БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра программного обеспечения информационных технологий

Факультет ФНиДО

Специальность ПОИТ

Контрольная работа № 1

по дисциплине «Основы компьютерной техники»

часть 1

Вариант № 6

Выполнила студентка: Ващило А.Г.

Минск 2014

**Контрольная работа 1**

## Задание 1.1

Найти C1=A+B, C2=A–B, C3=B–A, C4=–A–B, где A=–477, B=+1794; при выполнении вычислений необходимо использовать двоично-десятичный обратный код.

**Решение**

A(2-10) = – 0000 0100 0111 0111

B(2-10) = + 0001 0111 1001 0100

[A]ПК = 1. 0000 0100 0111 0111

[B]ПК = 0. 0001 0111 1001 0100

**[C1]ПК = A+B =**

1. 1111 1011 1000 1000 [A]ИК = [A]ОК+6

+ 0. 0001 0111 1001 0100 \_ [B]ОК

10. 0001 0011 0001 1100

+ 1 \_ корректировка переполнения

0. 0001 0011 0001 1101

+ 1010 \_ корректировка в тетрадах, где не было переноса

**0. 0001 0011 0001 0111 = 1317(10)**

|C2|=|C3|

–[C2]ПК = [C3]ПК =

0. 0000 0100 0111 0111 – [|A|]ОК

+ 0. 0001 0111 1001 0100 \_ – [|B|]ОК

0. 0001 1100 0000 1011

+ 0110 0110 0110 \_ – корректировка

0. 0010 0010 0111 0001 = 2271(10)

**[С2]ПК = 1. 0010 0010 0111 0001 = –2271(10)**

**[С3]ПК = 0. 0010 0010 0111 0001 = 2271 (10)**

**[C4]ПК = –А–В =**

0. 0000 0100 0111 0111 [–A]ОК

+ 1. 1110 1000 0110 1011 \_ [–B]ИК=[–B]ОК+6

1. 1110 1100 1110 0010

+ 0110 \_ корректировка в тетрадах, где был перенос

1. 1110 1100 1110 1000 [C4]ИК

**1. 0001 0011 0001 0111 = –1317(10)**

## Задание 1.2

Задание предполагает выполнение заданной операции над числами А и В, представленными с плавающей точкой.

У2 – умножение с младшего разряда множителя со сдвигом множителя.

A = –0.56 × 10–2

B = 0.51

Код – дополнительный.

**Решение**

С точки зрения представления чисел с плавающей точкой поиск произведения С2 =А\*В сводится к поиску С2п и С2м, соответственно порядку и мантиссе произведения на основании порядка ап и мантиссы ам множимого и порядка вп и мантиссы вм множителя. Учитывая общую запись чисел с плавающей точкой, произведение двух операндов представляется в виде:

С2 = А\*В = 2 ап × а м × 2 вп × вм = 2 ап+вп × а м × вм = 2 С2п × С2м.

Отсюда следует, что порядок произведения определяется как сумма порядков сомножителей, а мантисса – как произведение мантисс сомножителей. Однако, учитывая то, что при умножении мантисс может произойти нарушение нормализации, в результате указанных действий будет найдено предварительное значения порядка и мантиссы искомого произведения. Окончательные значения порядка и мантиссы произведения будут определены после устранения нарушения нормализации в предварительном результате. При формировании мантиссы произведения нормализованных чисел с плавающей точкой возможно только один вид нарушения нормализации – нарушение нормализации справа от точки с появлением нуля в старшем разряде мантиссы.

Сформируем прямой код для чисел A и B, учитывая, что разрядность порядка – 3, мантиссы – 6.

[аП]ПК = 1. 010

[аМ]ПК = 1. 100100

[bП]ПК = 0. 000

[bМ]ПК = 0. 100001

Знак произведения, представляемый знаком его мантиссы, – отрицательный, т.к. знаки мантисс сомножителей неодинаковы.

Определим предварительное значение порядка произведения:

**[C2П\*]ДК = aП+bП = 1. 010**

Найдем абсолютное значение предварительного значения мантиссы произведения.

[|C2М\*|] =

100100 – [|аМ|]ДК

× 100001 \_ – [|bМ|]ДК

100100

+ 100100 \_

10010100100

Получим абсолютное предварительное значение мантиссы искомого произведения с учетом округления:

[|C2М\*|] = 0.100101

Мантисса частного денормализована, поэтому необходим сдвиг вправо.

