots, x_0).$$

$$x + x \cdot y = x + 0 \cdot y = x;$$

$$x + \bar{x} \cdot y = x + 0 \cdot y = x + y.$$

7

Розкладання Ріда

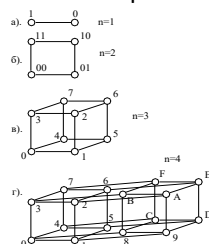
- Дає можливість будь-яку функцію n змінних зобразити у вигляді полінома n -го ступеня виду

$$f(x_{n-1}, \dots, x_i, \dots, x_0) = a_0 \oplus \sum_{i=0}^{n-1} b_1 x_i \oplus \sum_{i=0}^{n-1} b_2 x_i x_k \oplus \dots$$

8

Геометрична інтерпретація логічних функцій

- Карта Карно* – це компактна форма представлення таблиці істинності логічної функції із застосуванням для позначення (кодування) комбінацій змінних циклічного коду Грея
- N -вимірні куби,
або кубічні комплекси



9

Методи мінімізації логічних функцій

- Метою мінімізації є зменшення вартості технічної реалізації логічних функцій незалежно від використовуваних апаратних засобів.
- *Мінімізація логічної функції* – це заміна логічної функції, що представлена у вигляді логічної суми мінтермів або логічного добутку макстермів, іншою логічною функцією з мінімальною кількістю логічних змінних та операцій над ними.

10

Мінімізація за допомогою карт Карно (діаграм Вейча)

- клітини карти, координати яких відрізняються лише параметрами однієї змінної, називаються *сусідніми*;
- сусідні клітини, значення функцій в яких або тільки істинні, або тільки хибні, можуть об'єднуватися в групи по 2^m клітин, де m – ціле число ($m = 0, 1, 2, 3, \dots$);
- при переході до аналітичної форми запису логічної функції з карти Карно вона може записуватись незмінними координатами об'єднаних груп клітин;
- у випадку неповності визначеної функції невизначені клітини можуть бути довизначеними, виходячи з умови одержання більшої кількості об'єднаних клітин;
- одна клітина може об'єднуватись у декілька груп.

11

Мінімізація на основі використання кубічних комплексів

З кубічного комплексу $K(y)$ завжди можна виділити множину кубів $P(y)$ таких, що кожний член комплексу K_0 , тобто вершини куба (мінтерми), буде включеним по крайній мірі в один куб множини $P(y)$. Множина кубів $P(y)$ називається *покриттям комплексу $K(y)$, або покриттям логічної функції*.

Складність отриманої таким шляхом логічної функції прийнято характеризувати поняттям "*ціна покриття*" ($\mathcal{C}_n$). Ціна покриття дорівнює сумі цін всіх кубів, які складають дане покриття $P(y)$:

$$\mathcal{C}_n = \sum \mathcal{C}_k.$$

Ціна одного r -куба логічної функції n змінних визначається як різниця повного числа входних змінних і рангу відповідного куба, тобто дорівнює числу змінних у відповідній диз'юнкції:

$$\mathcal{C}_k = n - r.$$

Для логічної функції трьох змінних ціна 0-куба дорівнює трьом, а 2-куба – одиниці.

Задача мінімізації логічних функцій зводиться до пошуку покриття $P(y)$ кубічного комплексу $K(y)$, що має мінімальну ціну.

Покриття $P(y)$ комплексу $K(y)$, що має мінімальну ціну, називається *покриттям Квайна*, а відповідна цьому покриттю логічна функція – *мінімальною диз'юнктивною нормальною формою*.

12

Мінімізація логічних функцій за допомогою метода Квайна–Мак-Класкі

1. Знаходиться покриття $\Pi(y)$ заданої функції. Для цього формується кубічний комплекс логічної функції і в кожному i -му кубічному комплексі відмічають куби (імпліканти), які не утворили $(i + 1)$ -й кубічний комплекс. Відмічені імпліканти, які називаються *простими*, створюють покриття заданої логічної функції;
2. Будують таблицю покриттів матриці Квайна. Рядки вказаної таблиці відповідають простим імплікантам, а стовбці – 0-кубам (конституентам одиниці) логічної функції. На перетині i -го рядку та j -го стовбця ставиться мітка, якщо імпліканта i покриває конституенту j (імпліканта i покриває конституенту j в тому випадку, якщо вона відрізняється від неї незалежними аргументами);
3. Визначають покриття мінімальної вартості. Для цього:
 - виділяють ядро Квайна. Якщо 0-куб заданої логічної функції покривається лише однією простою імплікантою, то остання є суттєвою та входить в ядро Квайна і, відповідно, у покриття мінімальної вартості;
 - з таблиці викреслюються стовбці і рядки, які покриті імплікантами ядра Квайна. Якщо в отриманій після викреслювання таблиці містяться прості імпліканти, вони також включаються в ядро Квайна з послідовним викресленням відповідних рядків і стовбців;
 - стискається таблиця по стовбцях, для чого з неї викреслюються стовбці, які повністю включаються в будь-який з решти стовбців;
 - стискається таблиця по рядках, для чого з неї викреслюються рядки, які повністю включаються в будь-яку з решти рядків.
4. Послідовно стискаючи таблицю по рядках і стовбцях, отримують *циклічну таблицю*, імпліканти якої повинні входити в покриття логічної функції мінімальної вартості.

13

Список використаних джерел

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.

14

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

15

Лекція №3

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №3. Мова VHDL.
Ідентифікатори в мові VHDL. Об'єкти,
операції та оператори мови VHDL.
Опис інтерфейсу пристрою

Слайд 2

Зміст

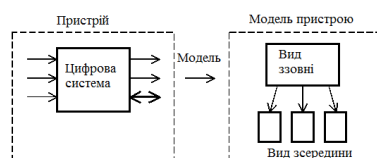
- Мова VHDL
- Операції мови VHDL
- Оператори мови VHDL
- Опис інтерфейсу пристрою
- Опис архітектури пристрою

2

Слайд 3

Мова VHDL

- Мова VHDL (Very high speed integrated circuits Hardware Description Language) служить для опису моделі цифрового пристрою (приладу, системи). Опис на мові VHDL визначає зовнішні зв'язки пристрої («вид ззовні» або інтерфейс) і один або декілька «видів зсередини».
- Вид зсередини визначає функціональні можливості пристрою або його структуру. Внутрішня будова об'єкта визначається архітектурою.



3

Ідентифікатори в мові VHDL

Abs	Access	after	alias	All
And	architecture	array	begin	Block
Body	Buffer	case	component	Configu-ration
Constant	disconnect	downto	else	Elsif
End	Entity	file	for	function
generate	Generic	guarded	if	In
inout	Is	label	library	linkage
loop	Map	mod	nand	New
next	Nor	not	null	Of
on	Open	or	others	Out
package	Port	procedure	process	Range
record	Register	rem	select	severity
signal	Subtype	then	to	Transport
type	Units	until	use	vriable
wait	When	while	with	Xor

4

Об'єкти мови VHDL

- Constant (константа) – може зберігати окреме значення певного типу. Це значення визначається об'єкту на початку моделювання і не може змінюватися в процесі моделювання.
- Variable (змінна) – об'єкт цього класу може зберігати окреме значення певного типу, проте, в процесі моделювання йому можуть присвоюватися різні значення. Для цього використовуються вирази привласнення (variable assignment statement).
- Signal (сигнал) – об'єкт даного класу має попереднє значення, має поточне значення і набір наступних значень.

5

Декларація констант

- constant RISE_TIME: TIME: = 10ns; - оголошується об'єкт RISE_TIME, який зберігає значення типу TIME, об'єкту на початку моделювання присвоюється величина 10 наносекунд
- constant BUS_WIDTH: INTEGER: = 8; - оголошується, що BUS_WIDTH (ширина шини) типу INTEGER (ціле) і їй присвоєно значення 8

6

Декларація змінних

- variable CTRL_STATUS: BIT_VECTOR (10 downto 0); - декларується змінна CTRL_STATUS як масив з 11 елементів, причому, кожен елемент типу BIT
- variable SUM: INTEGER range 0 to 100: = 10; - змінна SUM декларується як цілочисельна, що лежить в діапазоні від 0 до 100, на початку моделювання змінній присвоюється значення 10. Якщо змінній на початку моделювання не задано значення, то використовується значення за замовчуванням. Їм служить «крайнє ліве» значення в наборі значень даного типу
- variable FOUND, DONE: BOOLEAN; - початкове значення змінних FOUND і DONE буде взято за замовчуванням, тобто FALSE.

7

Декларація сигналів

- signal CLOCK: BIT; - декларується об'єкт CLOCK типу BIT, початкове значення при моделюванні буде взято за замовчуванням, тобто 0 (набір значень типу BIT: 0,1 і крайнє ліве значення 0)
- signal DATA_BUS: BIT_VECTOR (0 to 7);
- signal GATE_DELAY: TIME: = 10 ns;

8

Типи даних

- Scalar (скалярні), які в свою чергу поділяються на чотири типи:
 - Enumeration (перераховуваний тип),
 - Integer (цілочисельний тип),
 - Physical (фізичний тип),
 - Floating point (тип «з плаваючою комою»).
- Composite (компонитні) – вони складаються з елементів одного типу (масиви) або різного типу (записи),
- Access type (типи доступу) – забезпечують доступ до даного типу через покажчики,
- File types (файл) – забезпечує доступ до об'єктів, що містять послідовності значень даного типу.

9

Операції мови VHDL

1. Логічні операції
2. Операції відношення
3. Операції додавання/віднімання
4. Операції множення/ділення
5. Інші (операція знаходження абсолютного значення (abs) сумісна з будь-яким числовим типом операнда; операція піднесення до степеня (**) операндом зліва має цілочисельний або тип з плаваючою комою, а в якості правого операнда (ступінь) – тільки цілочисельний тип)

Пріоритет операцій зростає від категорії 1 до категорії 5. Операції однієї категорії мають однаковий пріоритет і виконуються в послідовності зліва направо. Дужки використовуються для зміни послідовності виконання.

10

Оператори мови VHDL

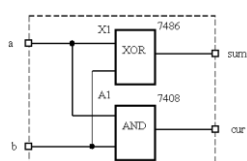
- Послідовні оператори (Sequential statements)
 - оператори присвоювання (Assignment statements)
 - умовні оператори if (if statements) та case (case statements)
 - оператори циклу (loop - next statements)
- Паралельні оператори (Concurrent statements)
 - оператор process
 - оператор паралельного виклику процедури
 - оператор конкретизації компонента
 - оператор генерації (generate)

11

Опис інтерфейсу пристрою

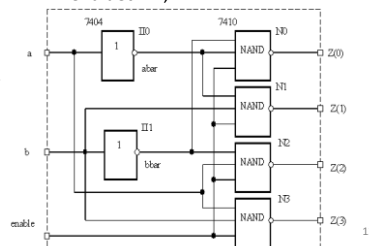
- Декларація інтерфейсу напівсуматора


```
entity half_adder is
port (a, b: in BIT; sum, cur: out BIT);
end half_adder;
```



- Декларація інтерфейсу дешифратора


```
entity dec2x4 is
port (a, b, enable: in BIT; z: out BIT_VECTOR (0 to 3));
end dec2x4;
```



12

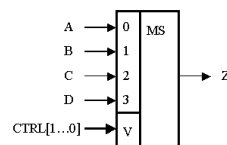
Поведінковий опис архітектури дешифратора

```
entity dec2x4 is
port (a, b, enable: in BIT; z: out BIT_VECTOR (0 to 3));
end dec2x4;
architecture dec_seq of dec2x4 is
begin
process (a, b, enable)
variable abar, bbar: BIT;
begin
abar: = not a; - вираз 1
bbar: = not b; - вираз 2
if enable = '1' then - вираз 3
z (3) <= not (a and b); - вираз 4
z (2) <= not (a and bbar); - вираз 5
z (1) <= not (abar and b); - вираз 6
z (0) <= not (abar and bbar); - вираз 7
else z <= "1111";
end if;
end process;
end dec_seq;
```

13

Поведінковий опис архітектури мультимплексера 4x1

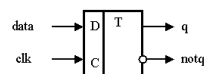
```
entity MUX is
port (A, B, C, D: in BIT; CTRL: in BIT_VECTOR (0 to 1);
Z: out BIT);
end MUX;
architecture MUX_BEHAVIOR of MUX is
begin
PMUX: process (A, B, C, D, CTRL)
variable TEMP: BIT;
begin
case CTRL is
when "00" => TEMP: = A;
when "01" => TEMP: = B;
when "10" => TEMP: = C;
when "11" => TEMP: = D;
end case;
Z <= TEMP;
end process PMUX;
end MUX_BEHAVIOR;
```



14

Поведінковий опис архітектури D-тригера

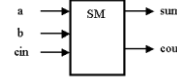
```
entity dff is
port (data, clk: in BIT
q, notq: out BIT);
end dff;
architecture behav of dff is
begin
process (clk)
begin
if (clk'event and clk = '1 ') then
q <= data;
notq <= not data;
end if;
end process;
end behav;
```



15

Потоковий опис архітектури однорозрядного суматора

```
entity full_adder is
port (a, b, cin: in BIT; sum, cout: out BIT);
end full_adder;
architecture full_ad_conc of full_adder is
begin
    sum <= (a xor b) xor cin after 15 ns;
    cout <= (a and b) or (b and cin) or (cin and a) after 10ns;
end full_ad_conc;
```



16

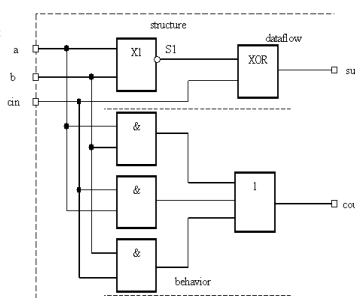
Структурний опис архітектури напівсуматора

```
entity half_adder is
port (a, b: in BIT; sum, cur: out BIT);
end half_adder;
architecture h_a_str of half_adder is
component a_7486
port (a_2: in BIT;
a_3: in BIT;
a_1: out BIT);
end component;
component a_7408
port (a_2: in BIT;
a_3: in BIT;
a_1: out BIT);
end component;
begin
    X1: a_7486 port map (a, b, sum);
    A1: a_7408 port map (a, b, cur)
end h_a_str;
```

17

Комбінований опис архітектури однорозрядного суматора

```
library altera;
use altera.maxplus2.all;
entity full_adder is
port (a, b, cin: in BIT; sum, cout: out BIT);
end full_adder;
architecture fa_mix of full_adder is
signal s1: BIT;
begin
    X1: a_7486 port map (a, b, s1); - вираз 1
    process (a, b, cin) - вираз 2
        variable t1, t2, t3: BIT;
        begin
            t1 := a and b;
            t2 := b and cin;
            t3 := a and cin;
            cout <= t1 or t2 or t3;
        end process;
        sum <= s1 xor cin; - вираз 3
    end fa_mix;
```



18

Список використаних джерел

1. Бибило П.Н. Основы языка VHDL: Учебное пособие. Изд. 5-е. – М.: Книжный дом «ЛИБРОКОМ», 2012. – 328 с.
2. Суворова Е., Шейнин Ю. Проектирование цифровых систем на VHDL. – СПб.: BHV, 2003. – С. 576.
3. Грушвицкий Р. И., Мурсаев А. Х., Угрюмов Е. П. Проектирование систем на микросхемах с программируемой структурой. – С-Пб.: БХВ-Петербург, 2006. – С. 736.
4. Бабак В. П., Корченко А. Г., Тимошенко Н. П., Филоненко С. Ф. VHDL. Справочное пособие по основам языка. – М.: Додэка - XXI, 2008. – С. 224

19

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

20

Лекція №4

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №4. Комутатори.
Мультиплексори. Пристрої зсуву.
Селектори-мультиплексори. Кодуючі
та декодуючі пристрої

Слайд 2

Зміст

- Комутатори
- Мультиплексори
- Пристрої зсуву
- Селектори-мультиплексори
- Кодуючі та декодуючі пристрої
- Дешифратори-демультиплексори

2

Слайд 3

Комутатори

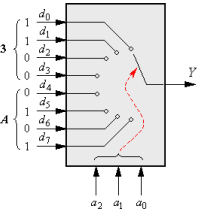
- *Комутатором* (у загальному випадку на n входів та m виходів – скорочено $n \times m$) називають цифровий пристрій, який забезпечує передачу сигналів з i -го вхідного на j -ий вихідний виводи.
- У цифровій схемотехніці найбільшого поширення набули такі комутуючі пристрої, як *мультиплексори, селектори-мультиплексори* (з організацією $n \times 1$) та *пристрої зсуву* ($n \times n$).

