**Дискретное устройство (ДУ)**

**Содержание**

1. Задание на курсовой проект

2. Введение

3. Проектирование дискретного устройства на элементах малой интеграции.

3.1. Проектирование блока триггера.

3.2. Проектирование микроопераций

3.2.1. Микрооперация установки триггера в " 0 ".

3.2.2 Микрооперация записи данных.

3.2.3. Микрооперация сдвига кода влево на один разряд.

3.2.4. Логическая микрооперация (“И”).

3.2.5. Микрооперация счета с параллельным переносом.

3.2.6. Микрооперация вычитание.

3.2.7. Микрооперация сравнения (>).

3.2.8. Микрооперация преобразования в дополнительный код.

3.2.9. Микрооперация свертки по модулю 2

(тип переноса последовательный (сквозной) )

3.3. Построение объединенных функций возбуждения триггеров.

3.4. Расчет быстродействия ДУ.

4. Проектирование ДУ на БИС.

4.1. Проектирование схемы ДУ.

4.2. Карта программирования ПЗУ.

**1. Задание на курсовой проект.**

Спроектировать дискретное устройство (ДУ) выполняющее следующие действия :

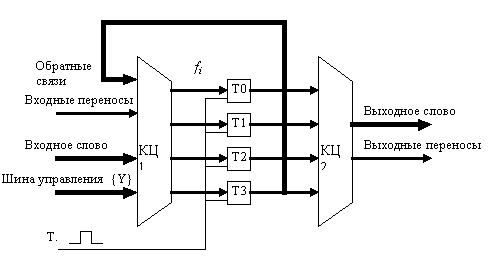
1. сброс элементов памяти
2. запись значения входного слова в ДУ
3. сдвиг кода в ДУ влево
4. логическая операция «&» (логическое “И”)
5. счетчик (-1) с параллельным переносом
6. арифметическая операция вычитание.
7. операция сравнения (>)
8. преобразование в дополнительный код
9. свертка по модулю 2 (тип переноса последовательный).

Дискретное устройство проектируется на JK - триггерах

Серия МИС 500, серия ПЗУ 1611

**2. Введение.**

При синтезе проектируемое устройство представляется в виде структуры, представленной на рисунке 2.1



*рисунок 2.1.*

Проектирование дискретного устройства сводится к синтезу элементарного дискретного устройства, обрабатывающего один бит информации и включающего один элемент памяти (триггер).

Входная комбинационная схема обеспечивает выполнение операций из множества Y, результатом которых являются соответствующие значения функций возбуждения f i для каждого из входов триггера:

– установка элементов памяти в нулевое состояние.

– запись входного слова в элементы памяти

– сдвиг влево

– логическую операцию

– декремент

– операция вычитания

– преобразование в дополнительный код.

Выходная комбинационная схема обеспечивает выполнение операции сравнения и формирование свертки содержимого ДУ.

Синтез элементарного ДУ сводится к получению функций возбуждения для каждой операции отдельно и, далее, построению объединенных функций возбуждения триггера элементарного ДУ. Получение функций возбуждения для каждой микрооперации сводится к выполнению следующих этапов :

– построение таблицы для выполняемой операции

– определение значений функций возбуждения для каждой таблицы выполняемой

операции с учетом словаря переходов заданного триггера

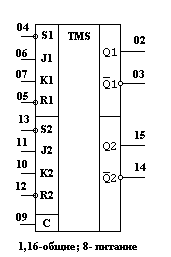
– определение логической функции для каждого входа заданного триггера

– минимизация логической функции в требуемом логическом базисе.