Тогда

**[C2П] = 1.010 [C2М] = 1.010010**

C2 = – 0.29 × 10–2

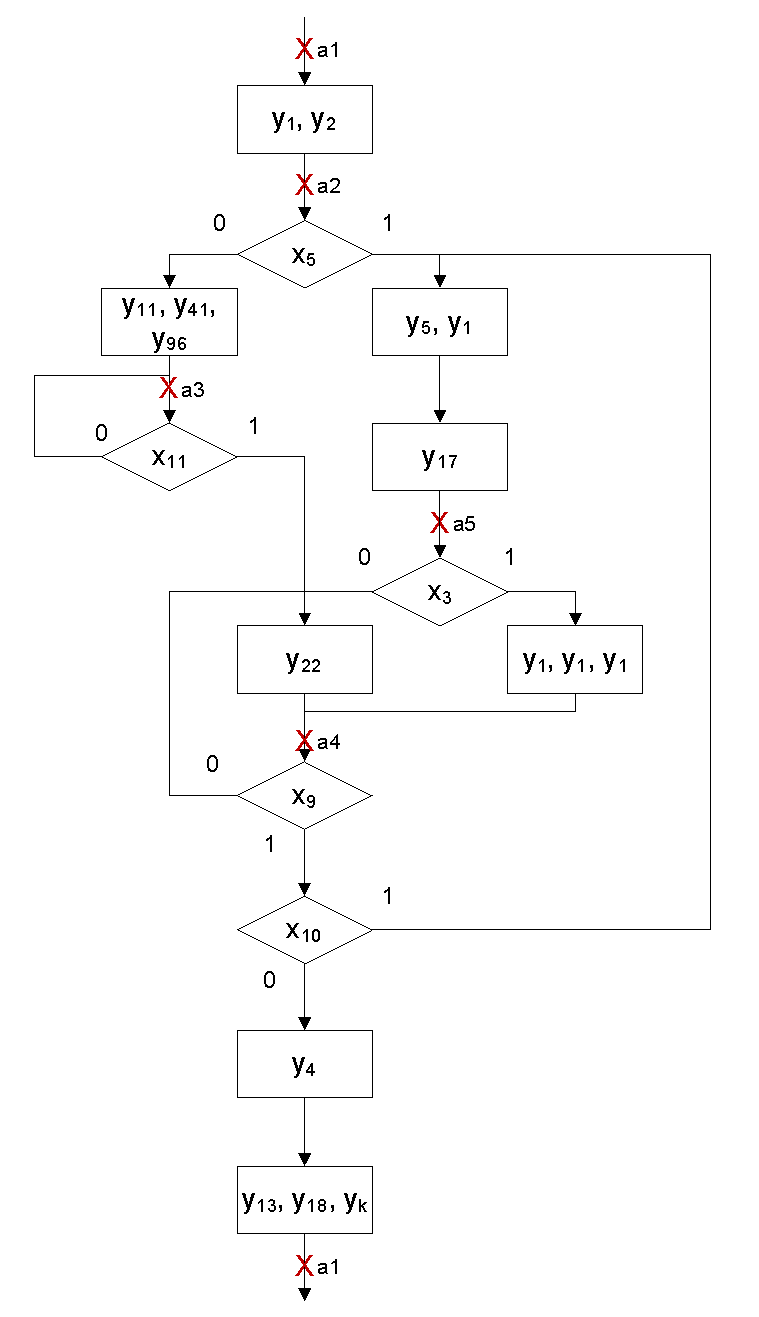
Контрольная работа 2

## Задание 2.1

Построить цифровой автомат Мили для заданной ГСА, используя D-триггер.

**Решение**

Граф автомата Мили представлен на рисунке 1.



**Рисунок 1 – ГСА с пометками для автомата Мили**

Таблица переходов D-триггера приведена в таблице 1.

