Содержание

1. Анализ поставленной задачи

2. Разработка структурной схемы устройства

3. Разработка и исследование модели модулей

3.1 Разработка интерфейса модулей

3.2 Разработка архитектуры модулей

3.3 Разработка архитектуры счетчика

3.4 Исследование работы моделей модулей

4 Исследование структурной модели устройства в целом с использованием моделей узлов

4.1 Анализ пригодности разработанных моделей для имплементации в PLD

Заключение

## 1. Анализ поставленной задачи

По сути необходимо разработать устройство, преобразующее аналоговый сигнал в, эквивалентный ему, цифровой код. Т.е. замены сигнала серией импульсов за некоторое определенное время.

Таким образом устройство должно сравнивать входной сигнал с нарастающим эталонным сигналом и за время, пока сигналы не стали равны, посчитать количество импульсов тактового генератора. Частота генератора должна быть в 2 раза меньше, чем скорость нарастания эталонного сигнала на 1 значение.Т. е., если скорость нарастания эталонного сигнала 1 В/10нс, то частота генератора должна быть 10/2=5нс, чтобы генератор выдал 1 импульс эквивалентный 1 В.

В результате выполненных расчетов и сделанных ранее выводов, мы обладаем достаточной информацией, чтобы приступить к разработке структурной схемы устройства.

## 2. Разработка структурной схемы устройства

На данном этапе проектирования разрабатывается структурная схема устройства, опираясь на выводы сделанные при анализе задачи.

Определение набора модулей, входящих в состав схемы. Прежде всего, необходим узел, выполняющий функцию вычитания напряжений. Таковым узлом является дифференциальный усилитель (ДУ). Как уже упоминалось в разделе 1, нам будут необходимы счетчик CЧ, и генератор тактовых импульсов ГТИ. Также необходим узел вырабатывающий напряжение для сравнения его с напряжением разности.

Этим узлом является генератор линейно изменяющегося напряжения (ГЛИН). Для того чтобы счетчик СЧ начинал каждый период отсчета с нуля, необходима схема обнуления счетчика (СО). В состав схемы должны три входить компаратора напряжения. Один (К1) является компаратором знака напряжения разности. Два других (К2, К3) сравнивают напряжение разности с нарастающим или падающим напряжением, подаваемым с ГЛИН. Структурная схема цифрового измерителя разности двух напряжений приведена на рисунке 2.1

Рисунок 2.1 - Структурная схема цифрового измерителя разности двух напряжений

Схема работает следующим образом. С выхода ДУ снимается напряжение равное разности входных напряжений Ua и Ub. Компаратор К1 сравнивает полученный сигнал с нулем. Если напряжение положительное, то на выходе К1 формируется уровень логической 1, если отрицательное - логического 0.

Для удобства индикации сигнала **Знак,** на выходе К1 стоит инвертор. Напряжение разности поступает на входы компараторов К2 и К3. Если напряжение положительное, то срабатывает компаратор К2, и начинает сравнивать его с нарастающим напряжением ГЛИН. Пока напряжение разности больше напряжения ГЛИН, на выходе К2 -1.Т.о. разрешается прохождение импульсов тактового генератора ГТИ, через вентиль, на счетный вход счетчика СЧ. Как только напряжение ГЛИН превысило напряжение разности, на выходе К2 -0, логический вентиль закрывается и счетчик перестает считать. На выходе счетчика сформирован цифровой код, эквивалентный напряжению разности. В момент перехода К2 в 0, срабатывает схема обнуления (СО), которая через некоторое время (необходимое для устойчивой индикации и верного считывания кода) выдаст сигнал на обнуление счетчика.

Если напряжение разности отрицательное, то срабатывает компаратор К3, сравнивающий его с падающим напряжением ГЛИН, и работа схемы повторяется.

## 3. Разработка и исследование модели модулей

## 3.1 Разработка интерфейса модулей

Разработка интерфейса дифференциального усилителя. Дифференциальный усилитель выполняет вычитание двух напряжений, поданных на его входы, и выдает на выход их разницу.

library IEEE;

use IEEE. STD\_LOGIC\_1164. all;

entity dif\_oy is

port (In1: in integer range - 5 to 5;

In2: in integer range - 5 to 5;

Out1: out integer range - 10 to 10);

end dif\_oy;

Разработка интерфейса ГЛИН.

