Міністерство освіти і науки, молоді та спорту України
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ ДО КУРСОВОЇ РОБОТИ З ДИСЦИПЛІНИ “ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ КОМП’ЮТЕРНИХ СИСТЕМ”

ОДЕСА ОНПУ 2012

Міністерство освіти і науки, молоді та спорту України

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ
ДО КУРСОВОЇ РОБОТИ
З ДИСЦИПЛІНИ
“ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ КОМП’ЮТЕРНИХ СИСТЕМ”

для студентів напряму підготовки

6.050102 “Комп’ютерна інженерія”

спеціальність

05010201 “Комп’ютерні системи та мережі”

Затверджено

на засіданні кафедри комп’ютерних

інтелектуальних системи та мережі

Протокол № 1 від 28.08.2012

ОДЕСА ОНПУ 2012

Методичні вказівки до курсової роботи з дисципліни “Технології проектування комп’ютерних систем”, для студентів напряму підготовки 6.050102 “Комп’ютерна інженерія” / Уклад. К.В. Защолкін. – Одеса: ОНПУ, 2012. – 22 с.

Укладач: **К.В.Защолкін**, канд. техн. наук, доцент

ЗМІСТ

ВСТУП

Дисципліна “Технології проектування комп’ютерних систем” є важливою складовою частиною циклу автоматизованого проектування навчального плану напряму “Комп'ютерна інженерія”. Теоретичні положення даної дисципліни закріплюються виконанням курсової роботи. Курсова робота виконується студентами паралельно з вивченням лекційного курсу в 7-у семестрі.

Мета курсової роботи полягає в закріпленні, поглибленні та узагальненні знань, одержаних студентами за час опанування дисципліни ті їх застосування до комплексного вирішення конкретного фахового завдання. Робота носить проектно-дослідницький характер. Об'єктами проектування в рамках роботи є цифрові пристрої, що генерують однорозрядні та багаторозрядні керуючі сигнали різної форми й частоти. Об'єктами дослідження в рамках даної роботи є мегафункції, що входять до складу стандартної бібліотеки САПР Altera Quartus II.

1 ЗАВДАННЯ НА КУРСОВУ РОБОТУ

В межах даної курсової роботи необхідно:

1) розробити цифровий пристрій, що виконує генерацію строба (керуючого імпульсу), який має задану форму та часові характеристики.

2) розробити цифровий пристрій, який виконує генерацію заданої послідовності 8-ми розрядних двійкових слів через задані інтервали часу.

3) виконати дослідження та опис заданої мегафункції з состава стандартної бібліотеки САПР Altera Quartus II і провести її апаратну реалізацію.

2 РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ ПЕРШОГО ЗАВДАННЯ

2.1 Загальна постановка задачі

При розробці цифрових систем часто виникає потреба в генерації керуючих (тактуючих, стробуючих) імпульсів, із заданою формою та часовими характеристиками. Для вирішення подібної задачі можна скористатися готовим апаратним рішенням – інтервальним таймером, виконаним

у вигляді окремої великої інтегральної схеми або реалізованим у складі мікроконтролера. Однак у багатьох випадках застосування такого готового генератора неможливе через особливості технічного завдання або елементної бази, яка використовується. У силу цього виникає необхідність в умінні проектувати й використовувати в апаратних проектах, власні генератори імпульсів, с заданими характеристиками.

У даному завданні потрібно, описати мовою VHDL, а також виконати апаратну реалізацію та налагодження на лабораторному стенді, генератора, що формує сигнал, форма якого задана студентам по варіанту.

Доступним для використання є наступне устаткування, що входить до складу лабораторного стенда Altera DE2:

– кварцовий генератор тактових імпульсів на 27 MHz;

– тумблери (toggle switches);

– світлодіодні індикатори (LEDs).

2.2 Порядок вирішення задачі

2.2.1 Варіант завдання являє собою часову діаграму стробуючого сигналу, який повинен генерувати розроблений пристрій. Приклад такої діграми наведений на рис. 1.

Діаграма показана в шкалі секунд на діапазоні від 0 с до 59 с. Пристрій повинен приймати на свій вхід меандр з частотою 27 MHz і видавати на вихід сигнал, заданої по варіанту форми.

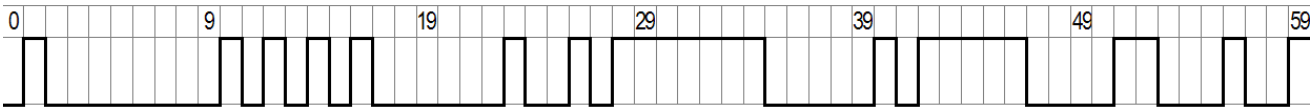

Рис. 1. Приклад часової діаграми строба для першого завдання

2.2.2 Структура пристрою, що генерує потрібний сигнал наведена на рис. 2 у вигляді діаграми потоку даних. На цій діаграмі прямокутниками показані блоки пристрою, які необхідно описати в окремих VHDL-файлах, а овалами – процеси. В верхній частині кожного умовного позначення процесу наведений його список чутливості.

Отже, пристрій, який слід розробити в першому завданні, складається з двох блоків: поділювана частоти та блока формування строба. Розглянемо кожен з цих блоків.