**3. Проектирование дискретного устройства на элементах малой интеграции.**

1. **Выбор триггера.**

В качестве триггера для ДУ выберем МС К500ТВ135, объединяющую в одном корпусе два двухтактных JK-триггера с возможностью установки “0” и “1”:



К500ТВ135

**3.2 Проектирование микроопераций.**

**3.2.1. Микрооперация установки триггера в " 0 ".**

При подаче тактового импульса на синхронизирующие входы триггеров и при разрешении операции обнуления (Y=0) по заднему фронту импульса, информация, находящаяся на входах триггеров переписывается на их выходы, т.о. обнуление триггеров происходит в том случае, если к моменту подачи тактового импульса на J-входах триггеров установлены уровни логической единицы, а на K-входах – уровни логического нуля.

|  |  |  |  |
| --- | --- | --- | --- |
| Qt | Qt+1 | J | K |
| 0 | 0 | 0 | \* |
| 1 | 1 | \* | 1 |



**3.2.2. Микрооперация записи данных.**

Когда операция записи данных разрешена, т.е. Y =1, и при подаче на синхровходы триггеров тактового импульса С по его заднему фронту, информация, находящаяся на входах триггеров, переписывается в ячейки памяти дискретного устройства и появляется на выходах Qi. При изменении входных данных, т.е. при переключении сигналов X i они с задержкой распространения импульса по цепям микросхем, участвующих в микрооперации записи данных, появляются на входах триггеров. При подаче следующего тактового импульса С микрооперация записи данных повторяется.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X | Qt | Qt+1 | J | K |  | J |  |  | K |  |  |
| 0 | 0 | 0 | 0 | \* |  |  | X |  |  | X |  |
| 0 | 1 | 0 | \* | 1 |  |  |  |  |  |  |  |
| 1 | 0 | 1 | 1 | \* |  | 0 | 1 |  | \* | \* |  |
| 1 | 1 | 1 | \* | 0 |  | \* | \* | Qt | 1 | 0 | Qt |



**3.2.3. Микрооперация сдвига кода вправо на один разряд.**

Когда Y =2, по заднему фронту тактового импульса информация со входов триггера поступает на их выходы, а затем осуществляется сдвиг кода, хранящегося в данный момент в ячейках памяти, в сторону старших разрядов т.е. вправо. При сдвиге каждый запоминающий элемент должен :

- передать хранимую информацию на элемент Ti+1

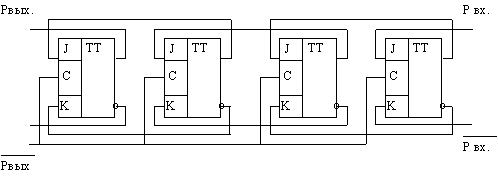
- изменить свое состояние за счет приема информации от триггера Т i-1 при сдвиге на один разряд вправо.

Передача от триггера Тi и изменение его состояния не могут проходить одновременно. В данном курсовом проекте эта трудность исключается за счет использования синхронных двухступенчатых JK-триггеров с динамическим управлением записью. Внутренняя организация таких триггерных схем предусматривает разделение во времени этапов приема входной информации и смены выходной. В них по переднему фронту синхронизирующего сигнала происходит прием информации, а по заднему - изменение состояния.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pi | Qt | Qt+1 | J | K |  |  | J |  |  | K |
| 0 | 0 | 0 | 0 | \* |  | Qt |  |  | Qt |  |
| 0 | 1 | 0 | \* | 1 |  |  |  |  |  |  |
| 1 | 0 | 1 | 1 | \* | Pi |  |  | Pi | \* | 1 |
| 1 | 1 | 1 | \* | 0 |  | 1 | \* |  |  |  |



Схема цепи, реализующей данную операцию, изображена на рис. 3.3



*рисунок 3.3*

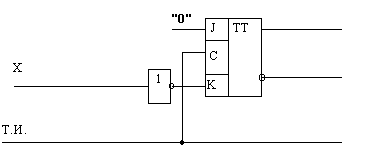
**3.2.4. Логическая микрооперация.**

В данном дискретном устройстве разрабатывается логическая микрооперация «Логическое “И”» .Когда Y=3 и на синхровходы триггеров подается тактовый импульс, по его заднему фронту информация на входах переписывается на выходы Q с задержкой прохождения сигнала через триггер. Затем выполняется заданная логическая операция над содержимом ячейки памяти и входными данными.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X | Qt | Qt+1 | J | K |  |  | J |  |  | K |
| 0 | 0 | 0 | 0 | \* |  | Qt |  |  | Qt |  |
| 0 | 1 | 0 | \* | 1 |  |  |  |  |  |  |
| 1 | 0 | 0 | 0 | \* | X | 0 | 0 | X | \* | \* |
| 1 | 1 | 1 | \* | 0 |  | \* | \* |  | 1 | 0 |