3

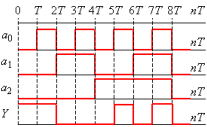
Умови і принцип перетворення двійкового коду з паралельного формату у послідовний

По одноканальній шині даних до пристрою підводиться двійкове слово $d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0 = A_{316}$, яке пристрій повинен зчитати і передати в послідовному форматі на вихід Y .

$$Y = \overline{a_2} \overline{a_1} \overline{a_0} d_0 + \overline{a_2} a_1 \overline{a_0} d_1 + \overline{a_2} a_1 a_0 d_2 + a_2 \overline{a_1} \overline{a_0} d_3 + a_2 \overline{a_1} a_0 d_4 + a_2 a_1 \overline{a_0} d_5 + a_2 a_1 a_0 d_6 + a_2 a_1 a_0 d_7$$



N	a ₂	a ₁	a ₀	Y	Код слова
0	0	0	0	d ₀	1
1	0	0	1	d ₁	1
2	0	1	0	d ₂	0
3	0	1	1	d ₃	0
4	1	0	0	d ₄	0
5	1	0	1	d ₅	1
6	1	1	0	d ₆	0
7	1	1	1	d ₇	1

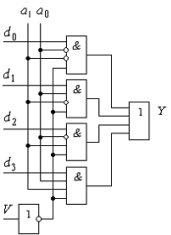
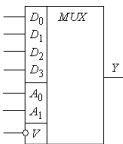


4

Мультиплексори

- Мультиплексор – це цифровий комбінаційний пристрій, який забезпечує перетворення інформації, що задана на входах $d_7 \dots d_0$ у паралельному форматі, в послідовний формат на виході Y .

$$Y = \overline{a_1} \cdot (\overline{a_1} \overline{a_0} d_0 + a_1 a_0 d_1 + a_1 \overline{a_0} d_2 + a_1 a_0 d_3)$$



5

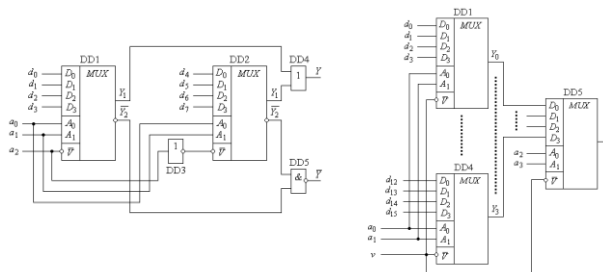
Перетворення логічних функцій для реалізації логічної функції на мультиплексорах

- у мінімізованій диз'юнктивній нормальній формі функції виділяються змінні, які мають найвищий ранг, тобто повторюються в найбільшій кількості диз'юнкцій. Наприклад, вибирається m змінних;
- виконується перетворення функції так, щоб забезпечити присутність виділених змінних в усіх диз'юнкціях. Для цього кожна диз'юнкція домножається на $1 = x + \overline{x}$;
- проводиться перегрупування змінних логічної функції таким чином, щоб виділені змінні були винесені за дужки у відповідних групах диз'юнкцій.

6

Схеми нарощування мультиплексорів

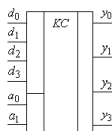
- Схема паралельного нарощування
- Схема пірамідального нарощування



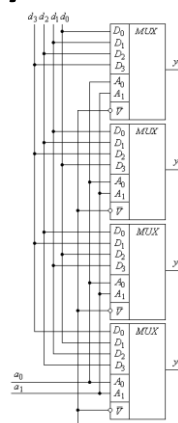
7

Пристрої зсуву

- Пристрої зсуву – це комбінаційні пристрої, призначені для забезпечення передачі вхідного коду на вихід зі зсувом на декілька розрядів у залежності від керуючого коду.



$$\begin{aligned}
 y_0 &= d_0 \bar{a}_1 \bar{a}_0 + d_1 \bar{a}_1 a_0 + d_2 \bar{a}_1 a_0 + d_3 \bar{a}_1 a_0; \\
 y_1 &= d_1 \bar{a}_1 \bar{a}_0 + d_2 \bar{a}_1 a_0 + d_3 \bar{a}_1 a_0 + d_0 \bar{a}_1 a_0; \\
 y_2 &= d_2 \bar{a}_1 \bar{a}_0 + d_3 \bar{a}_1 a_0 + d_0 \bar{a}_1 a_0 + d_1 \bar{a}_1 a_0; \\
 y_3 &= d_3 \bar{a}_1 \bar{a}_0 + d_0 \bar{a}_1 a_0 + d_1 \bar{a}_1 a_0 + d_2 \bar{a}_1 a_0.
 \end{aligned}$$



8

Селектори-мультиплексори

- Селектори-мультиплексори – прилади, які дають можливість забезпечувати не тільки побітну цифрову передачу інформації, а й вибирати цифрові потоки інформації з будь-якого входу на вихід.
- Робота полягає в тому, що двійковий цифровий код може подаватись на вхід Y , а зніматись побітово з виходів d_0, d_1, d_2, d_3 , забезпечуючи тим самим перетворення послідовного формату в паралельний.

9

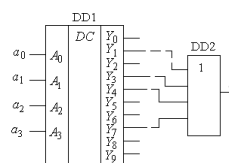
Кодуючі та декодуючі пристрої

- Шифрування полягає у перетворенні m -розрядного коду з k_m байдужих наборів вхідних змінних в однозначно відповідний йому n -розрядний код з меншим числом розрядів ($n < m$) і байдужих наборів. Під байдужими наборами в будь-якому коді маються на увазі ті набори, які не використовуються при перетворенні.
- Пристрої, які виконують вказані операції, називаються відповідно *шифраторами* та *дешифраторами* (кодерами та декодерами).

10

Дешифратори-демультиплексори

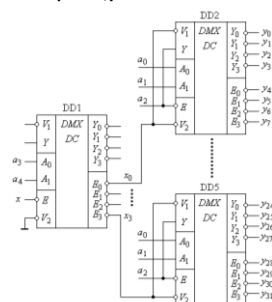
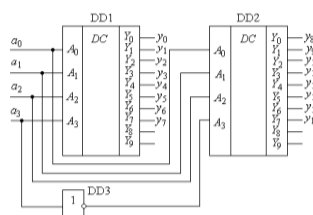
- *Демультиплексори* – це цифрові комбінаційні пристрої, в яких сигнали з одного інформаційного входу x розподіляються на 2^m виходів y_i , які комутуються m адресними входами, тобто фактично вони є генераторами мінтермів.



11

Схеми нарощування демультиплексорів

- Схема паралельного нарощування
- Схема пірамідального нарощування



12

Список використаних джерел

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д.
Цифрова схемотехніка: Навч. посібник. –
Львів: Новий світ – 2000, 2009, – 736 с.

13

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

14

Лекція №5

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №5. Суматори. Компаратори.
Арифметично-логічні пристрої.
Буферні елементи. Перехідні процеси
в логічних схемах

Слайд 2

Зміст

- Суматори
- Компаратори
- Арифметично-логічні пристрої
- Буферні елементи
- Перехідні процеси в логічних схемах

2

Слайд 3

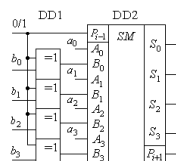
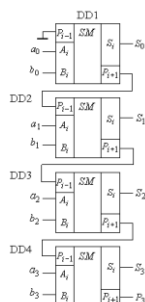
Принцип роботи повного суматора

- Суматор – це пристрій для знаходження суми багаторозрядних слів.

N	a_i	b_i	p_{i-1}	S_i	P_{i+1}
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

$$S_i = \vee 1, 2, 4, 7 = p_{i-1}(\bar{a}_i \cdot \bar{b}_i + a_i \cdot b_i) + \bar{p}_{i-1}(a_i \cdot \bar{b}_i + \bar{a}_i \cdot b_i)$$

$$P_i = \vee 3, 5, 6, 7 = p_{i-1}(\bar{a}_i \cdot b_i + a_i \cdot \bar{b}_i + a_i \cdot b_i) + \bar{p}_{i-1} a_i \cdot b_i = a_i \cdot b_i + p_{i-1}(a_i \oplus b_i)$$



3

Принцип роботи компаратора

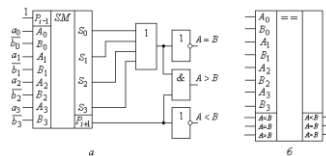
- Компаратор – це пристрій, що виконує порівняння двох чисел.

$$(A > B)_{\text{ВНХ}} = (A > B)_{n-m} + (A = B)_{n-m} \cdot (A > B)_{\text{ВХ}}$$

$$(A = B)_{\text{ВНХ}} = (A = B)_{n-m} \cdot (A = B)_{\text{ВХ}}$$

$$(A < B)_{\text{ВНХ}} = (A < B)_{n-m} + (A = B)_{n-m} \cdot (A < B)_{\text{ВХ}}$$

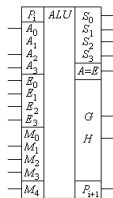
$$y_i = a_i \cdot b_i + \overline{a_i} \cdot \overline{b_i} = a_i \oplus b_i$$



4

Арифметично-логічні пристрої

- Арифметично-логічні пристрої (АЛП, ALU – Arithmetic-Logic Unit) – це спеціалізовані мікросхеми, які виконують арифметичні та логічні операції у відповідності до двійкового коду, який подається на керуючі входи мікросхеми.



Входи вибору функції				Вхід-вихід (логічна логіка)		Вхід-вихід (арифметична логіка)	
M ₃	M ₂	M ₁	M ₀	Логічні функції M ₃ = H	Арифметичні дії M ₃ = L; P = L	Логічні функції M ₃ = H	Арифметичні дії M ₃ = L; P = H
L	L	L	L	1	A - 1	1	A
L	L	L	H	A ∨ E	AE - 1	AE	A ∨ E
L	L	H	L	A ∨ E	AE - 1	AE	A ∨ E
L	L	H	H	1	1	0	1
L	H	L	L	AE	A + (A ∨ E)	AE	A + AE
L	H	L	H	E	A + (A ∨ E)	E	(A ∨ E) + AE
L	H	H	L	AE	A - E - 1	A(-)E	A - E - 1
L	H	H	H	A ∨ E	A ∨ E	AE	AE - 1
H	L	L	L	AE	A + (A ∨ E)	A ∨ E	A + AE
H	L	L	H	A(-)E	A - E	AE	A - E
H	L	H	L	E	AE + (A ∨ E)	E	(A ∨ E) + AE
H	L	H	H	A ∨ AE	AE	AE	AE - 1
H	H	L	L	0	A + A	1	A + A
H	H	L	H	AE	AE + A	A ∨ E	(A ∨ E) + A
H	H	H	L	AE	AE + A	A ∨ E	(A ∨ E) + A
H	H	H	H	A	A	A	A - 1

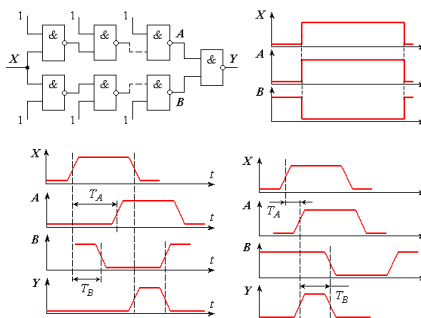
Буферні елементи

- Шина – це паралельна група провідників, по якій передаються інформаційні сигнали в обох напрямках і до якої може бути приєднана досить велика група пристроїв, що передають та приймають інформаційні сигнали:
 - адресні
 - командні
 - шини даних
- Основою будь-якого буферного елемента є двонаправлений підсилювач, який складається з двох ідентичних, зустрічно-паралельно з'єднаних підсилювачів потужності, кожен з яких вмикається лише при передачі інформації від його входу до виходу.

6

Перехідні процеси в логічних схемах

Явище появи на виході Y сигналу, який неможливо передавати на основі алгебри Буля без врахування часових затримок (одержаний сигнал є сигналом перешкоди, що може привести до непередбачених наслідків в послідовних схемах) називається „гонками” або „змаганнями”.



7

Способи боротьби з “гонками”

- *Синхронізація* – по всьому цифровому пристрою створюється єдина система синхронізуючих сигналів (однофазна/ двофазна)
- *побудова протигоночних систем* – вони будуються так, що в них відсутній ризик появи на виході сигналів, не передбачених логікою роботи схеми
- *врахування мінімального часу затримки* – проєктант паралельно зі схемотехнікою закладає й технологічні особливості виготовлення схеми з необхідними параметрами

8

Список використаних джерел

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.

9

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

10

Лекція №6

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція № 6. Тригери. Тригерні схеми.
Загальна характеристика тригерних схем. RS-, D-, JK-, T- та TV-тригери

Слайд 2

Зміст

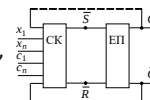
- Тригерні схеми
- Загальна характеристика тригерних схем
- RS-тригери
- D-тригери
- JK-тригери
- T- та TV-тригери

2

Слайд 3

Тригерні схеми

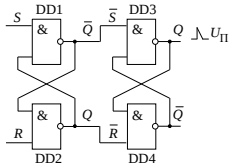
- У цифрових пристроях використовується велика кількість різноманітних тригерів, але всі вони мають у своєму складі *елемент пам'яті* та *схему керування*.
- Керування елементом пам'яті відбувається за допомогою входів *S* (*Set* – установлення стану $Q = 1$,) та *R* (*Reset* – повернення до початкового стану $Q = 0$,).
- *flip-flop* – вибірка входних сигналів і відповідна зміна виходів визначається в моменти дії тактових часових сигналів (синхронні тригери)
- *latch* – змінюють свій стан при зміні входних сигналів незалежно від наявності чи відсутності часових тактових сигналів



3

Загальна характеристика тригерних схем

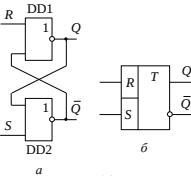
- Тригери за способом введення інформації поділяють на дві групи – *асинхронні* (змінюють свій стан безпосередньо після зміни інформаційних сигналів) та *синхронні* (необхідна наявність синхроімпульсу)



Вхід	Призначення
S	Вхід установки тригерів у стан Q = 1
R	Вхід установки тригерів у стан Q = 0
J	Вхід установки JK-тригерів у стан Q = 1
K	Вхід установки JK-тригерів у стан Q = 0
D	Вхід завантаження інформації у D-тригер
T	Тактовий вхід T-тригерів
V	Підготовчий вхід, дозволу прийому інформації
C	Вхід синхронізації, Виконавчий вхід прийому інформації

4

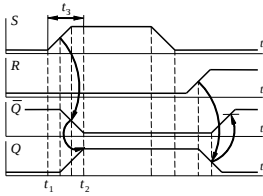
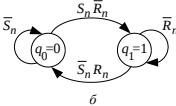
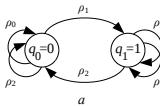
RS-тригери



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

R _n	S _n	Q _{n+1}
0	0	Q _n
0	1	1
1	0	0
1	1	x

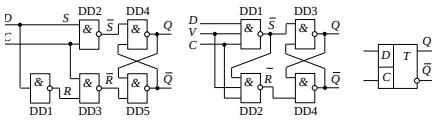
$$Q_{n+1} = S_n + Q_n \cdot \bar{R}_n$$
$$\bar{Q}_{n+1} = R_n + \bar{Q}_n \cdot \bar{S}_n$$



5

D-тригери

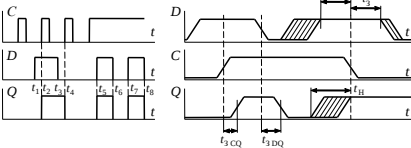
- Функціональна особливість – сигнал на виході Q в (n +1)-ому такті повторює значення сигналу на вході D в n-му такті.