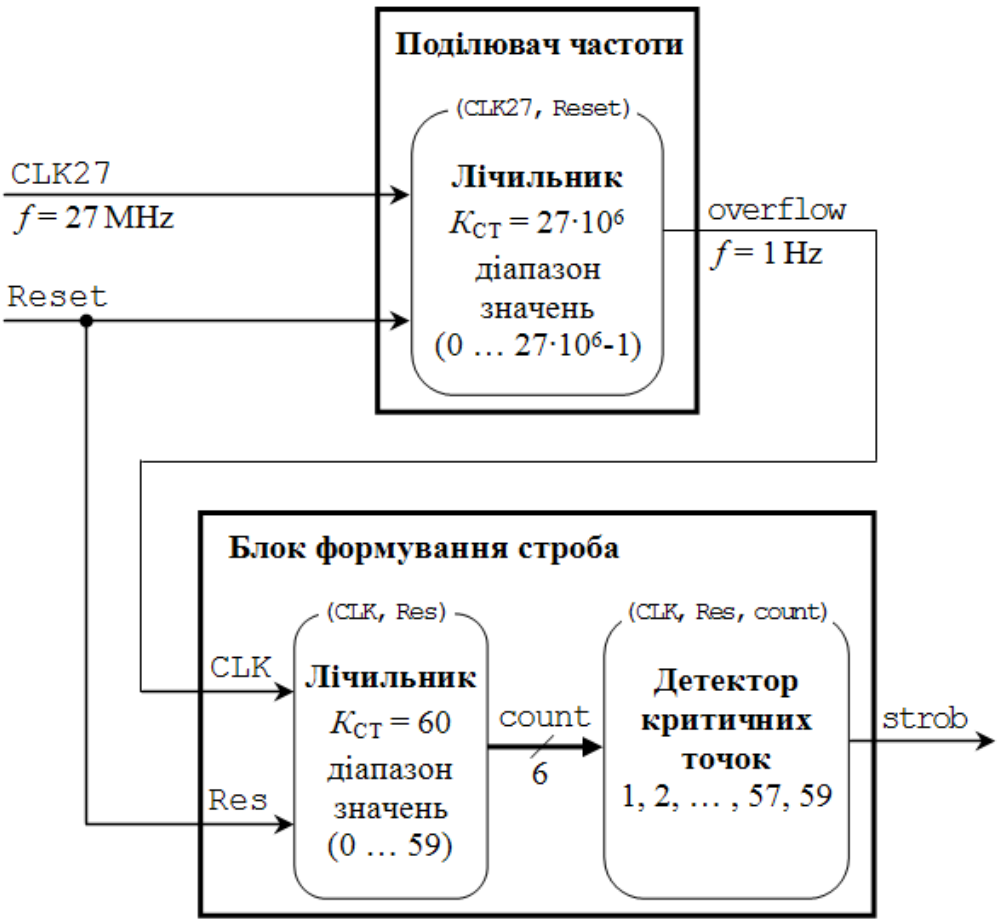

Рис. 2. Діаграма потоку даних генератора строба

2.2.3 ***Подільовач частоти*** це блок, який приймає на свій вхід сигнал, що змінюється з деякою постійною частотою і формує на своєму виході сигнал з меншою постійною частотою. Серед доступного устаткування лабораторного стенда є генератор тактових імпульсів з частотою $f_{27} = 27$ MHz. Для реалізації даного завдання необхідний сигнал з періодом, рівним довжині такту часової діаграми $T = 1$ с (тобто з частотою $f = 1$ Hz). Отже, в даному випадку подільовач частоти повинен приймати на свій вхід меандр частотою f_{27}, а на свій вихід видавати сигнал із частотою $f = 1$ Hz. Таким чином подільовач повинен ділити вхідну частоту на коефіцієнт $27 \cdot 10^6$.

Для реалізації такого подільована рекомендується використати двійковий лічильник з коефіцієнтом рахування $K_{CT} = 27 \cdot 10^6$. Цей лічильник повинен рахувати від нуля до значення $27 \cdot 10^6 – 1$ і в момент переповнення видавати протягом одного такту одиничний сигнал на свій вихід `overflow`. Оскільки за одну секунду такий лічильник виконує 27 мільйонів перемикань і кожні $K_{CT} = 27 \cdot 10^6$ тактів видає одиничний імпульс на вихід `overflow`, то цей імпульс буде формуватися один раз за період ро-

боти лічильника тобто один раз в секунду. Таким чином, передній фронт сигналу переповнення даного лічильника буде виникати з необхідною частотою $f = 1$ Hz.

Зазначений лічильник-поділювач рекомендується описати у вигляді окремого VHDL-файла, який містить один процес. Кількість розрядів лічильника дорівнює $]\log_2 27 \cdot 10^6[= 25$.

2.2.4 ***Блок формування строба*** складається з двох частин (рис. 2): лічильника тактів часової діаграми та детектора критичних точок.

Кількість тактів часової диграми дорівнює 60. Такти пронумеровані від 0 до 59. Для формування строба потрібний ***лічильник тактів*** з коефіцієнтом рахування з коефіцієнтом рахування $K_{CT} = 60$, і законом функціонування (0, 1, 2, … , 58, 59, 0, 1, 2, …). Для подання у двійковій системі, чисел з діапазону 0 – 59 необхідно, щоб лічильник мав 6 розрядів. Зазначений лічильник рекомендується описати у вигляді окремого процесу в VHDL-файлі блока формування строба. Для зберігання стану лічильника, у декларативній частині архітектурної секції цього VHDL-файлу необхідно оголосити 6-ти розрядний вектор для якого пропонується ім'я `count`. Далі показаний шаблон процесу для організації зазначеного лічильника.

```
-- Listing 1
signal count: STD_LOGIC_VECTOR(5 downto 0);

process(CLK27, Reset, count)
begin
     --опис лічильника
end process;
```

На вхід синхронізації (інкрементування) лічильника тактів повинен надходити сигнал з періодом $T = 1$ с (частотою $f = 1$ Hz), який одержується з виходу поділювача частоти.