Генератор линейно изменяющегося напряжения имеет два выхода: прямой (напряжение нарастает от 0 до10 В) и инверсный (напряжение падает от 0 до - 10В)

library IEEE;

use IEEE. STD\_LOGIC\_1164. all;

entity GLIN is

port (out1: out integer range 0 to 10;

not\_out1: out integer range - 10 to 0);

end entity GLIN;

Разработка интерфейса компаратора.

Компаратор имеет два входа типа integer и один цифровой выход типа std\_logic.

library IEEE;

use IEEE. STD\_LOGIC\_1164. all;

entity comp is

port (U1: in integer range - 10 to 10;

U2: in integer range - 10 to 10;

U\_out: out std\_logic);

end comp;

Разработка интерфейса генератора тактовых импульсов.

Генератор вырабатывает импульсы и имеет только один выход.

entity gen is

port (gen\_out: out std\_logic);

end gen;

Разработка интерфейса счетчика.

Счетчик должен иметь вход сброса, вход счета и 4 выхода.

library IEEE;

use IEEE. STD\_LOGIC\_1164. all, IEEE. Numeric\_STD. all;

entity counter is

port (reset: in std\_logic;

up: in std\_logic;

C\_out: out unsigned (3 downto 0));

end counter;

## 3.2 Разработка архитектуры модулей

Разработка архитектуры дифференциального усилителя.

Как было сказано ранее, дифференциальный усилитель (ДУ) выполняет функцию вычитания над напряжениями, поданными на его входы. Так как физически операционный усилитель, на базе которого построен ДУ, не может выдать на выходе напряжение больше, чем +U питания (+12 В), то входные напряжения не должны быть больше +5 В.

architecture A4 of dif\_oy is

begin

p1: process (In1, In2)

begin

if ( (In1-In2) <12 and (In1-In2) >-12) then

Out1<=In1-In2;

else Out1<=0;

end if;

end process;

end A4;

Разработка архитектуры ГЛИН.

Генератор линейно изменяющегося напряжения имеет два выхода: прямой (напряжение нарастает от 0 до10 В) и инверсный (напряжение падает от 0 до - 10В)

Скорость нарастания (падения) напряжения на выходах: 1В/100нс.

architecture A1 of GLIN is

begin

p1: process

begin

out1<=1;

wait for 100ns;

out1<=2;

wait for 100ns;

out1<=3;

wait for 100ns;

out1<=4;

wait for 100ns;

out1<=5;

wait for 100ns;

out1<=6;

wait for 100ns;

out1<=7;

wait for 100ns;

out1<=8;

wait for 100ns;

out1<=9;

wait for 100ns;

out1<=10;

wait for 100ns;

end process;

p2: process

begin

not\_out1<=-1;

wait for 100ns;

not\_out1<=-2;

wait for 100ns;

not\_out1<=-3;

wait for 100ns;

not\_out1<=-4;

wait for 100ns;

not\_out1<=-5;

wait for 100ns;

not\_out1<=-6;

wait for 100ns;

not\_out1<=-7;

wait for 100ns;

not\_out1<=-8;

wait for 100ns;

not\_out1<=-9;

wait for 100ns;

not\_out1<=-10;

wait for 100ns;

end process;

end A1;

Разработка архитектуры компаратора.