C _n	D _n	Q _n	Q _{n+1}
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

$$Q_{n+1} = C_n D_n$$

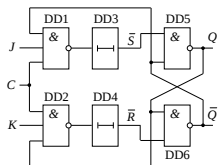
$$Q_{n+1} = C_n D_n + \bar{C}_n Q_n + D_n Q_n$$



6

JK-тригери

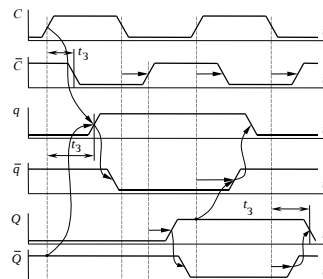
- За логікою роботи подібний до RS-тригерів, але, на відміну від них, не має невизначених переходів.



$$Q_{n+1} = J_n \bar{Q}_n + \bar{K}_n Q_n$$

$$Q_{n+1} = C(J_n \bar{Q}_n + \bar{K}_n Q_n) + \bar{C} Q_n$$

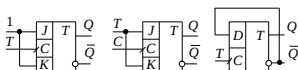
J_n	K_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q}_n$



7

T- та TV-тригери

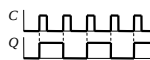
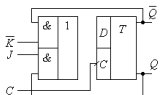
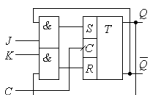
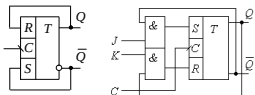
- Відносяться такі схеми, які за сигналом на T-вхід переключаються у протилежний стан.



T-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

$$Q_{n+1} = Q_n \bar{T}_n + \bar{Q}_n T_n$$

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



8

Список використаних джерел

- Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.

9

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

10

Лекція №7

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №7. Синхронізація в цифрових схемах. Узгодження вхідних сигналів. Синхронізатори. Практичні проблеми проектування синхронних цифрових схем

Слайд 2

Зміст

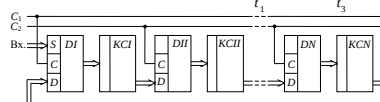
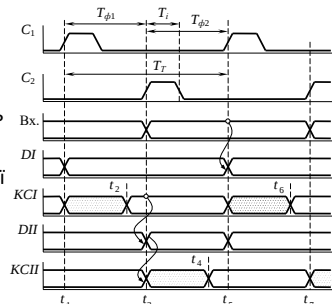
- Синхронізація в цифрових схемах
- Узгодження вхідних сигналів
- Синхронізатори
- Практичні проблеми проектування синхронних цифрових схем

2

Слайд 3

Двофазна синхронізація в цифрових схемах

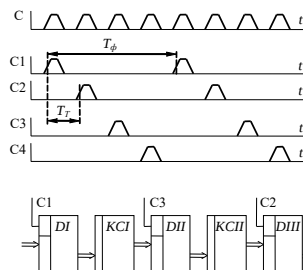
- Схеми синхронізуються двома послідовностями імпульсів C_1 та C_2 однієї частоти $f_T = 1/T_T$ та одного фазового зсуву T_Φ . Тривалість імпульсів двох послідовностей однакова і дорівнює T_i . Для симетричної двофазної синхронізації $T_T = T_\Phi$. Для несиметричної $T_{\Phi 1} \neq T_{\Phi 2}$.



3

Багатофазна синхронізація в цифрових схемах

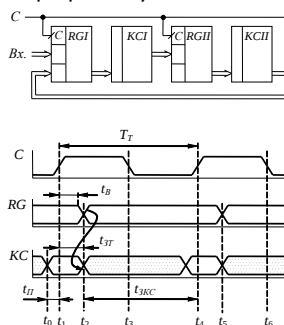
- в усіх схемах з двофазною синхронізацією петля зворотного зв'язку як з логічними елементами, так і без них повинна починатись з виходів тригерів, що синхронізуються однією фазою і закінчуватись на вході тригерів, що синхронізуються іншою фазою
- недопустимі зв'язки, які передають сигнали з виходу однієї групи тригерів на вхід другої, що синхронізується однією і тією ж фазою



4

Однофазна синхронізація в цифрових схемах

- Особливістю однофазної синхронізації є складність її використання при розгалуженій системі синхронізації.



5

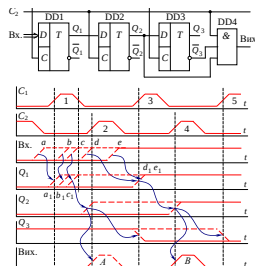
Узгодження вхідних сигналів

- Скорочення фронтів** – ряд джерел інформаційних сигналів має велику тривалість фронтів, яка значно перевищує тривалість фронтів логічних елементів. Навіть незначна неідентичність порогових рівнів логічних елементів призводить до значного часового розкиду моментів їх спрацювання. Для попередження подібних ситуацій необхідно перетворити пологі fronti в круті, які співпадали б з тривалістю фронтів використовуваних логічних елементів. Такі задачі вирішуються за допомогою тригерів Шмітта, які мають підвищені порогов спрацювання.
- Тремтіння контактів** – цей недолік будь-яких систем пов'язаний з вібрацією, що має місце при ударі. Це явище спостерігається в будь-яких контактних системах – реле, кнопках, клавіатурі комп'ютерів.

6

Синхронізатори

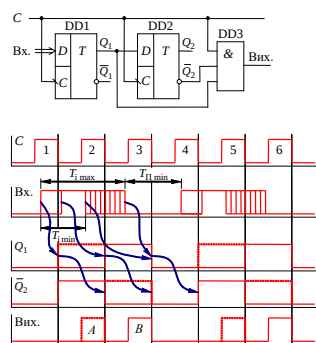
- Прив'язка зовнішніх сигналів до синхроімпульсів необхідна тому, що синхронні цифрові схеми приймають вхідні сигнали без похибок лише в визначені інтервали часу



7

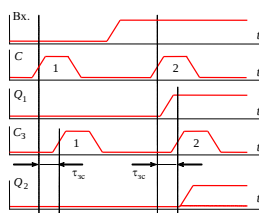
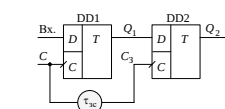
Асинхронний обмін інформацією

- Має місце між цифровими пристроями, кожен з яких має свою власну схему синхронізації. В такому випадку сигнали, що поступають з іншого пристрою, сприймаються приймачем як асинхронні.

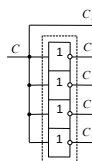


8

Практичні проблеми проектування синхронних цифрових схем



- Для того, щоб зменшити навантаження джерела синхросигналу, створюються копії синхросигналу за схемою



9

Список використаних джерел

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д.
Цифрова схемотехніка: Навч. посібник. –
Львів: Новий світ – 2000, 2009, – 736 с.

10

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

11

Лекція №8

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №8. Загальна характеристика скінченних автоматів. Автомати Мілі і Мура. Закони функціонування автоматів. Способи опису роботи автоматів. Еквівалентні перетворення автоматів

Слайд 2

Зміст

- Загальна характеристика скінченних автоматів
- Автомати Мілі і Мура. Закони функціонування автоматів
- Способи опису роботи автоматів
- Еквівалентні перетворення автоматів

2

Слайд 3

Загальна характеристика скінченних автоматів

- Цифрові пристрої з m тригерами ($m > 1$), стан виходів яких залежить не тільки від значень вхідних сигналів у заданий момент часу, а й від стану використовуваних тригерів у даний та попередній моменти часу, називаються *цифровими автоматами*.
- У загальному плані *цифровим автоматом* називається пристрій, який формує ряд вихідних дискретних сигналів у відповідності із вхідною бінарною комбінацією сигналів.
- Скінченні автомати можуть бути *синхронними*, зміна станів яких відбувається в тактові моменти часу, що задаються зовнішнім генератором.
- В *асинхронних* автоматах зміна станів відбувається внаслідок дії вхідних сигналів без затримок.

3

Загальна характеристика скінченних автоматів

- Абстрактний автомат задається множиною внутрішніх станів $Q=\{Q_1, Q_2, \dots, Q_m\}$ (алфавітом станів), множиною вхідних сигналів $X=\{X_1, X_2, \dots, X_p\}$ (вхідним алфавітом), множиною вихідних сигналів $Y=\{Y_1, Y_2, \dots, Y_k\}$ (вихідним алфавітом) і початковим станом Q_0 .
- Перехід з одного стану в інший визначається функцією переходів f_p , що визначає стан автомата Q_s , в який він переходить з попереднього стану Q_m при дії сигналу X_p : $Q_s = f_p(Q_m, X_p)$
- Значення виходів автомата задається функцією виходів λ , що залежить від стану автомата Q_m і вхідного сигналу X_p : $Y_k = \lambda(Q_m, X_p)$

4

Загальна характеристика скінченних автоматів

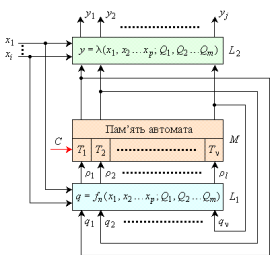
- Якщо прийняти можливу кількість слів вхідного алфавіту за P_x , вихідного алфавіту за K_y , то в залежності від співвідношення між P_x та K_y автомати можуть за функціональним призначенням розділятися на три групи:
 - При $K_y < P_x$ маємо автомати, які призначені для розв'язання задач керування, стиснення інформації, розпізнавання повідомлень.
 - Якщо $K_y > P_x$, то маємо ситуацію, коли довжина вихідних слів буде більшою, ніж вхідних, оскільки кількість слів повинна бути однаковою. Тобто вихідні слова будуть нести збиткову інформацію, що використовується для синтезу перешкодостійких кодів.
 - При $K_y = P_x$ кількість і довжина вхідних і вихідних слів однакові. Такі автомати використовуються для кодових перетворень

5

Автомати Мілі і Мура. Закони функціонування автоматів

$$\begin{cases} Q(t+1) = f_p [Q(t), X(t)]; \\ Y_k(t) = \lambda [Q(t), X(t)]; \end{cases} \quad t = 0, 1, 2, \dots$$

$$\begin{cases} Y_{n+1} = L_2(Q_{n+1}, X_n); \\ Q_{n+1} = L_1(Q_n, X_n); \end{cases} \quad \begin{cases} Y_{n+1} = L_2(Q_{n+1}); \\ Q_{n+1} = L_1(Q_n, X_n). \end{cases}$$



Тип використовуваного тригера	Характеристичне рівняння
Асинхронний RS-тригер	$Q_{n+1} = S_n + \bar{R}_n Q_n$
Синхронний статичний D-тригер	$Q_{n+1} = C_n D_n$
Динамічний D-тригер	$Q_{n+1} = D_n$
Динамічний JK-тригер (JK-MS-тригер)	$Q_{n+1} = J_n \bar{Q}_n + \bar{K}_n Q_n$
T-тригер	$Q_{n+1} = C_n \bar{Q}_n$
T-тригер з дозволяючим входом	$Q_{n+1} = C_n (V \bar{Q}_n + \bar{V} Q_n)$

6

Табличний спосіб опису роботи автоматів

- Таблиці переходів та виходів. Стовбці (рядки) цих таблиць позначають символами з множини Q , а рядки (стовбці) – символами з множини X .
- Кількість рядків таблиці переходів визначається кількістю комбінацій вхідних сигналів p , а кількість стовбців – відповідно, кількість станів M автомата.

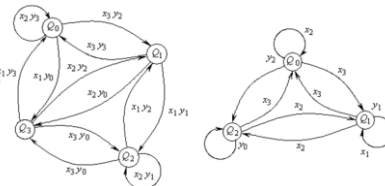
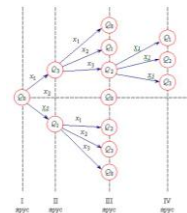
$X \backslash Q$	Q_0	Q_1	Q_2	Q_3
	Q_0	Q_1	Q_2	Q_3
X_1	Q_0	Q_2	Q_1	Q_3
X_2	Q_2	Q_0	Q_3	Q_1
X_3	Q_1	Q_3	Q_2	Q_0

$X \backslash Q$	Q_0	Q_1	Q_2	Q_3
	Q_0	Q_1	Q_2	Q_3
X_1	Q_0	Q_1	Q_2	Q_3
X_2	Q_1	Q_2	Q_3	Q_0
X_3	Q_2	Q_3	Q_0	Q_1

7

Графічний спосіб опису роботи автоматів

- Граф-схеми
- Стрічка цифрового автомата
- Дерево переходів



Тиск	0	1	2	3	4	5	6	7	8	9	10
Вихідний сигнал	X_2	X_1	X_2	X_3	X_1	X_2	X_3	X_1	X_2	X_3	X_1
Стан	Q_0	Q_2	Q_1	Q_3	Q_0	Q_2	Q_1	Q_3	Q_0	Q_2	Q_1
Вихідний сигнал	Y_0	Y_1	Y_2	Y_0	Y_1	Y_2	Y_1	Y_2	Y_1	Y_2	Y_1

8

Еквівалентні перетворення автомата Мілі у автомат Мура

- Для будь-якого автомата Мілі можна побудувати відповідний автомат Мура, і навпаки.

$X \backslash Q$	Q_0	Q_1	Q_2	Q_3
	Q_0	Q_1	Q_2	Q_3
X_1	Q_1	Q_2	Q_0	Q_3
X_2	Q_3	Q_0	Q_2	Q_1
X_3	Q_2	Q_3	Q_1	Q_0

$X \backslash Q$	Q_0	Q_1	Q_2	Q_3
	Q_0	Q_1	Q_2	Q_3
X_1	Q_1	Q_2	Q_0	Q_3
X_2	Q_3	Q_0	Q_2	Q_1
X_3	Q_2	Q_3	Q_1	Q_0

$X \backslash Q$	Q_0	Q_1	Q_2	Q_3
	Q_0	Q_1	Q_2	Q_3
X_1	Q_1	Q_2	Q_0	Q_3
X_2	Q_3	Q_0	Q_2	Q_1
X_3	Q_2	Q_3	Q_1	Q_0

9

Еквівалентні перетворення автомата Мура у автомат Мілі

$$f_{p2}(Q, X) = f_{p1}(Q, X)$$

$$\lambda_2(Q, X) = \lambda_1(f_{p1}(Q, X))$$

- Таблиця переходів еквівалентного автомата Мілі співпадає з таблицею переходів автомата Мура, а в кожному клітинку таблиці виходів записується символ, яким відмічений стан автомата Мура в заданій клітинці.
- При такому перетворенні граф автомата Мілі відрізняється від графу автомата Мура лише тим, що вихідні сигнали з вузлів графа перенесені на всі гілки, що входять у даний вузол.