В той же VHDL файл, у якому описаний лічильник тактів, необхідно помістити окремий процес, що безпосередньо формує, необхідний сигнал стробу. Для цього слід визначити критичні точки сигналу. Критичні точки – це моменти часу в які (такти, починаючи з яких) сигнал міняє своє значення на протилежне. Для часової діаграми наведеної на рис. 1 номера цих тактів дорівнюють: 1, 2, 10, 11, 12, 13, 14, 15, 16, 17, 23, 24, 26, 27, 28, 35, 40, 41, 42, 47, 51, 53, 56, 57, 59.

Список чутливості даного процесу повинен містити синхроімпульс, сигнал скидання та векторний сигнал значення лічильника тактів. Якщо на момент переднього фронту синхроімпульсу лічильник досяг значення чергової критичної точки, сигнал, що буде виданий на вихід у якості строба (`state`) повинен бути проінвертований. Далі показаний шаблон процесу для організації зазначеного детектора критичних точок.

```
01 -- Listing 2
02 process(CLK, Res, count)
03 begin
04    if(Res = '1') then state <= '0';
05    elsif(CLK'event and CLK = '1') then
06        if (count=1 or count=2 or count=10 or
07            count=11 or count=12 or count=13 or
08
09            -- і т.д.
10
11            count=56 or count=57 or count=59)
12        then state <= not state;
13        end if;
14    end if;
15 end process;
```

Коментар до опису *Listing 2*

Рядок **04** – при скиданні пристрою, сигнал строба `state` установлюється в нуль.

Рядок **05** – всі інші події в пристрої будуть відбуватися в момент переднього фронту сигналу на сході `CLK`.

Рядок **06 – 11** – виконується перевірка, чи перебуває стан лічильника `count` у значенні критичної точки.

Рядок **12** – якщо так, то виконується інвертування сигналу `state`.

За межами процесу, але в тілі архітектури, сигнал state необхідно видати на вихідний порт пристрою (відповідно до рис. 2 цей порт носить ім'я `strob`).

Таким чином у проекті, реалізованому в межах першого завдання повинно бути два текстових VHDL-файли. Перший з них складається з одного процесу й описує поділювач частоти. Другий складається з двох проце-

сів: перший процес відраховує такти (кванти) часової діаграми строба, другий детектує критичні точки та безпосередньо формує вихідний сигнал. Крім цього в проекті повинен міститися схемотехнічний файл, який об'єднує два зазначених блоки в загальну систему (відповідно до рис 2).

!!!

В загальному випадку, при проектуванні пристрою, що генерує строб необхідно вибрати величину кванта часу шкали часової діаграми, тобто визначити тривалість такту (T). Значення тривалості такту повинне задовольняти наступним умовам:

– не перевищувати тривалість самого короткого імпульсу часової діаграми (розглядаються і одиничні і нульові імпульси);

– всі імпульси часової діаграми повинні бути кратні по довжині величині T (з максимально можливим ступенем точності).

Чим менше значення T, тим легше виконати зазначені умови.

У прикладі, який був розглянутий вище (рис. 1) та у варіантах першого завдання курсової роботи всі імпульси кратні за довжиною однієї секунді. Тому в даному випадку тривалість такту $T = 1$ с.

2.3 Вибір індивідуального варіанта для першого завдання

Для визначення індивідуального варіанта студент повинен:

1) визначити двійкові, 4-х розрядні еквіваленти кількості букв у своєму прізвищі (**П**), повному імені (**І**) та по батькові (**ПБ**).

2) підставити в табл.. 1 замість символів “AAAA” двійковий еквівалент числа **П**, замість символів “BBBB” двійковий еквівалент числа **І**, замість символів “CCCC” двійковий еквівалент числа **ПБ**. Кожний розряд отриманого 60-ти розрядного двійкового рядка показує рівень строба у відповідному такті.

Розглянемо приклад визначення індивідуального варіанта. Нехай кількість букв у прізвищі студента **П** = 7, кількість букв у повному імені **І** = 10, кількість букв в по батькові **ПБ** = 12.

Знаходимо двійкові 4-х розрядні еквіваленти для зазначених десяткових чисел: $\mathbf{П} = 7_{10} = 0111_2$, $\mathbf{І} = 10_{10} = 1010_2$, $\mathbf{ПБ} = 12_{10} = 1100_2$.

На рис. 3, для даного приклада, показаний рядок, виписаний з табл. 1, крім цього показаний рядок, отриманий в результаті заміни послідовностей “AAAA”, “BBBB”, “CCCC” на двійкові еквіваленти та часова діаграма, яка відповідає цьому двійковому рядку.