Схема цепи, реализующей данную операцию, изображена на рис. 3.4



*рисунок 3.4*

**3.2.5. Микрооперация счета (-1) с последовательным переносом.**

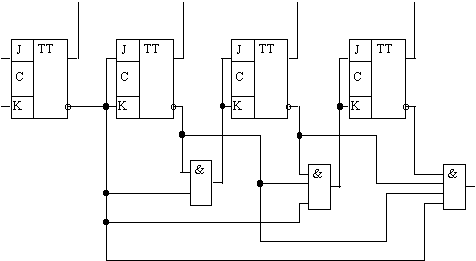
Учитывая тот факт, что синтезированный двухступенчатый триггер переключается по заднему фронту, получаем следующие функции возбуждения триггеров.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Zi-1 | Qt-1 | Qt | J | K | Zi |  | Qt | -1 | J | | Qt | -1 | Zi |
| 0 | 0 | 0 | 0 | \* | 0 | Zi-1 | 0 | \* | Zi-1 | | 0 | 0 |
| 0 | 1 | 1 | \* | 0 | 0 |  | 1 | \* |  | | 1 | 0 |
| 1 | 0 | 1 | 1 | \* | 1 |
| 1 | 1 | 0 | \* | 1 | 0 |  | Qt | -1 | K |
|  | | | | | | Zi-1 | \* | 0 |
|  | | | | | |  | \* | 1 |



Или  то есть, если  иначе 

Схема цепи, реализующей данную операцию, изображена на рис. 3.5



*рисунок 3.5*

**3.2.6. Микрооперация арифметического вычитания.**

В данном дискретном устройстве разработана арифметическая операция сложения.-Анализируя исходное состояние триггера, перенос, в соответствии со словарем переходов JK-триггеров составляем таблицу.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X | Zi | Qt | Qt+1 | Zi+1 | J | K |  | Х |  | | J |
| 0 | 0 | 0 | 0 | 0 | 0 | \* | Zi | 0 | \* | \* | 1 |
| 0 | 0 | 1 | 1 | 0 | \* | 0 |  | 1 | \* | \* | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | \* |  |  | Qt |
| 0 | 1 | 1 | 0 | 0 | \* | 1 |  | Х |  | | K |
| 1 | 0 | 0 | 1 | 1 | 1 | \* | Zi | \* | 0 | 1 | \* |
| 1 | 0 | 1 | 0 | 0 | \* | 1 |  | \* | 1 | 0 | \* |
| 1 | 1 | 0 | 0 | 1 | 0 | \* |  |  | Qt |
| 1 | 1 | 1 | 1 | 1 | \* | 0 |  | Х |  | Zi+1 | |
|  | | | | | | | Zi | 1 | 1 | 0 | 1 |
|  | | | | | | | | 1 | 0 | 0 | 0 |
|  | | | | | | | | | Qt |

Функции возбуждения триггера для данной операции имеют следующий вид:



