

Зміст

<u>Вступ</u>	2
<u>Мета і завдання комплексної контрольної роботи</u>	2
<u>Рекомендації до виконання роботи</u>	2
<u>Навчально-методичні матеріали</u>	6

Вступ

Аналіз є невід'ємною складовою частиною процесу проектування схем, який сьогодні виконується в середовищі автоматизованих систем. Сучасні системи проектування базуються на стандартних мовах опису і моделювання цифрової апаратури таких як VHDL, Verilog. Проведення адекватної верифікації проекту неможливе без розуміння переваг і обмежень використовуваних моделей і методів моделювання, вміння вибирати моделі відповідно до задач, які ставляться на окремих етапах проектування.

Мета і завдання комплексної контрольної роботи

Комплексні контрольні завдання з дисципліни охоплюють розділи: 2 «Моделі структурно – логічного рівня», 3 «Моделі функціонального і регістрового рівнів», 4 «Часова модель та алгоритм часового моделювання ЦС» [1-3]. Метою роботи є перевірка засвоєння студентами принципів побудови структурних і алгоритмічних, статичних і динамічних моделей, алгоритму подієвого моделювання стосовно їх реалізації у мові VHDL. Без розуміння роботи алгоритму правильний опис моделей цифрових пристроїв і проведення їх аналізу за допомогою стандартних мов VHDL або Verilog є не можливим.

Загальне завдання на контрольну роботу:

- 1) Вказати кількість паралельних процесів, списки чутливих входів для кожного процесу і списки розгалужень (зв'язків) для кожного сигналу. Якщо списки в процесі моделювання можуть мінятися, то вказати змінні і процеси, для яких на даному відрізку модельного часу це відбувається.
- 2) Описати хід виконання програми відповідно до алгоритму подієвого моделювання: зміна модельного часу T_m , постановка і витягання з черги (Списку Подій) подій вказаних типів, виконання обчислень і змін значень сигналів.

Кожен студент отримує індивідуальний варіант моделі схеми, описаний мовою VHDL. У варіантах завдань пропонуються структурно-вентильні або алгоритмічні моделі схем.

Рекомендації до виконання роботи

Підготовка до контрольної роботи складається з вивчення теоретичного матеріалу з розділів 2-4 і аналізу прикладів, що надаються.

Рекомендація: якщо важко описувати роботу програми відразу, то уявіть собі схему, яку вона описує, намалюйте уручну часову діаграму на даному відрізку часу, а потім розпишіть виконання по операторах і подіях.

Розглянемо приклад роботи подієвого алгоритму при моделюванні **структурно-вентильної моделі схеми**.

Поточні значення сигналів:

$x1=x2=x3=x4=y3=y4=1$; $y1=y2=0$

Події в СП:

$T_c=15ns$ $x3=0$.

```
entity SX is
port (x1, x2, x3, x4: in std_logic;
      y4: out std_logic);
end SX;
architecture ASX of SX is
signal y1, y2, y3: std_logic;
begin
    E1:    y1<=x1 nand x2;
    E2:    y2<=x3 nand x4;
    E3:    y3<=y1 nor y2;
    E4:    y4<=y2 or y3;
end ASX;
```

У моделі 4 паралельні процеси *E1-E4*.

Списки чутливих входів процесів у процесі моделювання не змінюються:

E1: x1,x2;

E2: x3,x4;

E3: y1,y2;

E4: y3,y2;

Списки розгалужень для сигналів:

x1: E1;

x2: E1;

x3: E2;

x4: E2;

y1: E3;

y2: E3,E4;

y3: E4;

y4: немає.

Виконання програми: $T_c=15\text{ns}$ $x_3=0$

- 1) $T_m=15\text{ns}$. З СП витягується подія - зміна для x_3 , і сигнал змінюється: $x_3=0$. За списком розгалужень сигналу x_3 в СП ставиться подія-обчислення для процесу E2 у час T_m .
- 2) З СП витягується подія-обчислення. Виконуються оператор E2. В результаті обчислення сигнал y_2 змінюється. Процес переходить в режим очікування. У T_m записуються в СП подія-зміна для нового значення $y_2=1$.
- 3) З СП витягується подія і змінній привласнюється нове значення $y_2=1$. За списком розгалужень сигналу y_2 в СП ставляться події-обчислення для процесів E3 і E4 у час T_m .
- 4) З СП витягуються події-обчислення. Виконується оператор E3, а потім E4. В результаті обчислення сигнал y_3 змінюється, а y_4 - ні. Процеси переходять в режим очікування. У T_m записуються в СП подія-зміна для нового значення $y_3=0$.
- 5) З СП витягується подія і змінній привласнюється нове значення $y_3=0$. За списком розгалужень сигналу y_3 в СП ставиться подія-обчислення для процесу E4 у час T_m .
- 6) З СП витягуються події-обчислення. Виконується оператор E4. В результаті обчислення сигнал y_4 не змінюється. Процес переходить в режим очікування. У СП нові події не ставляться.
- 7) Оскільки у СП подій більше немає, процес моделювання закінчується.