Таблиця 1

Такти																																																											
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
A	***A***	***A***	***A***	***B***	***B***	***B***	***B***	***C***	***C***	***C***	***C***	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0	0	1	1	0	0	1	1	***C***	***C***	***C***	***C***	***B***	***B***	***B***	***B***	***A***	***A***	***A***	***A***

Такти																																																											
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
A	***A***	***A***	***A***	***B***	***B***	***B***	***B***	***C***	***C***	***C***	***C***	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0	0	1	1	0	0	1	1	***C***	***C***	***C***	***C***	***B***	***B***	***B***	***B***	***A***	***A***	***A***	***A***
0	***1***	***1***	***1***	***1***	***0***	***1***	***0***	***1***	***1***	***0***	***0***	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0	0	1	1	0	0	1	1	***1***	***1***	***0***	***0***	***1***	***0***	***1***	***0***	***0***	***1***	***1***	***1***

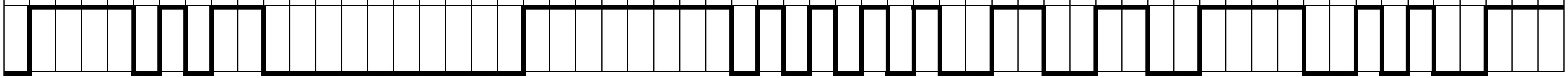

*Рис. 3. Приклад вибору варіанта часової діаграми для першого завдання (**П**=7_{10}=0111_2, **І**=10_{10}=1010_2, **ПБ**=12_{10}=1100_2)*

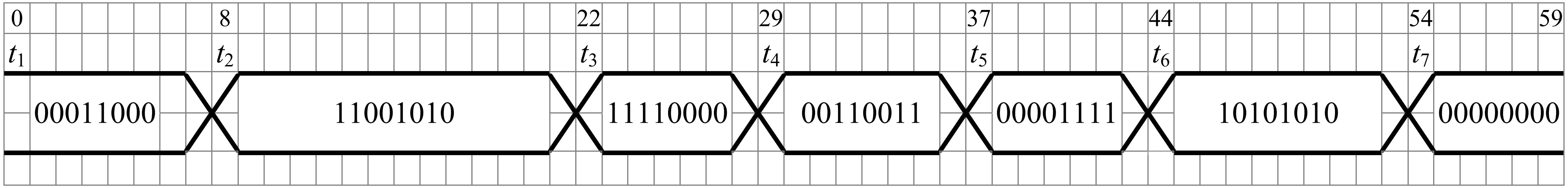

Рис. 4. Приклад часової діаграми для другого завдання

3 РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ ДРУГОГО ЗАВДАННЯ

3.1 Загальна постановка задачі

Поряд з задачею генерації двійкових імпульсів з заданою формою й часовими характеристиками, яка вирішувалася в першому завданні курсової роботи, при проектуванні цифрових систем часто виникає задача генерації послідовностей n-*розрядних* двійкових слів, які генеруються через задані інтервали часу.

У даному завданні необхідно, описати мовою VHDL, а також виконати апаратну реалізацію й налагодження на лабораторному стенді, пристрою, що генерує через зазначені інтервали часу задану за варіантом послідовність 8-ми розрядних двійкових слів. При цьому потрібно, щоб послідовність видавалася ***однократно***, а не циклічно.

3.2 Порядок рішення задачі

3.2.1 Функціонування пристрою, яких необхідно спроектувати задано у вигляді часової діаграми, у шкалі секунд на діапазоні від 0 з до 59 с. Розроблювальний пристрій повинен приймати на свій вхід меандр із частотою 27MHz і видавати на 8-мі розрядний вихідний порт, послідовність двійкових комбінацій.

Структура пристрою, який необхідно розробити в даному завданні подібна до структури генератора строба, реалізованого в першому завданні. Пристрій складається з двох блоків: поділювана частоти та блока формування вихідного сигналу. Для реалізації другого завдання рекомендується виконати модифікацію блока формування вихідного сигналу з попереднього завдання, а саме:

– внести зміни в лічильник тактів з тим, щоб він виконував не циклічне, а однократне рахування;

– внести зміни в інтерфейс блока, добавивши в нього вихідний 8-разрядний порт `out_word`;

– модифікувати процес детектування критичних точок часової діаграми.

3.2.2 Розглянемо рішення завдання на прикладі часової діаграми, представленої на рис. 4. Цю часову диграму слід розуміти наступним чином: в момент часу t_1, тобто на 0-му такті на вихідний порт пристрою `out_word` видається двійкова комбінація «00011000», в момент часу t_2, тобто на 8-му такті вона змінюється на комбінацію «11001010», і т.д.

В той саме VHDL-файл, у якому описаний лічильник тактів із $K_{СТ} = 60$, необхідно помістити окремий процес, що безпосередньо формує,

необхідну за завданням вихідну двійкову послідовність. Для цього, як і у випадку зі стробом, необхідно визначити критичні крапки часової діаграми. Критичні крапки – це моменти часу в які (такти, починаючи з яких) відбувається зміна значень на діаграмі (включаючи найперший такт). Для тимчасової діаграми, наведеної на рис. 4 номера цих тактів дорівнюють: 0, 8, 22, 29, 37, 44, 54.

Список чутливості даного процесу повинен містити синхроімпульс, сигнал скидання та векторний сигнал значення лічильника. По передньому фронту синхроімпульсу при досягнення лічильником значення, яке відповідає черговій критичній крапці, вихідний 8-м розрядний сигнал процесу повинен бути встановлений у нове значення. Далі показаний приклад процесу для організації генератора двійкових послідовностей.

```
01 -- Listing 3
02 process(CLK, Res, count)
03 begin
04  if (Res = '1') then out_word <= "00011000";
05  elsif(CLK'event and CLK = '1') then
06     if (count = 0) then out_word <= "00011000";
07     elsif (count = 8)  then out_word <= "11001010";
08     elsif (count = 22) then out_word <= "11110000";
09     elsif (count = 29) then out_word <= "00110011";
10     elsif (count = 37) then out_word <= "00001111";
11     elsif (count = 44) then out_word <= "10101010";
12     elsif (count = 54) then out_word <= "00000000";
13     end if;
14  end if;
15 end process;
```

Коментар до опису *Listing 3*

Рядок **04** – при скиданні пристрою на вихідний 8-і розрядний порт `out_word` видається перша двійкова комбінація `"00011000"`.