Схема цепи, реализующей данную операцию, изображена на рис. 3.6

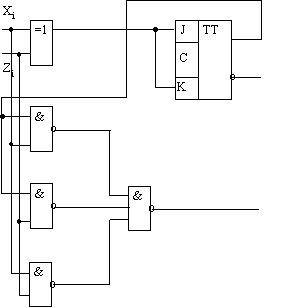


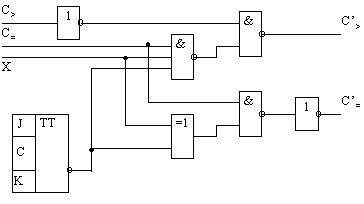
рисунок 3.6

Данная операция реализуется по следующим формулам.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C> | C= | X | Qt | C’> | C’= |  | | | C’> | | |
| 0 | 0 | 0 | 0 | 0 | 0 |  | | C= | |  | |  | | С> | С= |
| 0 | 0 | 0 | 1 | 0 | 0 |  |  |  | 1 |  |  |  | < | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |  | 1 | \* | \* | 1 |  |  | = | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | C> | 1 | \* | \* | 1 | Qt |  | > | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |  |  |  |  |  |  |  | x | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 |  | | | X | |  |
| 0 | 1 | 1 | 0 | 1 | 0 |  | | | C’= | | |
| 0 | 1 | 1 | 1 | 0 | 1 |  | | C= | |  | |
| 1 | 0 | 0 | 0 | 1 | 0 |  |  | 1 |  |  |  |
| 1 | 0 | 0 | 1 | 1 | 0 |  |  | \* | \* |  |  |
| 1 | 0 | 1 | 0 | 1 | 0 | C> |  | \* | \* |  | Qt |
| 1 | 0 | 1 | 1 | 1 | 0 |  |  |  | 1 |  |  |
| 1 | 1 | 0 | 0 | \* | \* |  | | | X | |  |
| 1 | 1 | 0 | 1 | \* | \* |
| 1 | 1 | 1 | 0 | \* | \* |
| 1 | 1 | 1 | 1 | \* | \* |

Функции переносов для данной операции имеют следующий вид:





*рисунок 3.7.*

**3.2.8. Микрооперация преобразования в дополнительный код.**

Преобразование в дополнительный код произведем согласно таблице.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **t** | | | | **t+1** | | | | **JK3** | | **JK2** | | **JK1** | | **JK0** | |
| Q3 | Q2 | Q1 | Q0 | Q3 | Q2 | Q1 | Q0 | J3 | K3 | J2 | K2 | J1 | K1 | J0 | K0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | \* | 0 | \* | 0 | \* | 0 | \* | |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | \* | 1 | \* | 1 | \* | \* | 0 | |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | \* | 1 | \* | \* | 0 | 0 | \* | |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | \* | 1 | \* | \* | 1 | \* | 0 | |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | \* | \* | 0 | 0 | \* | 0 | \* | |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | \* | \* | 1 | 1 | \* | \* | 0 | |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | \* | \* | 1 | \* | 0 | 0 | \* | |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | \* | \* | 1 | \* | 1 | \* | 0 | |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | \* | 0 | 0 | \* | 0 | \* | 0 | \* | |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | \* | 1 | 1 | \* | 1 | \* | \* | 0 | |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | \* | 1 | 1 | \* | \* | 0 | 0 | \* | |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | \* | 1 | 1 | \* | \* | 1 | \* | 0 | |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | \* | 1 | \* | 0 | 0 | \* | 0 | \* | |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | \* | 1 | \* | 1 | 1 | \* | \* | 0 | |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | \* | 1 | \* | 1 | \* | 0 | 0 | \* | |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | \* | 1 | \* | 1 | \* | 1 | \* | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Q0 |  |  |  | J3 |  | Q0 |  |  |  | K3 |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 0 | 1 | 1 | 1 |  |  | \* | \* | \* | \* |  |  |  |  |  |
|  | 1 | 1 | 1 | 1 |  |  | \* | \* | \* | \* |  |  |  |  |  |
|  | \* | \* | \* | \* |  |  | 1 | 1 | 1 | 1 |  |  |  |  |  |
|  | \* | \* | \* | \* |  |  | 0 | 1 | 1 | 1 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Q1 |  |  |  |  |  | Q1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

J3=Q2∨ Q1∨ Q0

K3=Q2∨ Q1∨ Q0

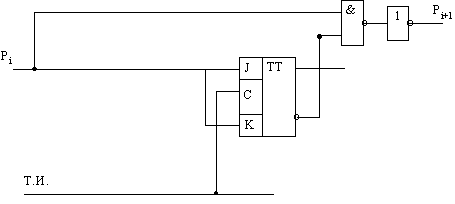
Если ввести понятие переноса, то микрооперация преобразования в дополнительный код реализуется по следующим формулам .