Компаратор сравнивает два сигнала, которые подаются на его входы. Если сигнал на первом входе больше чем на втором, то на выходе компаратора 1, если меньше - то 0.

architecture A2 of comp is

begin

p1: process (U1,U2)

begin

if (U1-U2) >0 then

U\_out<='1';

else U\_out<='0';

end if;

end process;

end A2;

Разработка архитектуры генератора тактовых импульсов

Генератор тактовых импульсов выдает импульсы длительностью 50нс, чтобы за время нарастания напряжения ГЛИН на 1В, выдать один импульс.

architecture A3 of gen is

begin

p1: process

begin

gen\_out<='0';

wait for 50ns;

gen\_out<='1';

wait for 50ns;

end process;

end A3;

## 3.3 Разработка архитектуры счетчика

Так как структура счетчика не особенно важна, при моделировании был использован поведенческий стиль моделирования. Счетчик имеет вход счета +1, асинхронный вход сброса и 4 выхода.

architecture A5 of counter is

begin

p1: process (reset, up)

variable C\_out1: unsigned (3 downto 0);

begin

if reset='0' then

C\_out1: = "0000";

else

if up'event and up='1'then

C\_out1: =C\_out1+1;

else C\_out1: =C\_out1;

end if;

end if;

C\_out<=C\_out1 after 3ns;

end process;

end A5;

## 3.4 Исследование работы моделей модулей

Исследование работоспособности моделей проводилось с помощью анализа временных диаграмм сигналов. Задавая входное возмущение, производится анализ соответствующих ему выходных сигналов. По таблице соответствий выносится заключение о работоспособности модели. Ниже приведены временные диаграммы, полученные по каждому из модулей.

В идеальном случае при тестировании модели необходимо подать все возможные входные комбинации и проанализировать выходные возмущения. Однако не всегда это является возможным и оправданным, на практике выбирается множество тестовых комбинаций описывающих основные свойства разрабатываемой модели.

Рисунок 3.1 - Временная диаграмма, полученная при анализе модели ДУ.

Из временной диаграммы видно, что дифференциальный усилитель выполняет вычитание In1-In2 и выдает на выход Out1 получившуюся разность.

Рисунок 3.2 - Временная диаграмма, полученная при анализе модели ГЛИН.

Генератор линейно изменяющегося напряжения выдает на выход out1, нарастающее со скоростью 100нс, напряжение от 1В до 10В. На инверсном выходе not\_out1 напряжение, падающее со скоростью 100нс, от -1В до 10В.

Рисунок 3.3 - Временная диаграмма, полученная при анализе модели генератора

На выходе устройства появляются логические уровни 0 и 1 длительностью 50нс.

Рисунок 3.4 - Временная диаграмма, полученная при анализе модели счетчика.

Как видно из диаграммы счетчик считает от 0 до 16.

Из полученных диаграмм можно сделать заключение о работоспособности отдельных модулей.

## 4.Исследование структурной модели устройства в целом с использованием моделей узлов

Для построения модели устройства в целом используется структурное описание. При этом модель представляется в виде совокупности отдельных модулей описанных ранее и связей между ними. Это позволяет применить принцип декомпозиции, значительно упрощающий процесс разработки модели.

В качестве внешних портов ввода-вывода устройства используются следующие:

Входное напряжение (Ua);

Входное напряжение (Ub);

вход сброса (Reset1);

выход знака разности (znak)

выход цифрового кода, эквивалентного разности напряжений (cod\_out).

*library IEEE;*

*use IEEE. STD\_LOGIC\_1164. all, IEEE. Numeric\_STD. all;*

*entity V\_metr is*

*port (Ua: in integer range - 5 to 5;*

*Ub: in integer range - 5 to 5;*

*Reset1: in std\_logic;*

*znak: out std\_logic;*

*cod\_out: out unsigned (3 downto 0));*

*end V\_metr;*

*Ниже приведен листинг, описывающий работу устройства. Узлы, разработанные ранее, подключены в составе библиотеки "RGR" (library RGR).*

*Так же используются некоторые простейшие логические элементы (и, или), их описание не приводилось в виду очевидности.*

architecture A10 of V\_metr is

component GLIN

port (out1: out integer range 0 to 10;

not\_out1: out integer range - 10 to 0);

end component;

component comp

port (U1: in integer range - 10 to 10;

U2: in integer range - 10 to 10;

U\_out: out std\_logic);

end component;

component gen

port (gen\_out: out std\_logic);

end component;

component dif\_oy

port (In1: in integer range - 5 to 5;

In2: in integer range - 5 to 5;

Out1: out integer range - 10 to 10);

end component;

component counter

port (reset: in std\_logic;

up: in std\_logic;