Рядок **05** – всі інші події в пристрої будуть відбуватися в момент переднього фронту сигналу на сході `CLK`.

Рядок **06** – у момент, коли лічильник перебуває в стані 0, тобто коли має місце перша критична крапка, видати на вихід першу комбінацію `"00011000"`.

Рядок **07** – у момент, коли лічильник дійшов до стану 8, тобто коли має місце друга критична крапка, видати на вихід другу комбінацію `"11001010"`.

Рядки **09 – 12** – на вихід видається 3-я – 7-а двійкова комбінацію, при знаходженні лічильника, відповідно в 3-ої – 7-ої критичних крапках.

3.2.3 Відповідно до умов завдання потрібно, щоб послідовність 8-мі розрядних двійкових слів видавалася на вихід пристрою ***однократно***. Для цього необхідно зупинити лічильник тактів після реалізації першого циклу рахування, тобто лічильник повинен "відробити" тільки один раз. З метою забезпечення даної функціональності пропонується ввести в опис пристрою внутрішній сигнал `stop`, при нульовому значенні якого лічильник буде виконувати процес рахування, а при одиничному – буде зупинений. Після початку функціонування системи або після її скидання сигнал `stop` повинен мати нульове значення. Після переповнення лічильника сигнал `stop` повинен одержати одиничне значення й тим самим зупинити лічильник.

Далі показаний процес, який описує такий лічильник з однократним рахуванням.

```
01 -- Listing 4
02 process(CLK, Res, count)
03
04 begin
05   if (Res = '1') then count <= "000000"; stop<='0';
06   elsif (CLK'event and CLK = '1') then
07         if (count = 59) then stop <= '1';
08         elsif (stop = '0') then count <= count + 1;
09         end if;
10   end if;
11 end process;
```

Коментар до опису *Listing 4*

Рядок **05** – при скиданні пристрою, сигнал `stop` також скидається в нуль.

Рядок **07** – якщо на момент переднього фронту сигналу на вхід `CLK` лічильник перебуває в останньому значенні циклу рахування (59), то привласнити сигналу `stop` одиничне значення.

Рядок **08** – інкрементування лічильника відбувається тільки при нульовому значенні сигналу `stop`.

3.3 Вибір індивідуального варіанта для другого завдання

3.3.1 Для формування індивідуального варіанта студент повинен:

1) визначити сім 8-і розрядних двійкових комбінацій;

2) визначити сім критичних крапок – номерів тактів, у яких ці комбінації починають видаватися на вихід.

Індивідуальними даними для формування варіанта є: **П** – кількість букв у прізвищі студента, **І** – кількість букв у повному ім'ї студента, **ПБ** –

кількість букв у по батькові студента, **М** – місяць народження студента, AAAA – двійковий 4-х розрядний еквівалент числа **П**, BBBB – двійковий 4-х розрядний еквівалент числа **І**, CCCC – двійковий 4-х розрядний еквівалент числа **ПБ**, MMMM – двійковий 4-х розрядний еквівалент числа **М.**

Індивідуальний варіант часової діаграми функціонування пристрою визначається за табл. 2 виходячи з наведених даних.

Таблиця 2

Номер такту, починаючи з якого на вихід видається комбінація	**Двійкова комбінація, яка видається на вихід**
$t_1 = 0$	$DATA_1$ = AAAA BBBB
$t_2 = \mathbf{П}$	$DATA_2$ = AAAA CCCC
$t_3 = t_2 + \mathbf{І}$	$DATA_3$ = AAAA MMMM
$t_4 = t_3 + \mathbf{ПБ}$	$DATA_4$ = BBBB CCCC
$t_5 = t_4 + \mathbf{М}$	$DATA_5$ = BBBB MMMM
$t_6 = t_5 + 12$	$DATA_6$ = CCCC MMMM
$t_6 = t_6 + 8$	$DATA_7$ = 1111 1111

3.3.2 Розглянемо приклад формування індивідуального варіанта для другого завдання. Нехай кількість букв у прізвищі студента **П** = 5, кількість букв у повному імені **І** = 11, кількість букв у по батькові **ПБ** = 12, місяць народження студента **М** = 4.

Знаходимо двійкові 4-х розрядні еквіваленти для отриманих, чисел: **П** = $5_{10} = 0101_2$, **І** = $11_{10} = 1011_2$, **ПБ** = $12_{10} = 1100_2$, **М** = $4_{10} = 0100_2$. Приклад варіанту завдання для наведених індивідуальних даних показаний в табл. 3.