10

Список використаних джерел

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.
2. Уилкінсон Б. Основы проектирования цифровых схем: Пер. с англ. – М.: Изд. дом «Вильямс», 2004. – 320 с.
3. Самовалов К.Г., Романкевич А.А., Валуйский В.М. Прикладная теория цифровых автоматов. –К: Вища шк., 1987. -372 с.
4. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.К. Прикладна теорія цифрових автоматів. – К. НАУ, 2007. – 364 с.

11

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

12

Лекція №9

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №9. Основи аналізу цифрових автоматів. Аналіз автоматів з D-тригерами. Особливості аналізу скінченних автоматів з JK-тригерами

Слайд 2

Зміст

- Основи аналізу цифрових автоматів
- Аналіз автоматів з D-тригерами
- Особливості аналізу скінченних автоматів з JK-тригерами

2

Слайд 3

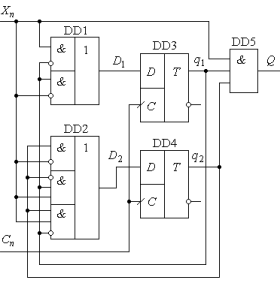
Основи аналізу цифрових автоматів

Аналіз роботи автомата виконується з метою визначення його стану в послідовний тактовий момент часу і передбачення послідовних станів.

1. визначаються стани на послідовному тактовому моменті часу і значення виходів вхідної та вихідної комбінаційних функцій L_1 і L_2 ;
2. використовуються функції L_1 і L_2 для побудови таблиці станів, яка повністю визначає послідовний стан і значення виходу схеми для кожної комбінації поточного стану і виходів;
3. будується діаграма станів, яка містить інформацію з попереднього кроку (граф переходів).

3

Аналіз автоматів з D-тригерами



Збуджуючі рівняння:

$$D_{1n} = X_n q_{1n} + X_n q_{1n} = X_n \oplus q_{1n}$$
$$D_{2n} = X_n q_{2n} + X_n q_{1n} q_{2n} + X_n q_{1n} q_{2n} = X_n q_{2n} + X_n (q_{1n} \oplus q_{2n})$$

Характеристичні рівняння:

$$q_{1(n+1)} = D_{1n} \cdot C_n$$
$$q_{2(n+1)} = D_{2n} \cdot C_n$$

Перехідні рівняння:

$$q_{1(n+1)} = C_n \cdot (X_n \oplus q_{1n})$$
$$q_{2(n+1)} = C_n \cdot (X_n q_{2n} + X_n (q_{1n} \oplus q_{2n}))$$

4

Табличний спосіб опису роботи автомата

Таблиця представлення перехідних рівнянь

C_n	X_n	q_{2n}	q_{1n}	$q_{2(n+1)}$	$q_{1(n+1)}$
1	0	0	0	0	0
2	0	0	1	0	1
3	0	1	0	1	0
4	0	1	1	1	1
1	1	0	0	0	1
2	1	0	1	1	0
3	1	1	0	1	1
4	1	1	1	0	0

Таблиця переходів

C_n	X_n	Q_n	Q_{n+1}
1	0	Q_0	Q_0
2	0	Q_1	Q_1
3	0	Q_2	Q_2
4	0	Q_3	Q_3
1	1	Q_0	Q_1
2	1	Q_1	Q_2
3	1	Q_2	Q_3
4	1	Q_3	Q_0

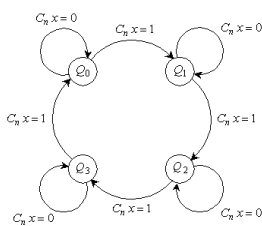
Скорочена таблиця переходів

$X \cdot C_n \backslash Q$	Q_0	Q_1	Q_2	Q_3
0	Q_0	Q_1	Q_2	Q_3
1	Q_1	Q_2	Q_3	Q_0

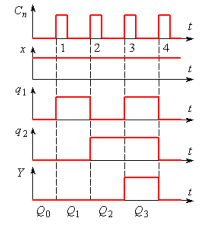
5

Графічний спосіб опису роботи автомата

Граф-схема автомата



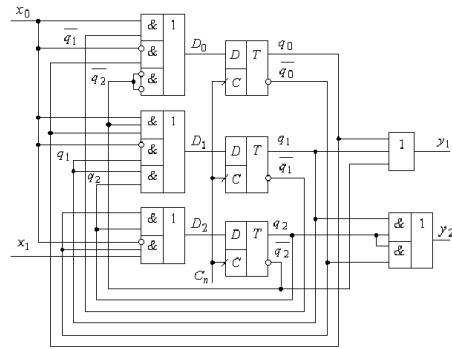
Часові діаграми, що описують роботу схеми



6

Слайд 7

Проаналізувати роботу скінченного автомата, принципова схема якого наведена на слайді. Побудувати його таблицю станів та часові діаграми роботи.



Слайд 8

C_n	x_0	x_1	q_{2n}	q_{1n}	q_{0n}	$q_{2(n+1)}$	$q_{1(n+1)}$	$q_{0(n+1)}$
1	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	1
3	0	0	0	1	0	0	1	0
4	0	0	0	1	1	0	1	1
5	0	0	1	0	0	1	0	1
6	0	0	1	0	1	0	0	1
7	0	0	1	1	0	1	1	1
8	0	0	1	1	1	0	1	1
9	0	1	0	0	0	1	0	0
10	0	1	0	0	1	0	0	1
11	0	1	0	1	0	1	1	0
12	0	1	0	1	1	0	1	1
13	0	1	1	0	0	1	0	1
14	0	1	1	0	1	0	0	1
15	0	1	1	1	0	1	1	1
16	0	1	1	1	1	0	1	1
17	1	0	0	0	0	0	0	1
18	1	0	0	0	1	0	1	1
19	1	0	0	1	0	0	0	0
20	1	0	0	1	1	0	1	0
21	1	0	1	0	0	1	0	1
22	1	0	1	0	1	0	0	1
23	1	0	1	1	0	1	1	1
24	1	0	1	1	1	0	1	1
25	1	1	0	0	0	0	0	1
26	1	1	0	0	1	0	1	1
27	1	1	0	1	0	0	0	0
28	1	1	0	1	1	0	1	0
29	1	1	1	0	0	1	0	1
30	1	1	1	0	1	0	0	1
31	1	1	1	1	0	1	1	1
32	1	1	1	1	1	0	1	1

8

Слайд 9

Особливості аналізу скінченних автоматів з JK-тригерами

Збуджуючі рівняння:

$$J_{0n} = x_0 \overline{x_1} \quad K_{0n} = x_0 \overline{x_1} + x_1 q_{1n}$$

$$J_{1n} = x_0 q_{0n} + x_1 \quad K_{1n} = x_1 \overline{q_{0n}} + x_0 \overline{x_1} q_{0n}$$

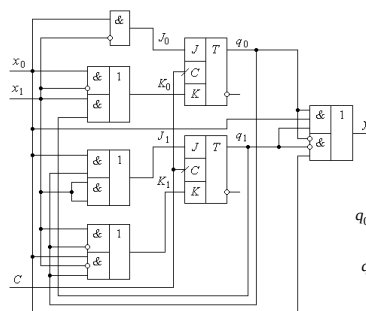
Характеристичні рівняння:

$$Q_{n+1} = (J_n \overline{Q_n} + \overline{K_n} Q_n) C_n$$

Перехідні рівняння:

$$q_{0(n+1)} = C_n (x_0 \overline{x_1} \overline{q_{0n}} + (x_0 \overline{x_1} + x_1) q_{0n})$$

$$q_{1(n+1)} = C_n ((x_0 q_{0n} + x_1) \overline{q_{1n}} + (x_1 \overline{q_{0n}} + x_0 \overline{x_1} q_{0n}) q_{1n})$$



9

Табличний спосіб опису роботи автомата

Таблиця станів

N	x ₀	x ₁	q _{0n}	q _{1n}	q _{0(n+1)}	q _{1(n+1)}
0	0	0	0	0	0	0
1	0	0	0	1	0	0
2	0	0	1	0	1	0
3	0	0	1	1	0	0
4	0	1	0	0	0	1
5	0	1	0	1	0	1
6	0	1	1	0	0	1
7	0	1	1	1	1	0
8	1	0	0	0	0	0
9	1	0	0	1	1	0
10	1	0	1	0	0	1
11	1	0	1	1	1	1
12	1	1	0	0	0	1
13	1	1	0	1	1	1
14	1	1	1	0	1	1
15	1	1	1	1	0	0

Таблиця кодування станів

N	q _{0n}	q _{1n}	Q
0	0	0	Q ₀
1	0	1	Q ₁
2	1	0	Q ₂
3	1	1	Q ₃

Таблиця переходів

Q _n	Q _{n+1}			
	x ₀ x ₁			
	00	01	10	11
Q ₀	Q ₀	Q ₁	Q ₀	Q ₁
Q ₁	Q ₀	Q ₁	Q ₂	Q ₃
Q ₂	Q ₂	Q ₁	Q ₁	Q ₃
Q ₃	Q ₀	Q ₂	Q ₃	Q ₀

10

Графічний спосіб опису роботи автомата

Граф-схема автомата

Рівняння, що описує залежність вихідного сигналу:

$$y = x_0 q_{0n} q_{1n} + x_1 \overline{q_{0n}} \overline{q_{1n}}$$

11

Список використаних джерел

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.
2. Уилкинсон Б. Основы проектирования цифровых схем: Пер. с англ. – М.: Изд. дом «Вильямс», 2004. – 320 с.
3. Самовалов К.Г., Романкевич А.А., Валуйский В.М. Прикладная теория цифровых автоматов. –К: Вища шк., 1987. -372 с.
4. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.К. Прикладна теорія цифрових автоматів. – К. НАУ, 2007. – 364 с.

12

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

13

Лекція №10

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №10. Синтез скінченних автоматів

Слайд 2

Зміст

- Етапи синтезу цифрових автоматів
- Абстрактний синтез цифрових автоматів
- Структурний синтез цифрових автоматів

2

Слайд 3

Задача синтезу скінченних автоматів

Етапи синтезу:

1. Задається закон функціонування автомата;
2. Мінімізується кількість внутрішніх станів автомата;
3. Кодуються стани автомата;
4. Визначаються функції збудження елементів пам'яті і функції виходів, а також забезпечується їх мінімізація;
5. Складається функціональна схема автомата в вибраному елементному базисі.

3

Етап 1. Задання закону функціонування асинхронного автомата

Початковою інформацією для першого етапу є словесний опис алгоритму функціонування автомата або аналіз часових діаграм його роботи. Результатом цього етапу є отримання формалізованого опису закону функціонування в вигляді таблиць переходів, графів переходів.

- Зручно задавати автомат за допомогою *початкових таблиць переходів* (ПТП), оскільки їхня структура є найбільш простою.
- ПТП характеризується тим, що в кожному її рядку є тільки один стійкий стан, а всі переходи між станами прості.

4

Початкова таблиця переходів

- Різноманітним комбінаціям вхідних та вихідних сигналів присуються відповідні номери, які характеризують всі можливі стани автомата. За початковий приймемо стан $Q_0=0$, який характеризується значеннями вхідних сигналів $x_1x_0=00$ і значенням вихідного сигналу $y=0$.

Q \ X	$x_1 x_0$				y
	00	01	11	10	
0	(0)	1			0
1		(1)			1

- Стан є *стійким*, тобто таким, який не зміниться без дії вхідних сигналів. В таблиці ПТП, що будується, стійкі стани будемо виділяти дужками.

- Нестійкі* стани автомата – такі, які можуть змінюватися на стійкі без зміни вхідних сигналів. Такі стани в ПТП будемо позначати їх номерами без дужок.

Q \ X	$x_1 x_0$				y
	00	01	11	10	
0	(0)	1	—	2	0
1	0	(1)	4	—	1
2	0	—	3	(2)	0
3	—	5	(3)	2	0
4	—	1	(4)	2	1
5	0	(5)	3	—	0

- У першому рядку ПТП залишилась незаповненою клітка з $X=11$. Оскільки стан $Q=0$ є стійким при $X=00$, то перехід в новий стан, що відповідає $X=11$, вимагає одночасної зміни обох вхідних сигналів x_1 і x_0 , що практично неможливо. Оскільки перехід $00 \rightarrow 11$ виконати неможливо, в ПТП в відповідній клітинці проставляється риска, яка інформує про наявність невизначеного стану.

5

Етап 2. Мінімізація кількості внутрішніх станів автомата

- Формальна процедура мінімізації кількості станів полягає в визначенні еквівалентних та псевдоеквівалентних станів, де два стани є *еквівалентними*, якщо неможливо визначитись з різницею в їх функціонуванні, розглядаючи лише поточні і майбутні стани виходів автомата. Пара еквівалентних станів може бути замінена одним.
- Скорочення частини станів асинхронного автомата базується на виявленні еквівалентних і псевдоеквівалентних станів, а також на виявленні і об'єднанні сумісних внутрішніх станів автомата (рядків ПТП).

6

Еквівалентні та псевдоеквівалентні стани

Еквівалентними станами автомата називаються стійкі стани, які задовольняють наступним умовам:

- Їм відповідає один і той же стан входу (тобто вони знаходяться в одному і тому ж стовбці ПТП);
- Їм відповідає один і той же стан виходу (однакові вихідні сигнали автомата);
- Будь-якій послідовності станів входу автомата відповідає одна і та ж сама послідовність станів його виходу, незалежно від того, який зі стійких станів автомата прийнято за початковий.

Псевдоеквівалентними називаються такі два стійкі стани автомата, яким:

- відповідає один і той самий стан входу автомата;
- відповідають стани виходів автомата, між якими немає протиріччя;
- будь-якій послідовності станів входу автомата відповідають непротирічні послідовності станів його виходу, незалежно від того, який з цих стійких станів взятий за початковий.

7

Сумісні стани

Під *сумісним внутрішнім станом* розуміють два або більше станів, яким відповідають рядки з розміщенням цифр в них, яке не має протиріччя, тобто такі рядки, в одному і тому ж стовбці яких знаходяться однакові цифри (або в одному рядку цифра, а в іншому – риска).

Об'єднання рядків з поєднаними стійкими станами виконується на основі наступних правил:

- два або більше рядків можуть бути об'єднані, якщо значення вихідних змінних (станів виходу) для цих рядків не мають протиріччя для автомата Мура і можуть бути будь-якими для автомата Мілі, а номери станів, що записані в одних і тих же стовпцях, співпадають між собою, або з прочерком;
- при об'єднанні станів з однаковими номерами в дужках і без них результуючий стан повинен бути у дужках;
- якщо при об'єднанні станів в одному з рядків знаходиться прочерк, а в інший – номер стану, то в скороченій таблиці пишеться номер стану.

8

Етап 3. Кодування станів автомата

Найважливішою задачею, що має місце при кодуванні станів автомата є виключення значень його елементів пам'яті при заданих переходах. Таке кодування називається протигоночним. Найпростішим способом протигоночного кодування є сусіднє кодування, при якому два стани, що пов'язані між собою простими переходами кодуються наборами двійкових чисел, що відрізняються станом лише одного елемента пам'яті (ЕП).

Основними вимогами до автомата, в якому забезпечується сусіднє кодування станів є:

- в графі автомата не повинно бути замкнених контурів, що містять непарну кількість вершин;
- два сусідні стани другого порядку не повинні мати більше двох станів, що лежать між ними. При цьому під станами другого порядку маються на увазі два стани, шлях між якими по графу автомата складається з двох ребер (незалежно від орієнтації).