C\_out: out unsigned (3 downto 0));

end component;

component And\_op

port (a: in std\_logic;

b: in std\_logic;

z: out std\_logic);

end component;

component Or\_op

port (a: in std\_logic;

b: in std\_logic;

c: in std\_logic;

z: out std\_logic);

end component;

component not\_op

port (a: in std\_logic;

z: out std\_logic);

end component;

signal Ua\_b: integer range - 10 to 10;

signal GLN: integer range 0 to 10;

signal NGLN: integer range - 10 to 0;

signal GND: integer range - 10 to 10;

signal nznak,R1: std\_logic;

signal K1,K2,NK1,NK2,NK1\_takt,NK2\_takt,clock: std\_logic;

signal takt,zk1,zk2,zk1\_k2,zk2\_k1,Res: std\_logic;

begin

u1: dif\_oy port map (In1=>Ua, In2=>Ub,Out1=>Ua\_b);

u2: comp port map (U1=>Ua\_b, U2=>GND, U\_out=>nznak);

u3: not\_op port map (a=>nznak, z=>znak);

u4: GLIN port map (out1=>GLN,not\_out1=>NGLN);

u5: comp port map (U1=>Ua\_b, U2=>GLN, U\_out=>K1);

u6: comp port map (U1=>Ua\_b, U2=>NGLN, U\_out=>K2);

u7: Or\_op port map (a=>K1, b=>K1,c=>K1, z=>NK1);

u8: not\_op port map (a=>K2, z=>NK2);

u9: gen port map (gen\_out=>takt);

u10: And\_op port map (a=>NK1,b=>takt,z=>NK1\_takt);

u11: And\_op port map (a=>NK2,b=>takt,z=>NK2\_takt);

u12: Or\_op port map (a=>NK1\_takt,b=>NK2\_takt,c=>NK2\_takt,z=>clock);

u13: counter port map (reset=>Res,up=>clock,C\_out=>cod\_out);

u14: not\_op port map (a=>NK1,z=>zk1);

u15: not\_op port map (a=>NK2,z=>zk2);

u16: And\_op port map (a=>zk1,b=>NK2,z=>zk1\_k2);

u17: And\_op port map (a=>NK1,b=>zk2,z=>zk2\_k1);

u18: Or\_op port map (a=>zk1\_k2,b=>zk2\_k1,c=>zk2\_k1,z=>R1);

u19: And\_op port map (a=>R1,b=>Reset1,z=>Res);

В результате анализа модели устройства получена следующая временная диаграмма.

Рисунок 3.5 - Временная диаграмма, полученная при анализе модели устройства

Из диаграммы видно, что на выходе cod\_out формируется цифровой код, эквивалентный разности напряжений Ua-Ub, поданных на вход устройства. С выхода znak снимается знак разности: 1, если Ua-Ub<0 и 0 если Ua-Ub>=0.

## 4.1 Анализ пригодности разработанных моделей для имплементации в PLD

При проектировании цифрового устройства, для последующего имплементации в PLD, появляется ряд дополнительных трудностей, связанных с невозможностью использования некоторых типов данных и языковых конструкций. Так же приходится учитывать особенности имплементации для ПМЛ разных фирм и устанавливаемые ограничения: максимальная частота, количество доступных элементов и т.д. Наиболее эффективным способом в этом случае является построение структурных моделей с использованием библиотек компонентов поставляемых производителем конкретного типа ПМЛ. Однако в данной расчетно-графической работе этого не было сделано, так как основной целью было - приобретение навыков построение моделей отдельных узлов и структурных моделей. В целом при построении моделей цифровых узлов не использовались запрещенные языковые конструкции и типы данных, также были учтены ограничения на количество доступных элементов памяти и максимальную частоту сигнала.

## Заключение

В ходе выполнения работы была построена и исследована модель цифрового измерителя разности двух напряжений.

Модель содержит не только цифровые блоки, но и аналоговые устройства. Были построены временные диаграммы сигналов, по которым можно судить об особенностях функционирования устройства.

При необходимости, для большей наглядности и удобства, модель можно дополнить дешифратором и цифровым семисегментным индикатором.