Таблиця 3

Номер такту, починаючи з якого на вихід видається комбінація	**Двійкова комбінація, видавана на вихід**
$t_1 = 0$	$DATA_1$ = 0101 1011
$t_2 = \mathbf{П} = 5$	$DATA_2$ = 0101 1100
$t_3 = t_2 + \mathbf{І} = 5 + 11 = 16$	$DATA_3$ = 0101 0100
$t_4 = t_3 + \mathbf{ПБ} = 16 + 12 = 28$	$DATA_4$ = 1011 1100
$t_5 = t_4 + \mathbf{М} = 28 + 4 = 32$	$DATA_5$ = 1011 0100
$t_6 = t_5 + 12 = 32 + 12 = 44$	$DATA_6$ = 1100 0100
$t_7 = t_6 + 8 = 44 + 8 = 52$	$DATA_7$ = 1111 1111

4 РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ ТРЕТЬОГО ЗАВДАННЯ

4.1 Загальна постановка задачі

Третє завдання курсової роботи складається в самостійному вивченні однієї мегафункції зі складу бібліотеки САПР Altera Quartus II. Для виконання завдання необхідно:

1) вивчити документацію для заданої за варіантом мегафункції з бібліотеки Altera Quartus II;

2) виконати докладний опис пристрою, описаного мегафункцією;

3) провести моделювання пристрою, описаного мегафункцією;

4) розробити проект, який включає в себе задану мегафункцію та демонструє основні режими її функціонування. Промоделювати, просинтезувати та налагодити цей проект на лабораторному стенді.

4.2 Вибір індивідуального варіанта для третього завдання

Студенти самостійно на свій розсуд вибирають одну їх мегафункцій, представлених у табл. 4 і повідомляють про свій вибір, викладачеві, який приймає курсову роботу. У третьому стовпці табл. 4 визначена кількість студентів, за якими може бути закріплена дана мегафункція.

Таблиця 4

№	Назва	Короткий опис	Кількість студентів
Арифметичні пристрої з плаваючою крапкою			
1	altfp_abs	Блок одержання абсолютного значення для числа із плаваючою крапкою	2
2	altfp_add_sub	Пристрій, що підсумовує та віднімає числа з плаваючою крапкою	2
3	altfp_compare	Пристрій порівняння чисел із плаваючою крапкою	2
4	altfp_convert	Пристрій перетворення між форматами з плаваючою та фіксованою крапкою	2
5	altfp_div	Пристрій ділення чисел із плаваючою крапкою	2
6	altfp_exp	Обчислювач виразу e^x	2
7	altfp_inv	Обчислювач зворотного значення для числа із плаваючою крапкою	2

Продовження таблиці 4

№	Назва	Короткий опис	Кількість студентів
8	altfp_inv_sqrt	Обчислювач зворотного значення квадратного кореня	2
9	altfp_log	Обчислювач натурального логарифма	2
10	altfp_mult	Помножувач з округленням результату	2
11	altfp_sqrt	Пристрій обчислення квадратного кореня	2
Арифметичні пристрої з фіксованою крапкою			
12	altaccomulate	Акумулятор, що параметризується	1
13	altmemmult	Помножувач на основі пам’яті	1
14	altmultacum	Накопичувальний помножувач	1
15	altmult_add	Група помножувачів з загальним накопичувачем	1
16	altmult_complex	Помножувач комплексних чисел	1
17	devide	Пристрій ділення	1
18	lpm_add_sub	Пристрій, що підсумовує та віднімає	1
19	lpm_compare	Пристрій порівняння	1
20	lpm_mult	Помножувач	1
21	parallel_add	Багатооперанднрй суматор	1
Пристрої уведення-виводу			
22	altddio_bidir	Приймач-передавач даних за обома фронтами синхросигнала (DDR – Double Data Rate)	2
23	altdqs	Пристроїв формування сигналів керування DDR-Пам'яттю	2
24	altlvds_rx	Приймач сигналів, представлених у стандарті LVDS	2
25	altlvds_tx	Передавач сигналів, представлених у стандарті LVDS	2
26	std_virtual_jtag	Пристрій, що забезпечує доступ до ресурсів FPGA, через JTAG-Інтерфейс	2
Запам'ятовуючі пристрої			
27	alt3pram	Трипортова оперативна пам’ять	1
28	altdpram	Двохпортова (чотирьохпортова) оперативна пам’ять	2

Продовження таблиці 4

№	Назва	Короткий опис	Кількість студентів
29	altmeminit	Пристрій ініціалізації пам'яті	1
30	altshift_taps	Сегментований зрушувальний регістр	1
31	dcfifo	Запам'ятовувальний пристрій типу FIFO із двома синхронізуючими входами	2
32	lpm_fifo	Запам'ятовувальний пристрій типу FIFO з одним синхронізуючим входом	2

4.3 Порядок виконання третього завдання

1) Коротка довідка про мегафункції може бути отримана через меню майстра мегафункції "**Documentation → Megafunction online Help**", повна документація може бути завантажена з Web-Вузла компанії Altera за допомогою меню "**Documentation → On the Web**" (рис. 5). Для деяких мегафункцій є можливість згенерувати зображення часової діаграми, що пояснює основні режими роботи пристрою. Генерація цього зображення виконується шляхом вибору пункту меню "**Documentation → Generate Sample Waveforms**". Зображення часової діаграми розташовується у вигляді jpeg-файлу в папці поточного проекту.

Студентові необхідно вивчити коротку довідку, повну документацію та часову діаграму мегафункції. Після цього виконати опис мегафункції, а саме, описати:

– призначення мегафункції;

– порти пристрою (назва, напрямок передачі даних, призначення, обов'язковість наявності порту);

– параметри мегафункції;

– принципи функціонування пристрою (з прикладами).