9

Етап 4. Визначення функцій збудження елементів пам'яті і функцій виходів автомата

- Функції збудження автомата залежать від використовуваних типів тригерів, що вибираються в якості ЕП.
- Для визначення функції виходу карта Карно будується на основі скороченої таблиці переходів, в якій стійкі стани замінюються значеннями у відповідному рядку, а нестійкі – символом *

10

Етап 5. Складання функціональної (принципової) схеми автомата в вибраному елементному базисі

- Отримані на попередньому етапі вирази функцій збудження ЕП і функції виходу дають можливість легко розробити функціональну і принципову схеми автомата.

11

Список використаних джерел

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.
2. Уилкинсон Б. Основы проектирования цифровых схем: Пер. с англ. – М.: Изд. дом «Вильямс», 2004. – 320 с.
3. Самовалов К.Г., Романкевич А.А., Валуйский В.М. Прикладная теория цифровых автоматов. –К: Вища шк., 1987. –372 с.
4. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.К. Прикладна теорія цифрових автоматів. – К. НАУ, 2007. – 364 с.

12

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

13

Лекція №11

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №11. Синтез асинхронних імпульсних автоматів. Особливості синтезу синхронних автоматів. Використання теореми Шенона при синтезі скінченних автоматів на основі JK-тригерів

Слайд 2

Зміст

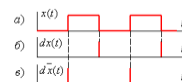
- Синтез асинхронних імпульсних автоматів
- Особливості синтезу синхронних автоматів
- Використання теореми Шенона при синтезі скінченних автоматів на основі JK-тригерів

2

Слайд 3

Синтез асинхронних імпульсних автоматів

- Асинхронними імпульсними автоматами називаються такі, на які входні сигнали впливають лише на короткі інтервали часу зміни їх рівня з "0" в "1" або навпаки.
- Для аналітичного формування закону функціонування асинхронних імпульсних автоматів вводиться оператор переходу d . Використання оператора переходу до потенційних сигналів $x(t)$ дає імпульсний сигнал $dx(t)$, який приймає одиничне значення в момент переходу $x(t)$ з "0" в "1".
- Задача синтезу асинхронних імпульсних автоматів може бути зведена до задачі синтезу асинхронних потенційних автоматів шляхом розробки методики перетворення функцій переходів, в які входять оператори d , в функції переходів виходів, що не містять операторів d . Таке перетворення називається інтегруванням функцій переходів імпульсного автомата або інтегруванням імпульсного автомата.



3

Приклад синтезу імпульсного JK-тригера

Таблиця повної функції переходів

dJ	dK	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Початкова таблиця переходів

Q	$X(J, K)$	00	01	11	10
0	(0), 0	2	4	6	
1	(1), 1	3	5	7	
2	0	(2), 0	5	6	
3	0	(3), 1	5	7	
4	1	3	(4), 0	6	
5	1	3	(5), 1	7	
6	1	3	4	(6), 0	
7	0	3	4	(7), 1	

Аналітичний вираз для закону функціонування (характеристичне рівняння)

$$Q_{n+1} = dJ\bar{Q}_n + d\bar{K}Q_n$$

4

Особливості синтезу синхронних автоматів

- У синхронних автоматах, на відміну від асинхронних, використовується тактовий сигнал C , що задає моменти часу, в які можуть мати місце зміни внутрішніх станів автомата.
- Здебільшого використовуються D - або JK -тригери.
- Інформаційні сигнали лише керують змінами станів тригерів, тому, оскільки на всі тригери автомата синхросигнал подається одночасно, то вони і спрацьовують одночасно, незалежно від моменту подачі інформаційних сигналів.
- Характерним є дискретний час, і стан автомата можна розглядати лише в дискретні моменти часу. Це дозволяє виключити з розгляду перехідні процеси, що протікають у комбінаційній схемі при зміні її вхідних сигналів.
- Особливості проектування:
 - комбінаційні схеми можуть проектуватись не вільними від гонок;
 - допускаються несусідні зміни станів входу;
 - допускаються критичні змагання ЕП, тобто допускається несусіднє кодування внутрішніх станів.

5

Використання теореми Шенона при синтезі скінченних автоматів на основі JK-тригерів

- Функція переходів будь-якого тригера скінченного автомата $Q_{(n+1)i} = f(x_0 \dots x_n; Q_0 \dots Q_n)$
- Теорема Шенона $f(x_{n-1}, \dots, x_i, \dots, x_0) = \bar{x}_i \cdot f_1 + x_i \cdot f_2$
- Відносно стану заданого i -го тригера $Q_{(n+1)i} = f_{1i} \cdot \bar{Q}_i + f_{2i} \cdot Q_i$
- Для JK -тригера $f_{1i} = J$ і $\bar{f}_{2i} = K$

$$J_i = f_{1i} = Q_{(n+1)i} \Big|_{Q_i=0}$$

$$\bar{K}_i = f_{2i} = Q_{(n+1)i} \Big|_{Q_i=1}$$

$$K_i = \bar{f}_{2i}$$

6

Список використаних джерел

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.
2. Уилкинсон Б. Основы проектирования цифровых схем: Пер. с англ. – М.: Изд. дом «Вильямс», 2004. – 320 с.
3. Самовалов К.Г., Романкевич А.А., Валуйский В.М. Прикладная теория цифровых автоматов. –К: Вища шк., 1987. -372 с.
4. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.К. Прикладна теорія цифрових автоматів. – К. НАУ, 2007. – 364 с.

7

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

8

Лекція №12

Слайд 1

Лекція № 12

З курсу: “Цифрові інформаційні системи”

Колективна поведінка автоматів

Слайд 2

Мета лекції

Дана лекція присвячена розробці нових автоматних моделей, побудові моделей колективної поведінки автоматів, вивченню надійності колективної поведінки і опису прикладу застосування методів моделювання гри в розміщенні для вирішення так званої задачі про розподіл обчислювальної техніки.

Слайд 3

Базові моделі автоматів

В основі базових понять і моделей теорії колективної поведінки і управління лежить гіпотеза простоти, висловлена М.Л. Цетлін: «будь-яка досить складна поведінка складається із сукупності простих поведінкових актів»

Постановка задачі

В якості об'єктів, поведінку яких будемо далі вивчати, використовуються скінченні автомати, тобто об'єкти, що мають кінцеве число внутрішніх станів $\varphi_i (i = 1, 2, \dots, n)$, кінцеве число вхідних сигналів $S_l (l = 1, 2, \dots, g)$ і кінцеве число вихідних сигналів $f_m (m = 1, 2, \dots, k)$.

Автомат з лінійною тактикою

Позначимо через n глибину пам'яті автомата. Чим більше n , тим більше інерційним є автомат. З ростом глибини пам'яті зростає доцільність поведінки автомата в стаціонарних середовищах. Обернена ситуація при малому значенні n .

Конструкція автомата за Криловим

При надходженні сигналу «виграш» зміна станів в ньому відбувається аналогічно автомату Цетліна. Але при сигналі «програш» такий автомат не поспішає міняти стан. Спочатку він «підкидає монетку» і по результату підкидання або переходить в інший стан, або зберігає той стан, в якому автомат отримав сигнал «штраф».

Гра в розміщення

В цій грі беруть участь n однакових автоматів. Кожний автомат $(\alpha = 1, 2, \dots, n)$ має $k > n$ стратегій $f_1, f_2, \dots, f_k$. Гра задається набором k чисел: $1 \geq \alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_k \geq 0$, величина α_j називається потужністю стратегії f_j .

Якщо автомат в момент t вибрав стратегію $f_{i\alpha}$ ($\alpha = 1, 2, \dots, n$), то набір $f(t) = (f_{i1}, f_{i2}, \dots, f_{in})$ стратегій, що вибрані автоматом, називається партією гри. Математичне очікування виграшу автомата в цій партії визначається за формулою:

$$M_{\alpha}(f_{i_1}, f_{i_2}, \dots, f_{i_v}) = \frac{a_{i_{\alpha}}}{m_{i_{\alpha}}}$$

Критерієм доцільності поведінки автомата може бути середній виграш за T партій:

$$W_{\alpha}(T) = \frac{1}{T} \sum_{t=1}^T U_{\alpha}(t)$$

Гра в "розміщення з загальною касою"

З грою B в розміщенні зв'язана ще одна гра – гра B^c в «розміщення з загальною касою». Визначається ця гра так само як і гра B , а виграш автоматів задається формулою:

$$M_{\alpha^c}(f_{i_1}, \dots, f_{i_v}) = \frac{1}{v} \sum_{\alpha=1}^v M_{\alpha}(f_{i_1}, \dots, f_{i_v})$$

Надійність в поведінці автоматів

Поведінка автоматів в грі з розміщенням володіє надійністю. Розглянемо гру B^c . В цій грі автомати так розприділяються по стратегіям, щоб отримати максимальний сумарний виграш. В кожний момент часу частка, що приноситься в сумарний виграш різними автоматами, різна (найбільший вклад вносить автомат, що обере стратегію f_1 і т.д.).

Будемо вважати, що автомати можуть виходити з ладу (закінчувати брати участь у грі). Тоді вихід з ладу автомата переводить гру $B^c(n)$ в гру $B^c(n-1)$, що відрізняється від $B^c(n)$ тільки числом n учасників. Так як для гри $B^c(n-1)$ зайнятими виявляються (якщо реалізується точка Мора) перші $(n-1)$ стратегій, то через деякий час після виходу автомата з ладу незайнятою виявиться n -та стратегія незалежно від того, який саме автомат вийшов з ладу.

Приклад №1

У грі беруть участь p автоматів з величиною пам'яті $n=10$. Задано сім стратегій з потужностями: $a_1=0.9$, $a_2 = a_3 = \dots = a_7 = 0.33$. Результат моделювання наведений в табл. 1, де W і S — середні виграші учасників в іграх B і B^c відповідно, коли число учасників рівне n .

Таблиця 1

v	5	4	3	2	1
W	0,38	0,39	0,41	0,45	0,9
W^c	0,44	0,47	0,52	0,61	0,9

Задача про розподіл обчислювальної техніки

Нехай є деяка множина задач $C = c^i(a_1^i, \dots, a_p^i), i = 1, 2, \dots, N$. Рішення задачі c^i заключається в виборі числа β із множини $\{1, 2, \dots, p\}$, яке забезпечує виграш a_β^i . Цей вибір будемо називати вибором стратегії. Припустимо, що ці задачі виникають випадково і p_i — імовірність появи задачі c^i . Ми аж ніяк не припускаємо, що параметри a_β^i і p_i раніше задані.

Припустимо тепер, що є M обчислювальних машин $R_k, k = 1, 2, \dots, M$. Кожна з них вмикається при появі індексу i (мітки) тої задачі, на яку вона налаштована. При вмиканні обчислювальна машина обирає одне з чисел $\beta \in \{1, 2, \dots, p\}$, при цьому математичне очікування її виграшу рівне a_β^i .

У випадку, коли $M=N$, було б логічно налаштувати кожну з машин на свою мітку, тоді робота кожної з них звелася б до функціонування в стаціонарному середовищі i .

Організація колективної поведінки обчислювальних машин

Задачу про розподілення обчислювальної техніки можна розглядати як задачу про організацію колективної поведінки обчислювальних машин. Потрібно намагатися організувати їх взаємодію так, щоб доцільна поведінка окремих обчислювальних машин привела до оптимальної поведінки всієї системи, що виконує задачу. Для цього потрібно, щоб найбільш вигідним для кожної з обчислювальних машин був такий спосіб настройки, який забезпечив би найбільший сумарний виграш. Це досягається процедурою введення загальної каси.

З іншого боку, в кожен такт виграш кожної обчислювальної машини залежить не тільки від мітки, обраної ним для вирішення завдання, а й від того, наскільки точно він це завдання вирішує. З цього випливає, що при будь-якій фіксованій настройці обчислювальних машин на мітки повинен пройти деякий час для того, щоб обчислювальні машини зуміли здійснити досить точний вибір стратегій.

Ця особливість задачі про розподіл обчислювальної техніки породжує необхідність організації двох рівнів для її рішення

Рівні рішення задачі про розподіл обчислювальної техніки

В кожному такт $t = 1, 2, \dots$, пред'являється та чи інша задача з множини $C^1, C^2, \dots, C^N$, вмикається відповідна обчислювальна машина, відбувається вибір стратегії і оцінка цього вибору. В моменти $\tau, 2\tau$ відбувається оцінка якості вибору міток і зміна налаштування на мітки.

В кожному момент $t = 1, 2, \dots$ дія кожної обчислювальної машини характеризується двома числами (β, i) — стратегії, що обирається β ($\beta = 1, 2, \dots, p$) і номером мітки i ($i = 0, 1, \dots, N$).

В якості обчислювальних машин R_k ($k = 1, 2, \dots, M$) будемо використовувати пари автоматів з лінійною тактикою $(A_k^{(1)}, A_k^{(2)})$ ($k = 1, 2, \dots, M$). Діями автомата $A_k^{(1)}$ є різні стратегії $\beta = 1, 2, \dots, p$; діями автомата $A_k^{(2)}$ є вибір мітки $i = 0, 1, \dots, N$. При цьому в інтервалі $t = (r-1)\tau + 1, \dots, r\tau - 1$ стани автоматів $A_k^{(2)}$ і їх дія не змінюються.

Якщо автомати $(A_k^{(1)}, A_k^{(2)})$ ($k = 1, 2, \dots, M$) належать до асимптотичної оптимальної послідовності автоматів, то можна допускати, що в межі (при $m, n \rightarrow \infty$) вони так обирають мітку і стратегії, щоб отримати максимально можливий виграш $W_{\max}^S$.

Окремі випадки

а) система S складається з однієї пари автоматів $(A^{(1)}, A^{(2)})$ нульова мітка відсутня, тобто $i = 1, 2, \dots, N$, тоді

$$W_{\max} = \max_{i=1, 2, \dots, N} \left\{ p_i [\max(a_1^i, a_2^i, \dots, a_p^i)] + \sum_{r \neq i}^N \frac{a_1^r + \dots + a_p^r}{\rho} p_r \right\}.$$

б) система S складається з однієї пари автоматів $(A^{(1)}, A^{(2)})$ але $A^{(2)}$ може мати мітки $i = 0, 1, 2, \dots, N$. Тоді в цьому випадку $W_{\max}^S = \max W_i$ для $i = 0, 1, 2, \dots, N$ і

$$W_0 = \max \left[\sum_{i=1}^N p_i a_1^i, \sum_{i=1}^N p_i a_2^i, \dots, \sum_{i=1}^N p_i a_p^i \right],$$

$$W_i = p_i [\max(a_1^i, a_2^i, \dots, a_p^i)] + \sum_{r \neq i}^N \frac{a_1^r + \dots + a_p^r}{\rho} p_r$$

$$i = 1, 2, \dots, N.$$

в) система S складається з M пар автоматів $(A_k^{(1)}, A_k^{(2)})$ ($k = 1, 2, \dots, M$) автомати $A_k^{(2)}$ можуть мати лише мітки, відмінні від 0. В цьому випадку має місце формула:

$$W_{\max}^S = \max_{i_1, i_2, \dots, i_M} W(i_1, i_2, \dots, i_M),$$

де

$$W(i_1, i_2, \dots, i_M) = \sum_{\delta=1}^M p_{i_\delta} [\max(a_1^{i_\delta}, a_2^{i_\delta}, \dots, a_p^{i_\delta})] +$$

$$+ \sum_{\substack{r \neq i_1 \\ \dots \\ r \neq i_M}}^N \frac{a_1^r + a_2^r + \dots + a_p^r}{\rho} p_r.$$

г) система S складається з M пар автоматів $(A_k^{(1)}, A_k^{(2)})$ ($k = 1, 2, \dots, M$), автомати $A_k^{(2)}$ можуть мати мітки $i = 0, 1, 2, \dots, N$, тоді виграш W системи при тій чи іншій настройці на мітки визначається формулою з випадку в), якщо немає автомата з нульовою міткою, а якщо один з автоматів $A_k^{(2)}$ налаштований на мітку 0, то

$$W_{(i_a=0, i_1, \dots, i_M)} = \sum_{\substack{\delta=1 \\ \delta \neq a}}^M p_{i_\delta} [\max(a_1^{i_\delta}, a_2^{i_\delta}, \dots, a_p^{i_\delta})] +$$

$$+ \left(1 - \sum_{\substack{\delta=1 \\ \delta \neq a}}^M p_{i_\delta} \right) \max \left[\sum_{\substack{r \neq i_1 \\ \dots \\ r \neq i_M}}^N p_r a_1^r, \dots, \sum_{r=1}^N p_r a_p^r \right],$$

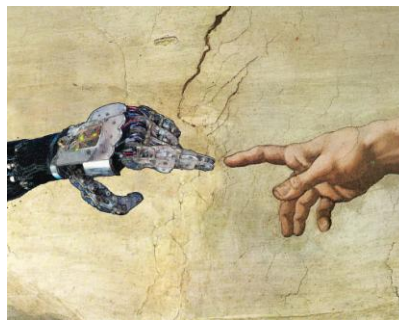
$$W_{\max}^S = \max_{i_1, i_2, \dots, i_M} [W_{(i_a=0, i_1, \dots, i_M)}, W_{(i_1, i_2, \dots, i_M)}].$$

Приклад №2

З імовірністю 0.5 на вході з'являються середовища $c^1 = (0.6; 0.9)$, $c^2 = (0.45; -0.45)$. В грі бере участь одна пара автоматів $A^{(1)}$, $A^{(2)}$ пам'ять n автомата $A^{(1)}$ рівна п'яти, пам'ять $A^{(2)}$ — семи. Налаштування на мітки перевіряється кожні 20 тактів ($\tau = 20$). Мітка може бути будь-якою від 0 до 2.