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pi | Qt | Qt+1 | J | K | Pi+1 |  | Qt |  | J | | Qt |  | Pi+1 |
| 0 | 0 | 1 | 1 | \* | 0 | Pi | 1 | \* | Pi | | 0 | 1 |
| 0 | 1 | 0 | \* | 1 | 0 |  | 0 | \* |  | | 0 | 0 |
| 1 | 0 | 0 | 0 | \* | 1 |
| 1 | 1 | 1 | \* | 0 | 0 |  | Qt |  | K |
|  | | | | | | Pi | \* | 1 |
|  | | | | | |  | \* | 0 |

Функции возбуждения триггера для данной операции имеют следующий вид:



Схема цепи, реализующей данную операцию, изображена на рис. 3.8.



*рисунок 3.8*

**3.2.9. Микрооперация** **свертки по модулю 2 (послед.).**

Свертка по модулю 2 применяется для контроля кода.



Схема цепи, реализующей данную операцию, изображена на рис. 3.9.

=1

=1

=1

=1

Q0

Q1

Q2

Vi

Q3

Vi+1

&

U8

1

*рисунок 3.9*

**3.3. Построение объединенных функций возбуждения**.

Построение объединенных функций возбуждения триггеров реализуется следующим образом :

- построение объединенной таблицы функционирования для каждой микрооперации и каждого установочного входа триггера ;

запись и минимизация функций возбуждения .

Таблица 3.1 является объединенной таблицей функционирования ДУ. В ней строки соответствуют микрооперациям ,а столбцы установочным входам триггеров и переносам. Для каждого установочного входа записывается функция возбуждения

На основании оставленной таблицы синтезируем функции возбуждения триггеров

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Q3 | | Q2 | | Q1 | | Q0 | |
|  | J3 | K3 | J2 | K2 | J1 | K1 | J0 | K0 |
| Запись 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Запись слова | x3 |  | x2 |  | x1 |  | x0 |  |
| Сдвиг вправо | P3 |  | P2 |  | P1 |  | P0 |  |
| Логическая операция | 0 |  | 0 |  | 0 |  | 0 |  |
| Вычитание |  |  |  |  |  |  |  |  |
| -1 (пар.) | Z`3 | Z`3 | Z`2 | Z`2 | Z`1 | Z`1 | Z`0 | Z`0 |
| дополнитель-ный код | P3` | P3` | P2` | P2` | P1` | P1` | P0` | P0` |



























Преобразуем выражения к требуемому базису:







**3.4. Расчет быстродействия ДУ.**

Быстродействие дискретного устройства следует оценивать по максимальной частоте тактовых импульсов, поступающих на синхронизирующий вход. Максимальная частота тактовых импульсов – это максимально допустимая частота подаваемых в дискретное устройство импульсов, не приводящих к нарушению его работы. В общем случае для расчета максимальной тактовой частоты необходимо определить минимальные длительности тактового импульса и паузы между тактовыми импульсами. Тогда частоту fmax можно определить по формуле :



Длительность такта для синхронных схем дискретного устройства определяется исходя из быстродействия используемого триггера. Длительность паузы между тактовыми импульсами определяется временем переходного процесса в дискретном устройстве и оценивается максимальной длиной функциональной цепи :

 ,

где  ­– максимальная задержка сигнала на одном элементе;

*n* – число уровней функциональной цепи (глубина КСх);

Вычисляем максимальную тактовую частоту дискретного устройства :



**4. Проектирование ДУ на БИС.**

**4.1. Проектирование схемы ДУ.**

**4.2. Карта программирования ПЗУ.**