Оскільки для даної моделі затримки зміни сигналів дорівнюють 0, для одного і того ж моменту часу $T_m=15\text{ns}$ відбулося декілька ітерацій постановки і витягання подій з СП. Слід звернути увагу на те, що спочатку відбувається витягання з черги подій – причин, а потім постановка у чергу подій – наслідків, які будуть оброблятися на наступній ітерації того ж часу. Таким чином, причини і наслідки між собою не змішуються.

Розглянемо приклад роботи подієвого алгоритму при моделюванні *алгоритмічної моделі схеми*. Це D– тригер, який управляється рівнем 0:

$D=0$; $C=0$; $Q=0$; $nQ=1$

Події в СП:

$T_c=10\text{ns}$ $D=1$.

```
entity D_tr is
port (D, C : in bit; Q, nQ : inout bit)
end D_tr;
architecture D_tr of D_tr is
begin
  (1) E1:process (C, D)
  (2) begin
```

```

(3) if (C='0') then  $Q \leq D$  after(4ns);  $nQ \leq \text{not } (D)$  after(5ns);
(4) end if;
(5) end process;
end D_tr;

```

У моделі 1 паралельний процес.

Список чутливих входів процесу *E1*: *C*, *D*.

Списки розгалужень для сигналів:

C: *E1*:точка входу (1);

D: *E1*:точка входу (1);

Q: немає;

nQ: немає.

Під час моделювання при виклику процесу *E1* список його чутливих входів буде періодично видалятися як і точка входу у цей процес буде видалятися із списків розгалужень сигналів *C* і *D*, а після виконання операторів 2-5 і переходу процесу у режим очікування на операторі 1 списки знов будуть створюватись такими ж самими.

Виконання програми: $T_c=10\text{ns}$ $D=1$

- 1) З СП вибирається нове значення $T_m=10\text{ns}$. Для T_m з СП витягується подія - зміна для *D*, і *D* змінюється. За списком розгалужень сигналу *D* в СП ставиться подія-обчислення для процесу *E1* (точка входу (1)).
- 2) З СП витягується подія-обчислення. Список чутливих входів процесу *E1* очищується, зі списків розгалужень сигналів *C* і *D* точка входу у процес *E1*(1) видаляються. Виконуються оператори (2–5), потім знову оператор (1). Процес переходить у режим очікування. Список чутливих входів процесу відновлюється, точка входу у процес повертається до списків розгалужень сигналів *C* і *D* (тепер при їх зміні процес знов буде викликатися). В СП записуються події – зміни для нових значень: $Q=1$ для $T_m=10\text{ns}+4\text{ns}=14\text{ns}$ та $nQ=0$ для $T_m=10\text{ns}+5\text{ns}=15\text{ns}$. Більше для часу $T_m=10\text{ns}$ подій в СП немає.
- 3) Наступний час у СП $T_m=14\text{ns}$. З СП витягується подія – зміна і привласнюється нове значення сигналу $Q=1$. Оскільки його список розгалужень є пустими, нові події у СП не ставляться. Більше для часу $T_m=14\text{ns}$ подій в СП немає.
- 4) $T_m=15\text{ns}$. З СП витягується подія – зміна і привласнюється нове значення сигналу $nQ=0$. Оскільки його список розгалужень є пустими, нові події у СП не ставляться. Більше для часу $T_m=15\text{ns}$ подій в СП немає.

Якщо у СП подій більше немає, програма припиняє свою роботу, інакше процес чекатиме нових подій для *C* або *D* в моменти часу $> 15\text{ns}$.

Слід ще раз звернути увагу на те, що виклик процесу на виконання відбувається у той самий момент часу, коли змінюється один або більше сигналів з його списку чутливих входів, а зміна вихідних сигналів планується у СП на новий час, який дорівнює поточному часу плюс затримка вихідного сигналу.

Навчально-методичні матеріали

Основна література

1. Семенец В.В, Хаханова В.І., Хаханов В.І. Проектування цифрових систем з використанням мови VHDL. Навчальний посібник. – Харків: ХНУРЕ, 2003. - 492 с.
2. Бибило П.Н. основы языка VHDL. – М.: Солон, 2000. - 208с
3. Савельев А.М. VHDL для проектирования вычислительных устройств. - К.: ЧП «Корнейчук», ООО «ТИД «ДС», 2003. - 208с.

Методичні матеріали

6. Конспект лекцій до курсу «Аналіз цифрових схем»,
що знаходиться на файловому сервері кафедри системного проектування у
розділі Dragon2\Techmat\georgizova\analiz_SX