Із формул для випадку б) отримуємо наступні теоретичні виграші: $W_0 = 0.525$, $W_1 = 0.45$, $W_2 = 0.6$. Автомат $A^{(2)}$ обирає мітку 2. Виграш системи W^s в середньому дорівнює 0.6.

Дякую за увагу!



Лекція №13

Слайд 1



Слайд 2

ЩО ВОНО ТАКЕ???

Теорія ігор аналізує прийняття рішень економічними суб'єктами, яких називають, за традицією, **гравцями**, в ситуаціях, коли на результати цих рішень впливають дії, що вживаються іншими економічними суб'єктами. Такі ситуації прийнято називати **іграми**.

Гра - математична модель конфліктної ситуації
Хід у грі - вибір і здійснення гравцем однієї з передбачених правилами гри дій
Стратегія - послідовність всіх ходів до закінчення гри

Слайд 3

ЩО ЗА ІГРИ ТАКІ?

В іграх з **нульовою сумою** одні гравці виграють за рахунок інших, тобто сумарний виграш всіх гравців дорівнює нулю.

Парні ігри з нульовою сумою називаються **антагоністичними**.
 Кінцеві антагоністичні ігри називаються **матричними іграми**.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} = (a_{ij})_{m \times n}$$

$\alpha = \max_i \min_j a_{ij} = \max_i \alpha_i$
Нижня ціна гри (максимін):
 $\beta = \min_j \max_i a_{ij} = \min_j \beta_j$
Верхня ціна гри (мінімакс):

Чиста стратегія гравця - це можливий хід гравця, обраний ним з ймовірністю, яка дорівнює одиниці.
Змішаною стратегією першого (другого) гравця називається вектор

$$\bar{x} = (x_1, x_2, \dots, x_m), \quad x_i \geq 0, \quad i = \overline{1, m}, \quad \sum_{i=1}^m x_i = 1$$

$$\bar{y} = (y_1, y_2, \dots, y_n), \quad y_j \geq 0, \quad j = \overline{1, n}, \quad \sum_{j=1}^n y_j = 1$$

ПАМ'ЯТАЄТЕ ПОПЕРЕДНІЙ СЛАЙД?

Якщо $x_i > 0, y_j > 0$, гра називається **активною**

Стратегії $\bar{x} = (x_1^*, x_2^*, \dots, x_m^*), \bar{y} = (y_1^*, y_2^*, \dots, y_n^*)$

називаються **оптимальними**, якщо для довільних стратегій $\bar{x} = (x_1, x_2, \dots, x_m), \bar{y} = (y_1, y_2, \dots, y_n)$

виконується умова $f(\bar{x}, \bar{y}) \leq f(\bar{x}^*, \bar{y}) \leq f(\bar{x}^*, \bar{y}^*)$

Рішенням гри називається сукупність оптимальних стратегій і ціни гри.



А ТЕОРЕМИ Є?

Теорема (про активні стратегії). Якщо один гравець дотримується своєї оптимальної змішаної стратегії, то виграш залишається незмінним і рівним ціні гри, якщо інший гравець не виходить за межі своїх активних стратегій.

Теорема фон Неймана (основна теорема матричних ігор). Будь-яка матрична гра має принаймні одне рішення в змішаних стратегіях - дві оптимальні стратегії і відповідну їм ціну $\langle \bar{x}^*, \bar{y}^*, v = f(\bar{x}^*, \bar{y}^*) \rangle$



ЩЕ ТРОХИ ВИЗНАЧЕНЬ

СТАТИЧНІ ІГРИ З ПОВНОЮ ІНФОРМАЦІЄЮ

Під **статичною** розуміють таку гру, в якій всі її учасники приймають рішення, не знаючи, які саме рішення приймають інші.

Під **іграми з повною інформацією** розуміють гри, в яких кожен з гравців точно знає характеристики інших гравців.

Гра в нормальній формі задається наступною сукупністю об'єктів

$$G = \{S_i, u_i, i \in N\}$$

$N = \{1, 2, \dots, n\}$ - множина гравців

де S_i - множина стратегій i -го

$u_i : \prod_{i \in N} S_i$ - функція виграшу гравця i

Для гри в нормальній формі

стратегія $G = \{S_i, u_i, i \in N\}$
 S_i' строго домінує стратегію S_i''

якщо $u_i(s_i', s_{-i}) > u_i(s_i'', s_{-i}), \forall s_{-i} \in S_{-i}$



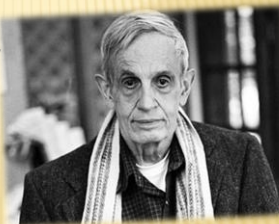
Рівновагою в (строго) домінуючих стратегіях називається профіль (строго) домінуючих стратегій, якщо такі стратегії існують для кожного гравця.

НЕШ?ХМ...ЩОСЬ ЗНАЙОМЕ

Рівновагою Неша (РН) в грі в нормальній формі називається такий профіль стратегій s^* всіх гравців, що

$$u_i(s^*) \geq u_i(s_i, s_{-i}^*), \quad \forall i \in N, \quad \forall s_i \in S_i$$

Сенс рівноваги Неша в тому, що жодному з гравців не вигідно відхилитися від своєї рівноважної стратегії поодиночі



ДИНАМІЧНІ ІГРИ

Динамічною називається гра, в якій кожен гравець може зробити кілька ходів, і принаймні один з гравців, роблячи хід, знає, який хід зробив інший гравець (можливо, він сам).

У динамічних іграх розрізняють **повну** і **неповну** інформацію. Якщо всі гравці у динамічній грі з повною і **досконалою** інформацією про правила гри і функції виграшу, то інформацію вважають **повною**.

Кажуть, що динамічна гра володіє **досконалою** інформацією, якщо всі зроблені ходи відразу ж стають відомі всім гравцям.

Іноді динамічну гру зручно представити у вигляді дерева. Таке уявлення називається **розгорнутою формою** гри.



У динамічних іграх з повною і **досконалою** інформацією зручно вирішувати гру методом **зворотної індукції**. Відповідно до методу зворотної індукції гра «розмотується» з кінця. Ігри з **недосконалою інформацією**, це коді здійснюючи черговий хід, гравці знають, що знаходяться в одній з вершин деякої підмножини множини всіх вершин дерева гри (так званої **інформаційної множини**).

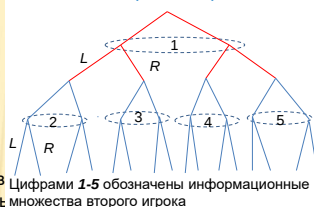
ПОВТОРЮВАНІ ІГРИ

Дуже часто, **ігрові взаємодії** між одними і тими ж гравцями

повторюються. Повторювані гри - це певний клас динамічних ігор з повною, але недосконалою інформацією, який має спеціальне прикладне значення. У разі повторюваних ігор можливо пов'язування стратегій покарання з останніми ходами порушників. Можлив застосування стратегій, що враховують кілька останніх ходів партнерів.

Під **повторюваною** розуміють таку динамічну гру, яка є послідовним повторенням деякої вихідної гри (неважливо, статичної або динамічної).

Дерево гри «Дилемма заключенного», повторення два рази.

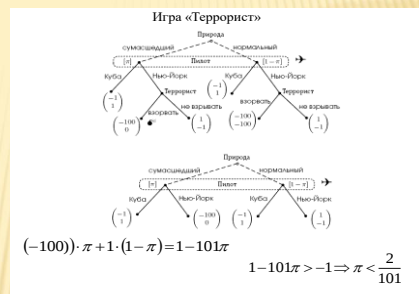


Припустимо, що в будь-який момент часу існує позитивна ймовірність того, що в наступний момент часу гра повториться. Ігрову взаємодію в такому випадку можна моделювати у вигляді гри, що повторюється **нескінченне число разів** (НПІ), за умови, що виграш в кожному наступному періоді дисконтується за певною ставкою.

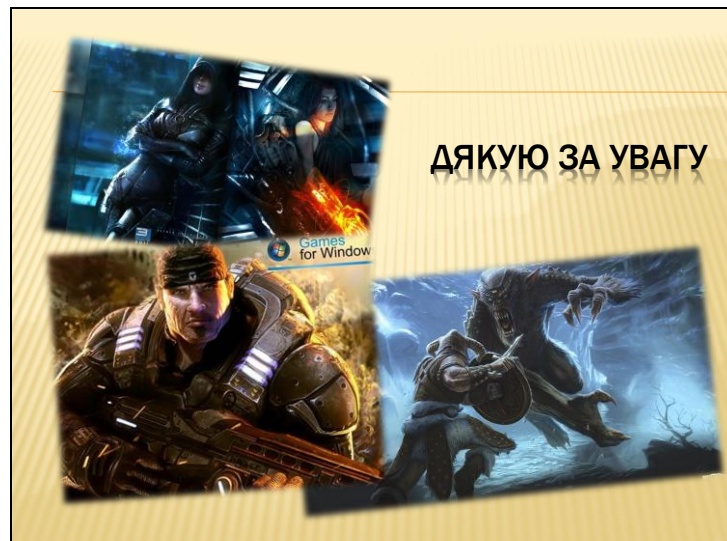
ІГРИ З НЕПОВНОЮ ІНФОРМАЦІЄЮ

Ігри в яких гравці можуть не знати точно переваги інших гравців. Уподобання гравців в цих іграх залежать від випадкових подій, при цьому гравці в різній мірі володіють інформацією про те, яка саме подія відбулася.

Формально це враховується за допомогою введення поняття **типу гравця**: кожен з гравців може бути декількох типів. При цьому вважається, що кожен з гравців знає тільки свій власний тип.



Такого роду гри називають іграми з неповною інформацією або баєсовими іграми.



Лекція №14

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №14. Типи лічильників та особливості їх роботи. Асинхронні лічильники. Лічильники з довільним модулем рахунку. Синхронні лічильники. Скінченні автомати на основі лічильників

Слайд 2

Зміст

- Типи лічильників та особливості їх роботи
- Асинхронні лічильники
- Лічильники з довільним модулем рахунку
- Синхронні лічильники
- Скінченні автомати на основі лічильників

2

Слайд 3

Особливості роботи лічильників

- *Лічильниками* називаються цифрові пристрої (автомати), призначені для підрахунку і фіксації кількості імпульсів, що подаються на їх інформаційні входи або синхровходи.
- Основні параметри:
 - статичний параметр – *коефіцієнт перерахунку* (модуль *перерахунку*) M – характеризує максимальну кількість імпульсів, яка може бути подана на лічильник, щоб привести його до початкового стану.
 - динамічні параметри лічильників характеризують їх швидкодію
 - час *установлення вихідного коду* ($t_{уст}$) – це інтервал часу з моменту подачі вхідного імпульсу до моменту встановлення коду на виходах лічильника;
 - *дозволяюча спроможність лічильника* (t_{cd}) визначається як мінімальний інтервал часу між двома вхідними імпульсами;
 - величина $f_{max} = t_{cd}^{-1}$ називається *максимальною частотою роботи лічильника*.

3

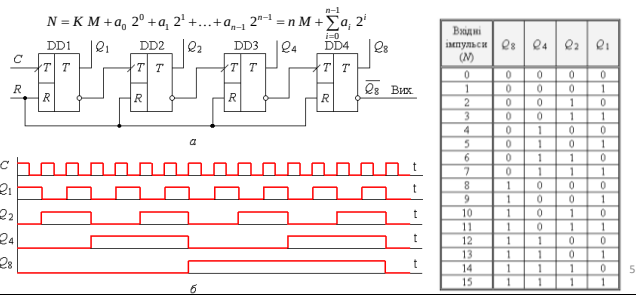
Типи лічильників

- У синхронних лічильниках синхросигнал на всі тригери подається одночасно, а зміна стану тригера відбувається лише тоді, коли на інформаційних входах будь-якого тригера підготовлені відповідні дані.
- В асинхронних лічильниках, які часто називають послідовними, вхідна послідовність імпульсів подається лише на перший тригер, а решта тригерів спрацьовує в залежності від зміни стану попереднього.

4

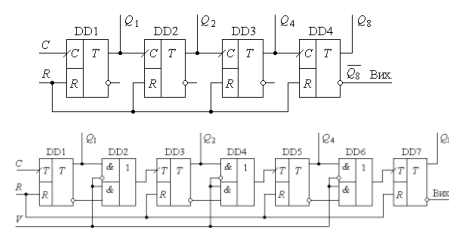
Асинхронні лічильники

- Використовують властивості T-тригера, оскільки він може як зберігати свій стан, так і додавати за модулем 2 вхідний сигнал до інформації, записаної попередньо. Одиночний T-тригер ділить на 2 частоту вхідної послідовності імпульсів. Послідовне включення m таких тригерів дає можливість поділити частоту вхідних імпульсів у $M = 2^m$ разів, або утворює лічильник з коефіцієнтом перерахунку M .



Типи асинхронних лічильників

- Зазвичай код, що визначається вихідними станами тригерів, змінюється у зростаючому напрямку (додаючі лічильники). Операція збільшення вмісту лічильника на одиницю називається інкрементуванням. При зворотній зміні станів лічильники звуться віднімаючими, а якщо напрямок рахунку може змінюватися, то лічильник має назву реверсивного. Операція зменшення вмісту лічильника на одиницю називається декрементуванням. Напрямок рахунку визначається як динамічними властивостями тригерів (перемикання за фронтом чи за спадом), так і способом з'єднання виходів попереднього тригера зі входом наступного.