2) Після з'ясування принципів функціонування мегафункції, необхідно в схемотехнічному файлі додати до неї виводи (pins)

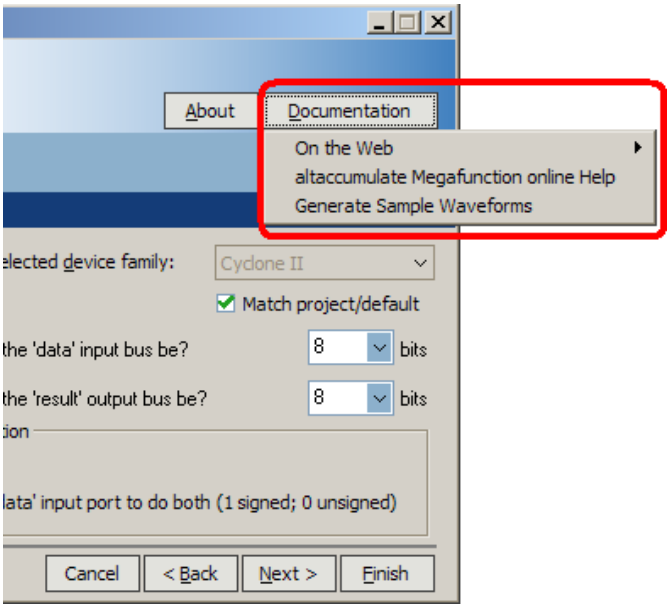

Рис. 5. Одержання інформації про мегафункцію

і виконати моделювання отриманого проекту. У пояснювальній записці необхідно привести часові діаграми основних режимів роботи пристрою й коротко прокоментувати їх.

3) Після цього слід виконати модифікацію проекту з метою демонстрації на лабораторному стенді можливостей мегафункції.

4.4 Приклад виконання опису мегафункції

Розглянемо приклад виконання опису мегафункції `altecc_encoder`. Для виконання цього опису знадилося звернутися до короткої довідки про мегафункцію, а також завантажити і ознайомитися з документацією до неї.

Призначення мегафункції. Мегафункція, дозволяє одержати пристрій, що виконує завадостійке (ECC – Error Control Code) кодування даних.

Порти пристрою, що описує мегафункція. Інформація про порти пристрою представлена в табл. 5.

Таблиця 5

Назва порту	Обов'язковість порту	Напрямок	Призначення порту
`data[]`	Так	Вхідний	Вхідний порт даних. На вхід у паралельному коді (всі розряди одночасно) подається двійковий пакет, який підлягає кодуванню.
`clock`	Ні	Вхідний	Сигнал синхронізації процесу виконання кодування. Наявність даного порту вимагає установки значення (більшого за нуль) кількості тактів, після якої результат видається на вихід.
`clocken`	Ні	Вхідний	Сигнал дозволу синхронізації.
`aclr`	Ні	Вхідний	Асинхронне скидання пристрою.
`q[]`	Так	Вихідний	Вихідний порт, через який паралельно (всі розряди одночасно) видається закодований двійковий пакет даних.

Параметри мегафункції. Інформація про параметри мегафункції наведена в табл. 6.

Таблиця 6

Параметр	Обов'язковість параметра	Опис параметра
Розмірність вхідного порту	Так	Визначає розмірність вхідного порту в діапазоні від 2 до 64 розрядів
Розмірність вихідного порту	Так	Визначає розмірність вихідного порту в діапазоні від 6 до 72 розрядів
Кількість тактів синхронізації LPM_PIPELINE	Ні	Визначає наявність регістрів у схему пристрою. Параметр являє собою ціле число більше нуля. При значенні 1 на виході пристрою має місце регістр; при значенні 2 регістр має місце і на виході і на вході пристрою. При значенні, що перевищує 2 на вихід додаються додаткові регістри.

Принцип функціонування пристрою, що задається мегафункцією. Пристрій приймає на свій вхід n-*розрядний* двійковий пакет, виконує кодування й видає на вихід двійковий пакет розмірністю $n+k+1$. Кодування виконується відповідно до модифікованого методу Хеммінга й полягає в додаванні до вхідного пакета $k+1$ контрольних розрядів. Наявність у пакеті, контрольних розрядів дозволяє при декодуванні виявити двократну та виправити однократну помилку.

Залежність кількості контрольних розрядів від довжини вхідного пакета виражається формулою:

$$k =]\log_2(n + 1 +]\log_2(n+1)[)[$$

де,]·[– операція округлення до найближчого більшого цілого.

У вихідному пакеті k контрольних розрядів розташовуються на позиціях, номера яких дорівнюють цілої степені числа два, тобто на позиціях: 1, 2, 4, 8, ... Ще один контрольний розряд розташовується наприкінці пакета. На інших позиціях розміщаються розряди вхідного пакета. Наприклад, якщо розмірність вхідного пакета $n = 5$, то в результаті кодування до цього пакета буде доданий $k+1 = 4+1 = 5$ контрольних розрядів (K_1, K_2, K_3, K_4, K_5), які будуть розміщені відповідно до рис. 6.

a_1	a_2	a_3	a_4	a_5
D_1	D_2	D_3	D_4	D_5

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
K_1	K_2	D_1	K_3	D_2	D_3	D_4	K_4	D_5	K_5

Рис. 6. Вхідний і вихідний пакети (n = 5, k = 4)

Контрольні розряди обчислюються за виразами, отриманими з таблиці кодування. Приклад такої таблиці для випадку (n = 5, k = 4) наведений в табл. 7. Кількість стовпців основної частини даної таблиці дорівнює k. Кожний стовпець відповідає певному контрольному розряду. Кількість рядків основної частини дорівнює $n+k$. В основній частині таблиці записуються двійкові еквіваленти номерів розрядів кожного рядка.