**Таблица 1 — Таблица переходов D-триггера**

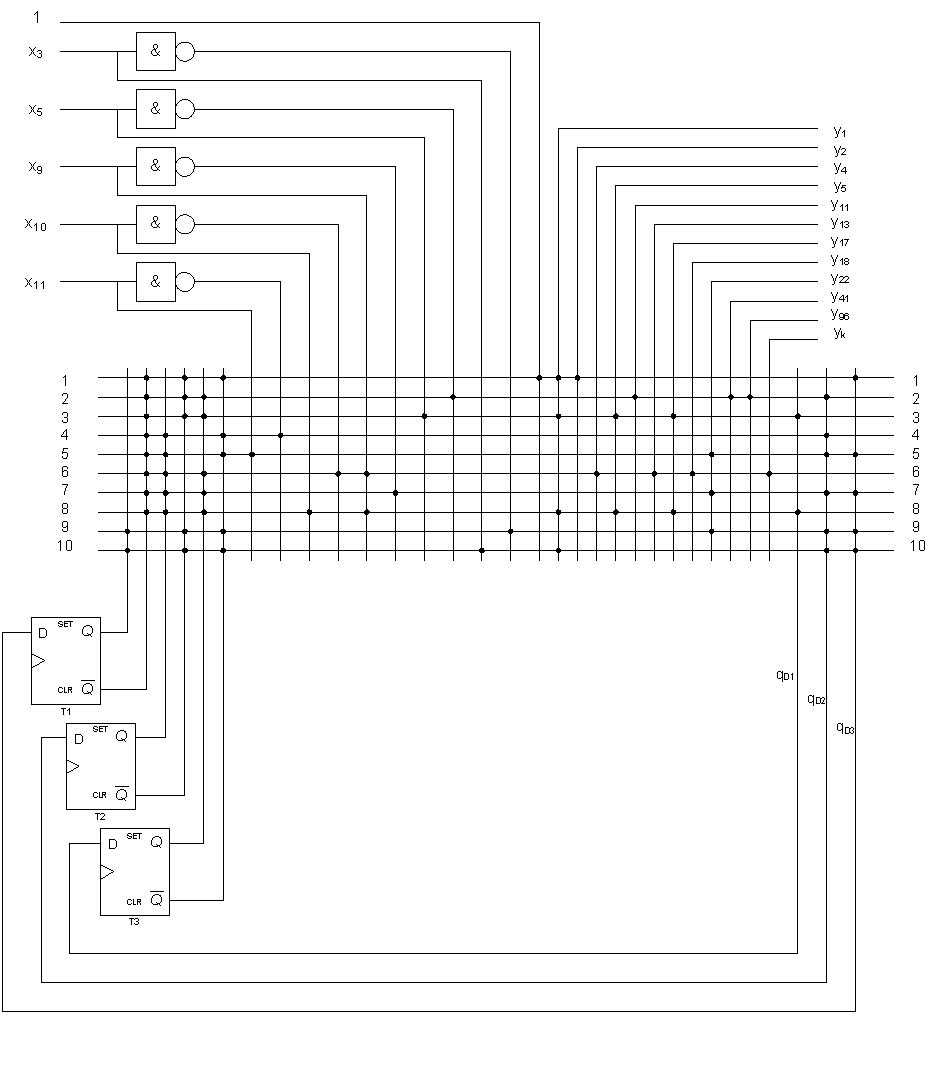
|  |  |  |
| --- | --- | --- |
| **D** | **Q(t)** | **Q(t+1)** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Составим структурную таблицу МПА (таблица 2).

**Таблица 2 — Структурная таблица МПА Мили**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № п/п | Начало пути | | | Конец пути | | Логич. условие | Вых. сигнал | Управл. памятью | | |
| A(t) | Код A(t) |  | A(t+1) | код A(t+1) | qD1 | qD2 | qD3 |
| 1 | a1 | 000 |  | a2 | 001 | 1 | y1y2 | 0 | 0 | 1 |
| 2 | a2 | 001 |  | a3 | 010 |  | y11y41y96 | 0 | 1 | 0 |
| 3 |  |  |  | a5 | 100 |  | y5y1y17 | 1 | 0 | 0 |
| 4 | a3 | 010 |  | a3 | 010 |  | – | 0 | 1 | 0 |
| 5 |  |  |  | a4 | 011 |  | y22 | 0 | 1 | 1 |
| 6 | a4 | 011 |  | a1 | 000 |  | y4y13y18yk | 0 | 0 | 0 |
| 7 |  |  |  | a4 | 011 |  | y22 | 0 | 1 | 1 |
| 8 |  |  |  | a5 | 100 |  | y5y1y17 | 1 | 0 | 0 |
| 9 | a5 | 100 |  | a4 | 011 |  | y22 | 0 | 1 | 1 |
| 10 |  |  |  | a4 | 011 |  | y1y1y1 | 0 | 1 | 1 |

Схема МПА представлена на рисунке 2.