6

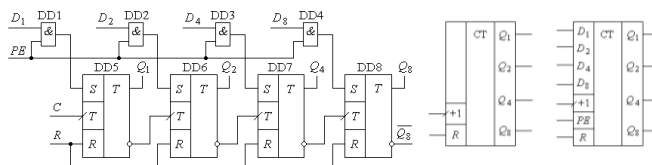
Лічильники з довільним модулем рахунку

- Лічильники з довільним модулем рахунку мають значення M , що відрізняється від цілого ступеню числа 2.
- Способи побудови:
 - виключення зайвих станів (застосування попередньої установки лічильника; використання примусового його обнуління);
 - зворотного зв'язку;
 - кратних модулів.

7

Попередня установка лічильника

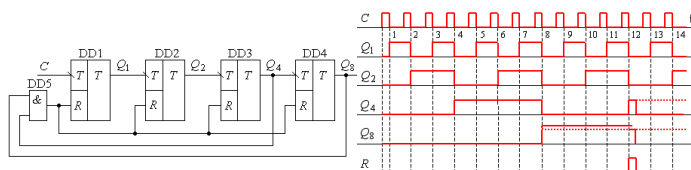
- З моменту подачі вхідних імпульсів лічильник рахуватиме, починаючи не з нуля, а з занесеного коду.



8

Примусове обнуління лічильника

- Із загальної кількості станів M виключаються ті, які своїми значеннями перевищують встановлений модуль перерахунку M_B :
 - обнуління лічильника при досягненні поточним кодом значення M_B ;
 - виключення всіх станів, що перевищують M_B .



9

Спосіб зворотного зв'язку та кратних модулів

- Припустимо, що лічильник містить у собі n тригерів. Виділимо з цієї групи меншу – з n_2 тригерами, тоді $n = n_1 + n_2$. У виділеній групі заводиться зворотній зв'язок так, щоб при появі одиниць на виходах тригерів перший тригер скидався в нуль. Тоді коефіцієнт перерахунку в цій групі становитиме , а загальний коефіцієнт перерахунку: $M = 2^{n_1}(2^{n_2}-1) = 2^n-2^{n_2}$

10

Синхронні лічильники з паралельним переносом

- Інформаційний сигнал, що записується у тригер, підготовляється одразу після встановлення попереднього значення, а синхросигнал подається паралельно на всі тригери.

11

Синтез синхронного лічильника з паралельним переносом з використанням JK-тригерів, який забезпечить коефіцієнт $M = 8$

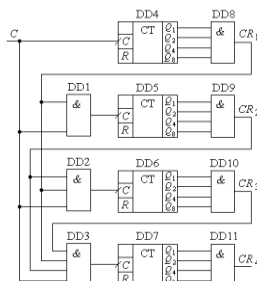
C_n	Q_{4n}	Q_{2n}	Q_{1n}	$Q_{4(n+1)}$	$Q_{2(n+1)}$	$Q_{1(n+1)}$	J_4	K_4	J_2	K_2	J_1	K_1
0	0	0	0	0	0	1	0	*	0	*	1	*
1	0	0	1	0	1	0	0	*	1	*	*	1
2	0	1	0	0	1	1	0	*	*	0	1	*
3	0	1	1	1	0	0	1	*	*	1	*	1
4	1	0	0	1	0	1	*	0	0	*	1	*
5	1	0	1	1	1	0	*	0	1	*	*	1
6	1	1	0	1	1	1	*	0	*	0	1	*
7	1	1	1	0	0	0	*	1	*	1	*	1

$J_N = Q_1 Q_2 \dots Q_{(N-1)} = J_{(N-1)} Q_{(N-1)}$
 $K_N = Q_1 Q_2 \dots Q_{(N-1)} = K_{(N-1)} Q_{(N-1)}$

12

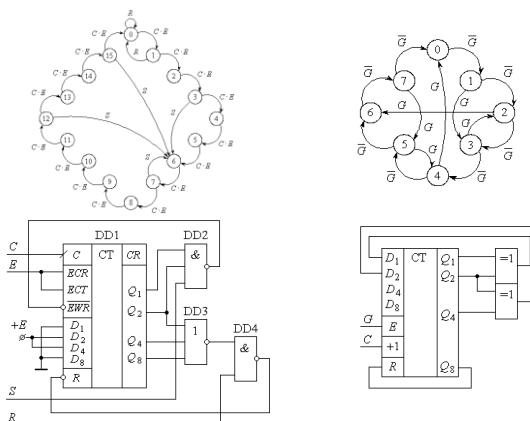
Синхронні лічильники з комбінованим паралельно-паралельним перенесенням

- Синхровхід запаралелюється через елемент АБО допоміжним інверсним входом *notES*. Цей вхід дає можливість організувати рахунок як за фронтом імпульсу, так і за зрізом, а при послідовному нарощуванні використовується як вхід асинхронного вводу від виходу Q_8 попереднього лічильника



13

Скінченні автомати на основі лічильників



14

Список використаних джерел

- Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.
- Уилкинсон Б. Основы проектирования цифровых схем: Пер. с англ. – М.: Изд. дом «Вильямс», 2004. – 320 с.

15

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

16

Лекція №15

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №15. Загальне поняття про регістри.
Регістри пам'яті. Регістри зсуву. Забезпечення обміну інформацією у послідовному форматі.
Регістрові лічильники імпульсів (розподільувачі).
Лічильники Джонсона. Системи контролю цифрової апаратури

Слайд 2

Зміст

- Загальне поняття про регістри
- Регістри пам'яті
- Регістри зсуву
- Забезпечення обміну інформацією у послідовному форматі
- Регістрові лічильники імпульсів (розподільувачі)
- Лічильники Джонсона
- Системи контролю цифрової апаратури

2

Слайд 3

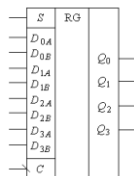
Загальне поняття про регістри

- *Регістрами* називають цифрові пристрої, призначені для тимчасового зберігання та перетворення інформації, що подається на них у вигляді багаторозрядних двійкових чисел.
- Кількість тригерів, розміщених паралельно або з'єднаних послідовно, визначає *розрядність* регістрів.
- Регістри розділяються на дві групи:
 - *накопичувальні* (або *регістри пам'яті*)
 - *послідовні* (або *регістри зсуву*)
 - *послідовний* спосіб введення та виведення інформації
 - *паралельний* спосіб введення та виведення інформації
 - *комбінований* спосіб введення та виведення інформації
- Записана у тригери інформація може виводитись у прямому коді, зворотному або одночасно в прямому та зворотному.
- Регістри зсуву
 - *однонаправлені* (забезпечують зсув лише в одному напрямку)
 - *реверсивні* (двонаправлені)

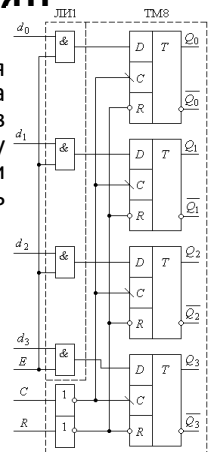
3

Регістри пам'яті

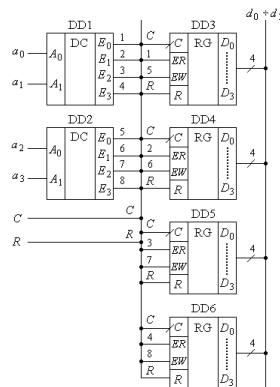
- Регістри пам'яті призначені для запису, тимчасового зберігання та передачі невеликих обсягів інформації, представлені у двійковому коді. Елементами пам'яті в них виступають здебільшого D-тригери.



Входи				Вихід
D_{1A}	D_{1B}	S	C	Q_i
0/1	x	0	L	0/1
x	0/1	1	L	0/1



Регістри, об'єднані єдиною шиною даних

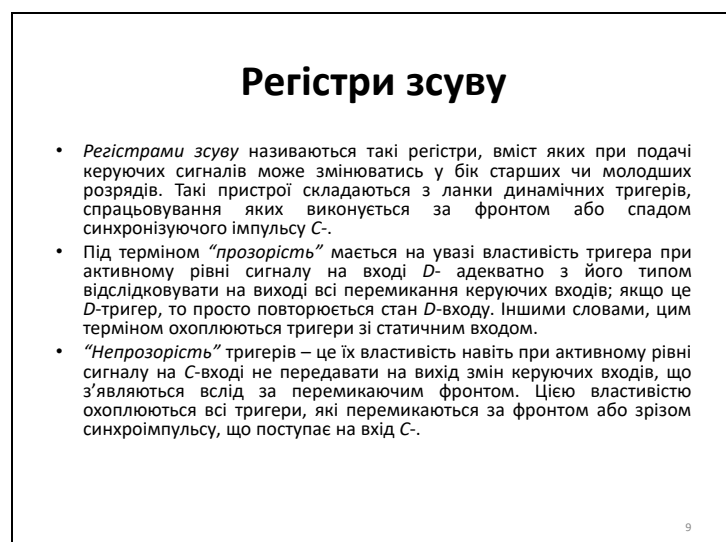
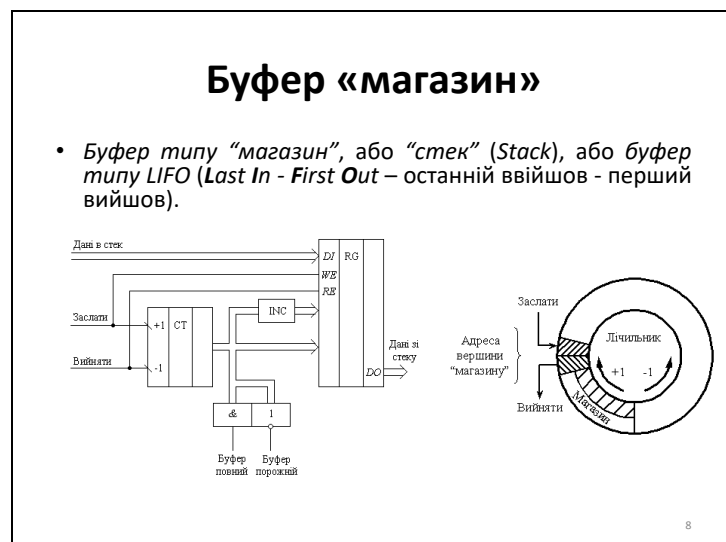
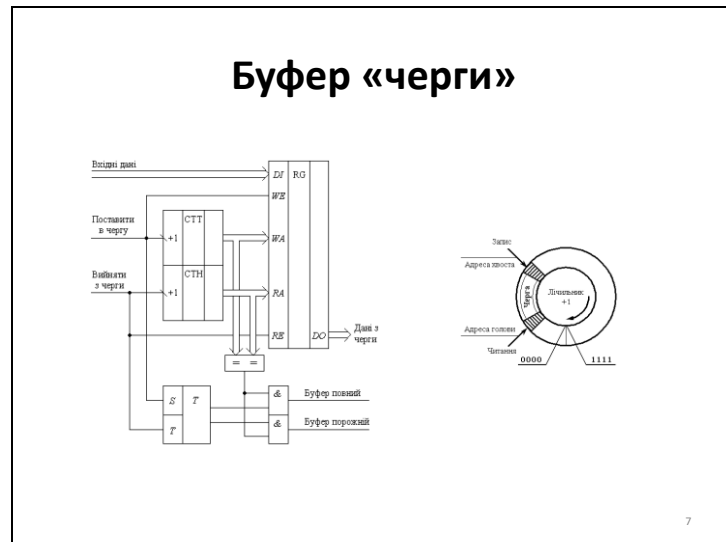


5

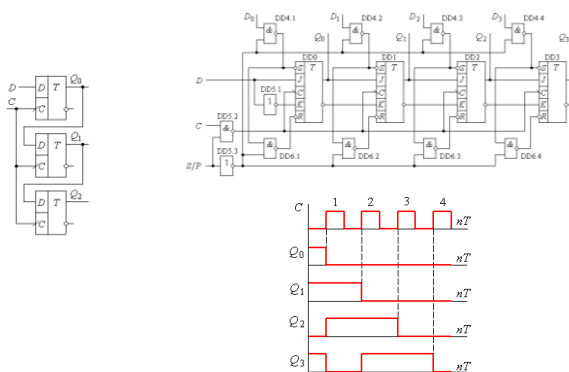
Буфери даних

- Запам'ятовуючі пристрої з "неявно" вираженою адресацією, які служать для зберігання черг.
- Необхідність у такому буфері виникає тоді, коли джерело даних постачає приймачу слова, що розподілені у часі нерівномірно, причому інтервали часу між двома словами можуть іноді бути меншими, ніж час, необхідний приймачу для обробки одного слова. Подібна ситуація може приводити до того, що частина даних втрачатиметься.
- Якщо такі втрати недопустимі, то між джерелом та приймачем включається буфер "черги" для створення і зберігання черги слів для їх послідуочної обробки, або буфер типу FIFO.

6



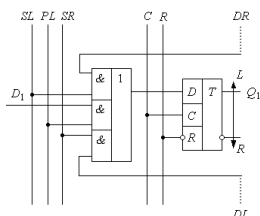
Схеми регістрів зсуву



10

Двонаправлений регістр зсуву

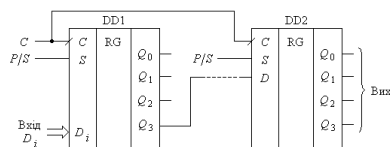
- До D -входу кожного тригера регістра приєднується вихід мультиплексора, який за допомогою керуючих сигналів SL (Shift Left – зсув ліворуч), SR (Shift Right – зсув праворуч) і PL (Parallel Load – паралельне завантаження) приєднує вхід i -го тригера відповідно до виходу послідовного тригера, до виходу попереднього тригера або до i -го розряду паралельного завантаження. За сигналом синхронізації у тригер завантажуються відповідні дані. Вихід i -го тригера приєднується до відповідних входів мультиплексорів сусідніх розрядів (напрямки L, R).