Таблиця 7

K_4	K_3	K_2	K_1	
0	0	0	1	a_1
0	0	1	0	a_2
0	0	1	1	a_3
0	1	0	0	a_4
0	1	0	1	a_5
0	1	1	0	a_6
0	1	1	1	a_7
1	0	0	0	a_8
1	0	0	1	a_9

Контрольний розряд K_i обчислюється як сума за модулем два, розрядів, проти яких міститься одиниця в стовпці K_i таблиці кодування (найперший з таких розрядів у підсумовуванні не бере участь). Наприклад для обчислення контрольного розряду K_1 необхідно за табл. 7 визначити номера розрядів проти яких міститься одиниця в стовпці K_1. Ці номери 1, 3, 5, 7, 9. Перший із цих розрядів в обчисленні не бере участь, інші підсумуються за модулем два. У результаті вираз для визначення першого контрольного розряду має наступний вигляд:

$$K_1 = a_3 \oplus a_5 \oplus a_7 \oplus a_9.$$

Аналогічно обчислюються інші контрольні розряди:

$$K_2 = a_3 \oplus a_6 \oplus a_7;$$

$$K_3 = a_5 \oplus a_6 \oplus a_7;$$

$$K_4 = a_9.$$

Останній контрольний розряд обчислюється як сума за модулем два, всіх розрядів отриманої кодової комбінації.

Приклад. Якщо на вході пристрою має місце 5-ти розрядний пакет «01101», то контрольні та інформаційні розряди будуть розміщені в такий спосіб (рис. 7).

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
K_1	K_2	0	K_3	1	1	0	K_4	1	K_5

Рис. 7. Приклад вихідного пакета (n = 5, k = 4)

Вирази для обчислення контрольних розрядів у цьому випадку будуть мати такий вигляд:

$K_1 = a_3 \oplus a_5 \oplus a_7 \oplus a_9 = 0 \oplus 1 \oplus 0 \oplus 1 = 0;$

$K_2 = a_3 \oplus a_6 \oplus a_7 = 0 \oplus 1 \oplus 0 = 1;$

$K_3 = a_5 \oplus a_6 \oplus a_7 = 1 \oplus 1 \oplus 0 = 0;$

$K_4 = a_9 = 1;$

$K_5 = a_1 \oplus a_2 \oplus \ldots \oplus a_9 = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 1.$

У підсумку на виході пристрою буде отриманий 10-ти розрядний закодований пакет «0100110111».

5 ЗМІСТ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Пояснювальна записка до курсової роботи оформляється відповідно до вимог ДСТУ для конструкторської документації й повинна містити наступні структурні елементи:

– титульний аркуш встановленого зразка;

– анотацію, у якій коротко описується суть виконаної роботи;

– зміст;

– опис вибору індивідуальних варіантів завдань;

– опис рішення кожного завдання курсової роботи;

– висновок про виконану роботу;

– список використаної літератури (у випадку наявності посилань на літературні джерела);

– додаток, що містить код опису розроблених пристроїв мовою VHDL.

Обсяг пояснювальної записки без урахування додатка: 20-25 аркушів формату А4.

СПИСОК ЛІТЕРАТУРИ

1. Кондратенко Ю.П., Сидоренко С.А., Підопригора Д.М. Поведінковий синтез цифрових пристроїв в середовищі Active-HDL. – Миколаїв: вид-во МФ НаУКМА, 2002. – 116 с.

2. Соловьев В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем. – М.: Горячая линия – Телеком, 2001. – 636 с.

3. Суворова Е.А., Шейнин Ю.Е. Проектирование цифровых систем на VHDL. – СПб.: БХВ, 2003. –576 с.

4. Точчи Р., Уидмер Н. Цифровые системы. Теория и практика, 8-е издание. – М.: Вильямс, 2004. – 1024 с.

5. Бибило П.Н. Системы проектирования интегральных схем на основе языка VHDL: StateCAD, ModelSim, Leonardo Spectrum. – М.: Салон-Р, 2005. – 384 с.

6. Бибило П.Н. Синтез логических схем с использованием языка VHDL. – М.:, Салон-Р, 2002. – 384 с.

7. Перельройзен Е.З. Проектируем на VHDL. – М.: Солон–Пресс, 2004. – 448 с.

8. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах с программируемой структурой. – 2-е издание. – СПб.: БХВ, 2006. – 736 с.

9. Угрюмов Е.П. Цифровая схемотехника. – 2-е издание. – СПб.: БХВ, 2004. – 800 с.

10. Стешенко В.Б. ПЛИС фирмы Altera: элементная база, системы проектирования и языки описания аппаратуры. –М.: Додека-XXI, 2002. – 576 с.

Навчальне видання

Методичні вказівки
до курсової роботи
з дисципліни
“Технології проектування комп’ютерних систем”
для студентів напряму підготовки
6.050102 “Комп’ютерна інженерія”

Укладач: Костянтин Вячеславович Защолкін

Редактори С.М.Шушпановська, Т.І.Лучньова
Коректор Н.К.Филиппович

Підписано до друку	Формат 60x84/16.	Папір газетний.
Друк офсетний.	ум. друк. арк.	обл.-вид. арк.
Тираж 100 пр.	Зам. №	

Одеський національний політехнічний університет
65044, Одеса, пр. Шевченка, 1