**Рисунок 2 – Схема МПА Мили**

## Задание 2.1

Построить цифровой автомат Мили для заданной ГСА, используя D-триггер.

Написать микропрограмму, соответствующую заданной ГСА (рисунок 3), с учетом заданных множества микроопераций (Y=25), множества проверяемых условий (Х=31), ёмкости запоминающего устройства (ЗУ=1000) и начального адреса размещения микропрограммы (МП=421) в ЗУ.

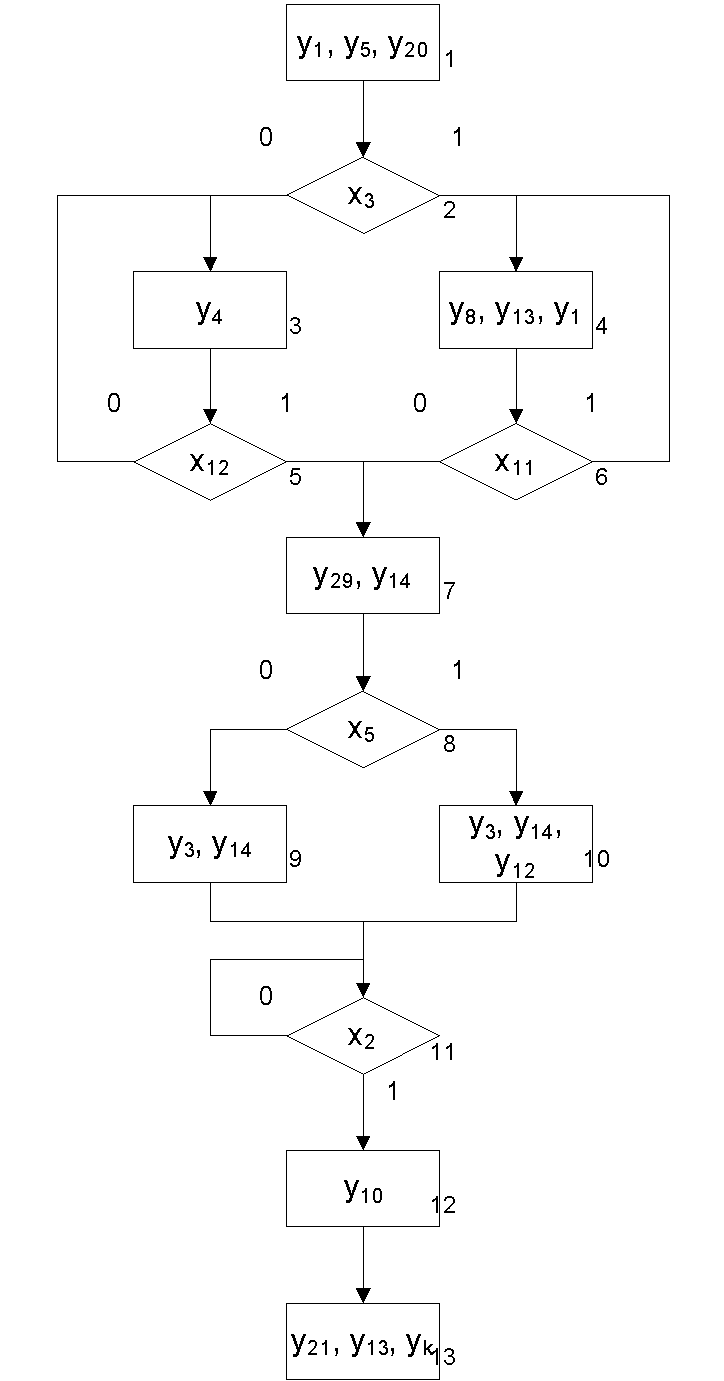
В каждом адресе запоминающего устройства может храниться 16 бит информации. Обозначение уk соответствует микрооперации, обозначающей последнюю микрокоманду в микропрограмме.

Если это допускает длина микрокоманды, использовать модификатор дисциплины перехода.