11

Забезпечення обміну інформацією у послідовному форматі

- однофазна трипровідна система
- застосування вирівнюючого регістру, що вмикається перед входом регістра-приймача
- асинхронні способи, в яких синхросигнал не передається по окремому провіднику, а за рахунок спеціального кодування передається по тому ж каналу, що й дані, розмежовуючись з ними у часі
- однопровідна передача
 - синхронізуючий перепад супроводжує кожен біт інформації (фазове і частотне кодування)
 - синхронізуючий перепад, який зветься стартовим бітом, супроводжує один байт



12

Регістрові лічильники імпульсів (розподільувачі)

- Розподільувачами називають функціональні вузли, які циклічно розподіляють потік імпульсів по декількох каналах:
 - кільцеві лічильники
 - лічильники Джонсона
 - лічильники Мебіуса
 - лічильники в коді Лібау-Крейга
 - кільця Реженера

$$Y = f(Q_0 \dots Q_N, \overline{Q_0} \dots \overline{Q_N})$$
$$Q_{0(n+1)} = f(Q_0 \dots Q_N, \overline{Q_0} \dots \overline{Q_N})_n$$
$$Q_{1(n+1)} = Q_{0n}$$
$$\dots$$
$$Q_{N(n+1)} = Q_{(N-1)n}$$

13

Лічильники Джонсона

N тактового імпульсу	Q ₀	Q ₁	Q ₂	Q ₃	Q ₄	Спосіб дешифрації стану	DO
0	0	0	0	0	0	$\overline{Q_0} \cdot \overline{Q_4} = \overline{Q_0} \vee \overline{Q_4}$	1
1	1	0	0	0	0	$Q_0 \cdot \overline{Q_1} = \overline{Q_0} \vee \overline{Q_1}$	1
2	1	1	0	0	0	$Q_1 \cdot \overline{Q_2} = \overline{Q_1} \vee \overline{Q_2}$	1
3	1	1	1	0	0	$Q_2 \cdot \overline{Q_3} = \overline{Q_2} \vee \overline{Q_3}$	1
4	1	1	1	1	0	$Q_3 \cdot \overline{Q_4} = \overline{Q_3} \vee \overline{Q_4}$	1
5	1	1	1	1	1	$Q_4 \cdot Q_0 = \overline{Q_4} \vee \overline{Q_0}$	0
6	0	1	1	1	1	$\overline{Q_0} \cdot Q_1 = \overline{Q_0} \vee Q_1$	0
7	0	0	1	1	1	$\overline{Q_1} \cdot Q_2 = \overline{Q_1} \vee Q_2$	0
8	0	0	0	1	1	$\overline{Q_2} \cdot Q_3 = \overline{Q_2} \vee Q_3$	0
9	0	0	0	0	1	$\overline{Q_3} \cdot Q_4 = \overline{Q_3} \vee Q_4$	0

1. $S_2 \rightarrow S_5 \rightarrow S_{11} \rightarrow S_{23} \rightarrow S_{14} \rightarrow S_{29}$
 $\rightarrow S_{26} \rightarrow S_{20} \rightarrow S_8 \rightarrow S_{17} \rightarrow S_2;$

2. $S_4 \rightarrow S_3 \rightarrow S_{19} \rightarrow S_6 \rightarrow S_{13} \rightarrow S_{27}$
 $\rightarrow S_{22} \rightarrow S_{12} \rightarrow S_{25} \rightarrow S_{18} \rightarrow S_4;$

3. $S_{10} \rightarrow S_{21} \rightarrow S_{10}$

14

Системи контролю цифрової апаратури

- Вузли ділення поліномів використовують також при контролі правильності роботи логічних схем методом сигнатурного аналізу.
- На входи схеми, що перевіряється, з генератора псевдовипадкових чисел подається визначеної довжини та завжди однакова послідовність псевдовипадкових чисел. Вихідна тестова послідовність проходить через поліноміальний розподільувач, який ділить її на достатньо складний поліном G. Поліноміальний розподільувач називають *сигнатурним аналізатором*. Залишок, який залишається в регістрі поліноміального розподільувача після закінчення вихідної тестової послідовності, називають *сигнатурою*. На виході справної цифрової системи сигнатура від заданої цифрової тестової послідовності завжди заздалегідь відома і може бути розпізнана, наприклад, за допомогою компаратора. Сигнатура несправної системи інша, причому, завдяки процедурі ділення та складному поліному G, будь-яка незначна відмінність значно посилюється. На теперішній час широкое використання отримав шістнадцятирозрядний аналізатор. Частка помилок, котра ним не виявляється, оцінюється величиною 2^{-16} .
- Сигнатурні аналізатори використовуються як при стендовому контролі цифрових систем, так і при створенні вбудованих пристроїв самоконтролю. В останньому випадку додатково застосовують так званий принцип "наскрізного регістра".

15

Метод наскрізного зсувного реєстра

- Для повного контролю складних цифрових схем необхідно мати доступ до великої кількості контрольних точок, які повинні бути виведені на роз'єми відповідних плат, або до виводів ВІС. Однак для цього повинна бути велика кількість виводів, що у більшості випадків забезпечити неможливо. У зв'язку з цим в якості контрольних точок зручно використовувати виводи тригерів. З'єднуючи всі тригери аналізованої схеми (а у ряді випадків – і додаткові зовнішні тригери) в один зсувний реєстр, можливо послідовним кодом вводити тестову інформацію, що аналізує контрольні точки. Зазвичай для вирішення цієї задачі використовується третій стан мікросхем, тригери відключаються від основної схеми та заводяться спеціальні серії синхросигналів. Перевірка працездатності наскрізного реєстра здійснюється за допомогою генераторів псевдовипадкових послідовностей та сигнатурних аналізаторів.

16

Список використаних джерел

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.
2. Уилкинсон Б. Основы проектирования цифровых схем: Пер. с англ. – М.: Изд. дом «Вильямс», 2004. – 320 с.

17

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

18

Лекція №16

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №16. Постійні запам'ятовуючі пристрої. Репрограмовані ПЗП.
Основи побудови флеш пам'яті.
Оперативні запам'ятовуючі пристрої

Слайд 2

Зміст

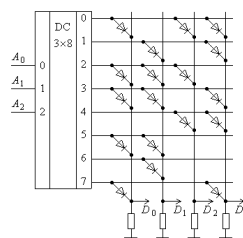
- Постійні запам'ятовуючі пристрої
- Репрограмовані ПЗП
- Основи побудови флеш пам'яті
- Оперативні запам'ятовуючі пристрої

2

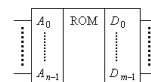
Слайд 3

Одновимірні постійні запам'ятовуючі пристрої

- Пристрої пам'яті, вміст яких не може бути змінений процесором під час виконання програми і зберігається при відсутності живлення

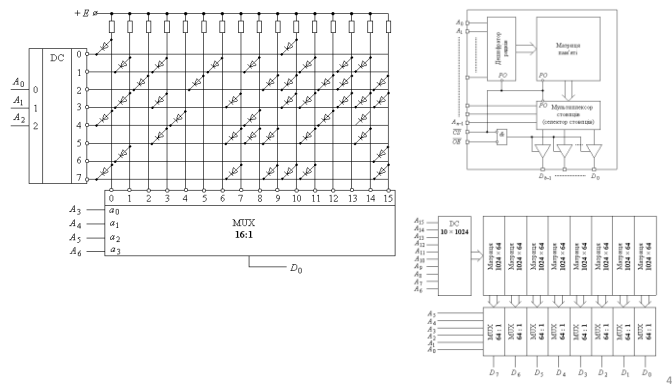


N	Входи			Виходи			
	A ₂	A ₁	A ₀	D ₃	D ₂	D ₁	D ₀
0	0	0	0	1	1	0	1
1	0	0	1	1	0	1	0
2	0	1	0	0	1	1	1
3	0	1	1	1	0	1	1
4	1	0	0	1	1	0	0
5	1	0	1	0	0	1	1
6	1	1	0	0	0	1	0
7	1	1	1	1	0	0	1



3

Двовимірні постійні запам'ятовуючі пристрої



Репрограммовані постійні запам'ятовуючі пристрої

- Репрограммованими називають ПЗП, які є енергонезалежними приладами і інформація в яких може неодноразово перезаписуватись.
 - EPROM (Electrically Programmable ROM) або РПЗП-УФ (репрограмовані ПЗП з ультрафіолетовим стиранням записаної інформації);
 - EEPROM (Electrically Erasable PROM) або РПЗП-ЕС (репрограмовані ПЗП з електричним стиранням записаної інформації);
 - пам'ять типу flash (флеш).
- Паралельні EEPROM
 - одна напруга живлення для всіх операцій;
 - 8-розрядна організація;
 - вбудовані фіксатори адреси та даних;
 - вбудований автомат програмування;
 - побайтове і секторне програмування;
 - типова кількість циклів програмування – не менш 10000;
 - виконання для побутової (діапазон робочих температур 0...+ 70° C) і промислової (– 40...+ 85° C) апаратури.
- Послідовні EEPROM
 - мікросхеми з послідовним протоколом запису та читування даних (мікросхеми з двошвидкісною шиною типу IIC (I²C) та з трьохшвидкісною шиною типу MICROWARE (SPI))

Основи побудови флеш пам'яті

- Стирання записаної в неї інформації забезпечується не окремими словами (байтами), а в цілому для всієї мікросхеми або досить великих її розмірів.
 - параметричні – мікросхеми пам'яті для зберігання даних і програм, які не дуже інтенсивно змінюються (мікросхеми з поблочним стиранням)
 - мікросхеми з великою ємністю ((64...128) М), які використовуються для зберігання великих обсягів інформації, тобто для заміни носіїв на магнітних дисках

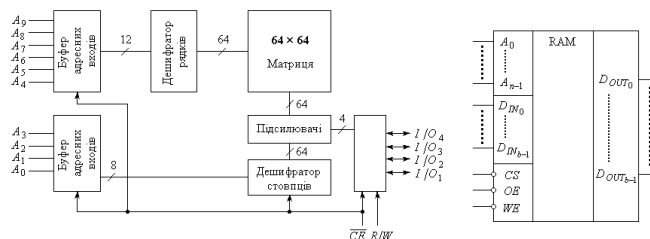
3FFFFH	16K Byte Boot Block
3E000H	
3DFFFH	8K Byte Parameter Block
3D000H	
3CFFFH	8K Byte Parameter Block
3C000H	
3BFFFH	96K Byte Parameter Block
30000H	
2FFFFFH	128K Byte Main Block
20000H	
1FFFFH	128K Byte Main Block
10000H	
0FFFFH	128K Byte Main Block
00000H	

Оперативні запам'ятовуючі пристрої

- Призначенні для швидкого запису, зчитування та тимчасового зберігання обсягів інформації, які використовуються оперативно при виконанні програм.
- *Статичні ОЗП* в якості елементів пам'яті використовують *D-тригери*, інформація в яких при наявності напруги живлення зберігатиметься від запису до перезапису. Тривалість зберігання інформації у статичних ОЗП необмежена.
- *Динамічні ОЗП* в якості елементів пам'яті використовують конденсатори ємністю порядку 0,5 пФ. У динамічних вона обмежується часом саморозряду конденсаторів, що вимагає спеціальних засобів відновлення (регенерації) даних і додаткових витрат часу на цей процес.

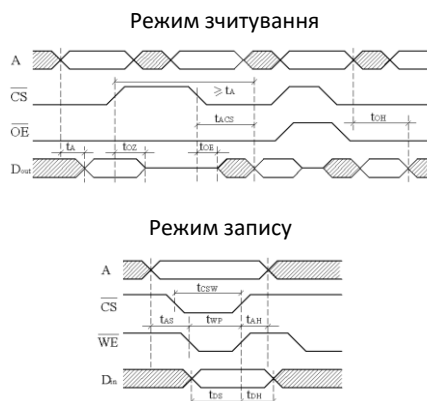
7

Статичні оперативні запам'ятовуючі пристрої



8

Часові діаграми ОЗП



9

Список використаних джерел

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, – 736 с.
2. Уилкинсон Б. Основы проектирования цифровых схем: Пер. с англ. – М.: Изд. дом «Вильямс», 2004. – 320 с.
3. Точки Р., Уидмер Н. Цифровые системы, теория и практика, 8-е изд.: Пер. с англ. – М.: Изд. дом «Вильямс», 2004. – 1024 с.

10

**ЛЕКЦІЮ ЗАКІНЧЕНО.
ДЯКУЮ ЗА УВАГУ!**

11

Лекція №17

Слайд 1

«ЦИФРОВІ ІНФОРМАЦІЙНІ СИСТЕМИ»

Лекція №17. Основи побудови структур простих програмованих логічних матриць. Сучасні ПЛІС. Основні параметри ПЛІС

Слайд 2

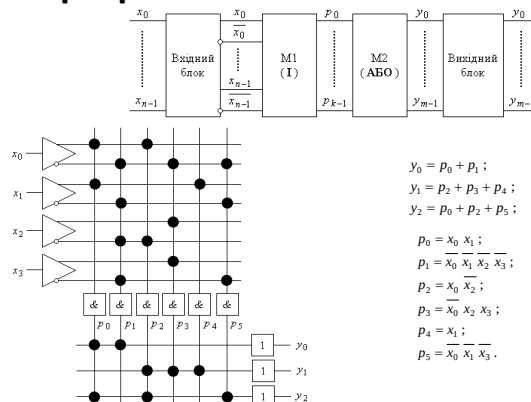
Зміст

- Основи побудови структур простих програмованих логічних матриць
- Сучасні ПЛІС
- Основні параметри ПЛІС

2

Слайд 3

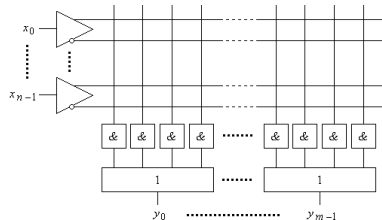
Основи побудови структур простих програмованих логічних матриць



3

Програмована матрична логіка

- Є спрощеним варіантом ПЛМ і характеризується тим, що має програмовану матрицю елементів І та фіксовану матрицю АБО



4

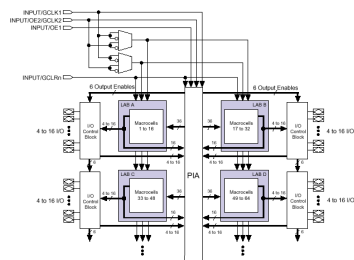
Сучасні ПЛІС

- Програмованість НВІС забезпечується наявністю в ній великої кількості двополюсників (ключів), які за допомогою програмних засобів можуть переводитись користувачем в стан високої або низької (нульової) провідності
- Типи програмованих ключів:
 - перемички типу *antifuse* (тонкі діелектричні пробиті перемички)
 - ЛІЗМОН-транзистори з подвійним затвором
 - ключові польові транзистори, керовані тригерами пам'яті конфігурації

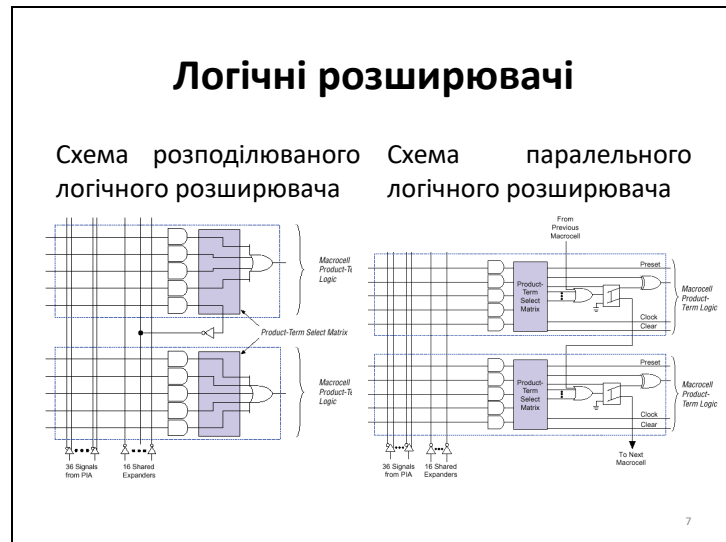
5

Основні параметри ПЛІС

- Критерії:
 - логічна ємність, достатня для реалізації проекту, з можливістю майбутнього оновлення;
 - швидкість ПЛІС;
 - схемотехнічні та конструктивні параметри;
 - наявність або вартість засобів розробки, що включає в себе апаратні та програмні засоби;
 - наявність технічної та методичної підтримки;
 - вартість мікросхеми та надійність каналів їх придбання
- Основні елементи функціональної схеми:
 - логічні блоки (ЛБ) (Logic Array Blocks (LAB))
 - макроелементи МЕ (Macrocells)
 - логічні розширювачі (Expanders): паралельний (Parallel) та розподілений (Shareble)
 - програмована матриця з'єднань (ПІМЗ) (Programmable Interconnected Array (PIA))
 - елементи вводу-виводу (ЕВВ) (I/O Control Block)



6



Список використаних джерел

1. Рябенський В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ – 2000, 2009, - 736 с.
2. Угрюмов Е.П. Цифровая схемотехника / Учебное пособие – СПб.: БХВ, 2001. - 528 с.
3. Уилкинсон Б. Основы проектирования цифровых схем: Пер. с англ. – М.: Изд. дом «Вильямс», 2004. – 320 с.

8

ЛЕКЦІЮ ЗАКІНЧЕНО. ДЯКУЮ ЗА УВАГУ!

9