МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

**Факультет информационных технологий и робототехники (ФИТР)**

**Отчёт по лабораторной работе №3**

По дисциплине: «Основы защиты информации»

На тему: «РЕЖИМЫ РАБОТЫ БЛОЧНЫХ ШИФРОВ»

Выполнил: ст. г. 10701219 Харлап С. А.

Приняла: Белова С.В.

Минск 2020

**Цель работы**

Изучить режимы работы блочных шифров и их применение.

**Задания на лабораторную работу**

1. Привести схему и уравнения процессов шифрования и расшифрования для своего варианта задания (Взять из лекции 4).
2. Выполнить контрольный пример для своего варианта задания.
3. Написать программу шифрования и расшифрования открытого текста, состоящего из произвольного количества символов, в одном из режимов согласно варианту.

Открытый текст и ключ, а также дополнительные параметры, необходимые для режима (вектор инициализации, счетчики), вводит пользователь. Результат шифрования должен выводиться в десятичном или двоичном виде.

**Результаты выполнения**

**Вариант 2) CBC.**

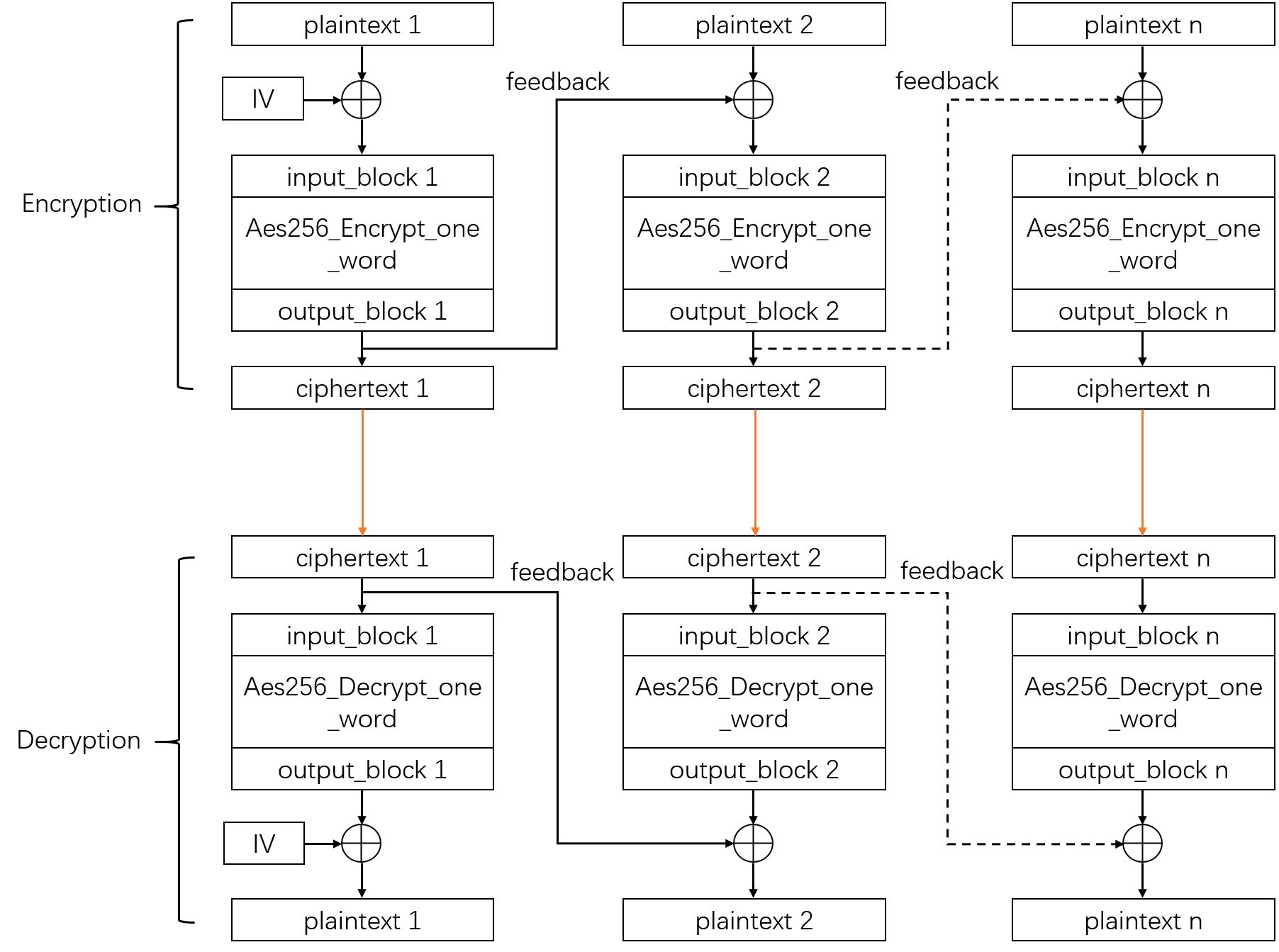
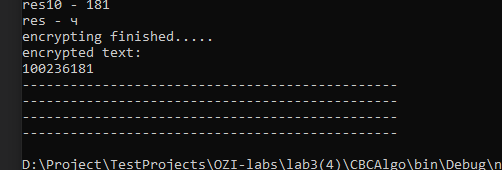


Рисунок 1 – Схема режима CBC

****

**Abc key 412**

**Вывод:** изучил режимы работы блочных шифров и их применение. Написал программу шифрования и расшифрования открытого текста, состоящего из произвольного количества символов, в режиме шифрования со счетчиком.

**Приложение А**

using System;

using WindowsFormsApp1;

using Microsoft.VisualBasic;

namespace CBCAlgo

{

public class CBC

{

private readonly ILogger \_logger;

public string Iv { get; set; } = "00000000";

public CBC(ILogger logger)

{

this.\_logger = logger;

}

public string Encrypt(string text, int key)

{

var answer = "";

this.\_logger.Log("encrypting started.....");

var des = new S\_Des(this.\_logger);

des.InitKeys(key);

var previous = this.Iv;

foreach (var character in text)

{

var bitwise = S\_Des.ToBiteString((int) character, 8);

var afterXor = S\_Des.Xor(previous, bitwise);

var encrypted = des.Encrypt(afterXor);

answer += ((int)encrypted).ToString();

previous = S\_Des.ToBiteString((int) encrypted, 8);

}

this.\_logger.Log( "encrypting finished....." );

return answer;

}

public string Decrypt(string text, int key)

{

var answer = "";

this.\_logger.Log("decrypting started.....");

var des = new S\_Des(this.\_logger);

des.InitKeys(key);

var previous = this.Iv;

foreach (var character in text)

{

var bitwise = S\_Des.ToBiteString((int) character, 8);

var decrypted = S\_Des.ToBiteString(des.Decrypt(bitwise), 8);

var afterXor = S\_Des.Xor(previous, decrypted);

answer += (char)Convert.ToInt32( afterXor, 2 );

previous = bitwise;

}

this.\_logger.Log( "decrypting finished....." );

return answer;

}

public void GenerateIv()

{

this.Iv = "01011100";

}

}

}