**Решение**

Исходя из характеристик управляемого объекта следует:

* длина кода для кодирования микрооперации равна k=5, т.к. количество выполняемых команд – Y=25;
* длина кода для кодирования условий равна p=5, т.к. количество проверяемых условий – X=31;
* длина кода адреса равна десяти, т.к. количество адресов в памяти, учитывая, что длина адресуемой ячейки равна 16 битам (т.е. двум байтам), равно 1000.



**Рисунок 3 – ГСА для задания 2**

Тогда формат микрокоманд для заданного управляемого объекта имеет вид, представленный на рисунке 4.



**Рисунок 4 – Формат микрокоманд**

Формат операционной микрокоманды (МКО) имеет длину 16 бит и включает:

* поле типа микрокоманды (Т), имеющее длину в один бит, занимает 0-ой разряд микрокоманды; в этом поле для данного типа микрокоманды записано значение «1»;
* поле первой микрооперации (Y1), которое занимает разряды с 1-го по 5;
* поле второй микрооперации (Y2), которое занимает разряды с 6-го по 10;
* поле второй микрооперации (Y3), которое занимает разряды с 11-го по 15;

Таким образом, данный формат позволяет задавать, при необходимости, в одной микрокомандой три микрооперации.

Значение 00000 в поле Y1 – соответствует операции yk.

Формат микрокоманды перехода (МКП) имеет длину 16 бит и включает:

1. поле типа микрокоманды (Т), имеющее длину в один бит и занимающее 0-ой разряд микрокоманды; в этом поле для данного типа микрокоманды записано значение «0»;
2. поле проверяемого условия (Х), которое занимает разряды с 1-го по 5;
3. поле адреса (А), которое занимает разряды с 6-го по 15.

Поле модификатора дисциплины перехода не используется, т.к. длина микрокоманды не позволяет его реализовать, поэтому для безусловного перехода используется код 0. Если условие выполняется, следует переход по указанному адресу, в противном случае – переход к следующей ячейке.

При составлении микропрограммы с помощью микрокоманд необходимо реализовать все вершины, имеющиеся в ГСА, и обеспечить необходимые ветвления процесса.

Микропрограмма для ГСА приведена в таблице 2.

В приведенной таблице:

* в первой графе фиксируется номер строки;
* во второй графе приводится номер вершины, реализуемой микрокомандой этой строки;
* в третьей графе указан в двоичном коде адрес расположения данной микрокоманды в запоминающем устройстве;
* в четвертой графе располагается код микрокоманд;
* в пятой графе указаны номера вершин – ссылки, адреса которых должен быть указан в данной команде перехода.

В приведенной микропрограмме кодировка микроопераций и проверяемых условий осуществлена по их индексам в двоичном коде. Подчеркнутые коды адресов в микрокомандах перехода заполняются после записи последней строки формируемой микропрограммы, используя коды, соответствующие вершинам-ссылкам, указанные в графе «Адрес» строк.

**Таблица 3 — Микропрограмма**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N  пп | N  вер. | Адрес расположения микрокоманды в ЗУ | Код микрокоманды | Примечание |
|  | 1 | 01101 00101 | 1.00001.00101.10100 |  |
|  | 2 | 01101 00110 | 0.00011.01101 01010 | 4 |
|  | 3 | 01101 00111 | 1.00100.00000.00000 |  |
|  | 5 | 01101 01000 | 0.01100.01101 01100 | 7 |
|  | 5’ | 01101 01001 | 0.00000.01101 00111 | 3 |
|  | 4 | 01101 01010 | 1.01000.01101.00001 |  |
|  | 6 | 01101 01011 | 0.01011.01101 01010 | 4 |
|  | 7 | 01101 01100 | 1.11101.01110.00000 |  |
|  | 8 | 01101 01101 | 0.00101.01101 10001 | 10 |
|  | 9 | 01101 01110 | 1.00011.01110.00000 |  |
|  | 11 | 01101 01111 | 0.00010.01101 10011 | 12 |
|  | 11’ | 01101 10000 | 0.00000.01101 01111 | 11 |
|  | 10 | 01101 10001 | 1.00011.01110.01100 |  |
|  | 10’ | 01101 10010 | 0.00000.01101 01111 | 11 |
|  | 12 | 01101 10011 | 1.01010.00000.00000 |  |
|  | 13 | 01101 10100 | 1.10101.01101.00000 |  |
|  | 13’ | 01101 10101 | 1.00000.00000.00000 |  |