

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра РЭС

ОТЧЕТ
по лабораторной работе № 3
по дисциплине «Цифровые и микропроцессорные устройства»
Тема: Изучение принципов отладки программ с использованием
резидентных средств микроконтроллера

Студенты гр. _____

Преподаватель _____

Санкт-Петербург

2023

Цель работы.

Написать и запустить программы в соответствии с требованием пунктов 3.2-3.7

Экспериментальные результаты и их обработка.

1. Отладить и запустить циклическую программу анализа состояния входов порта P3. Произвести в программе инкремент ячейки 30H при наличии «0» на входе P3.3 и декремент при наличии «0» на входе P3.2. Вывести содержимое ячейки 30H на экран в десятичном коде с помощью подпрограмм ACCDEC, CRLF и DSEC. Проверить работу программы, кратковременно подключая P3.3 и P3.2 к шине «Общий». Модифицировать программу, добавив вывод в P3.2 инвертированного состояния входа P3.3. Подключать к шине «Общий» только вход P3.3.

ORG 4000H- адрес, указанный директивой ORG, в который размещается исполняемый машинный код, преобразованный из текстового файла;

ZXC: JB INT0,ARC - в метке ZXC осуществляется вызов подпрограммы, перехода на метку ARC при 1 иначе переход на следующую строку;

INC 30H - инкремент 30 ячейки

ARC: JB INT1,NEKO - в метке ARC осуществляется вызов подпрограммы, перехода на метку NEKO при 1 иначе переход на следующую строку;

DEC 30H- Декремент 30 ячейки;

NEKO:MOV A, 30H- запись в аккумулятор значения 30 ячейки;

CALL ACCDEC - вывод символа из ACC на дисплей в десятичном виде;

CALL CRLF- перевод строки на экране;

CALL DSEC- задержка;

JMP ZXC- переход на метка ZXC

RET- обеспечение выхода в монитор по окончании выполнения программы;

END- директива завершения текста программы.

2. Отладить и запустить циклическую программу вывода в порт P1 «бегущей единицы». Направление сдвига задавать входом P3.2. Управлять частотой сдвига входом P3.5, обнаруживая перепад 1/0. Подключить к P3.5 сигнал генератора макета. Наблюдать осциллограммы на выводах порта P1 и выходе ЦАП1, меняя частоту генератора и сигнал на входе P3.2.

ORG 4000H- адрес, указанный директивой ORG, в который размещается

MOV A,#1- запись 1 в ACC

ZXC: JB P3.5,ZXC - в метке ZXC осуществляется вызов подпрограммы, перехода на метку ZXC при 1 иначе переход на следующую строку;

ARC: JNB P3.5,ARC - в метке ARC осуществляется вызов подпрограммы, перехода на метку ARC при 0 иначе переход на следующую строку;

JNB P3.2,NEKO

RR A- циклический сдвиг вправо;

MOV P1,A – запись в P1 ACC

JMP ZXC- переход на метка ZXC

NEKO:RL A циклический сдвиг влево ;

MOV P1,A запись в P1 ACC;

JMP ZXC- переход на метка ZXC

RET- обеспечение выхода в монитор по окончании выполнения программы;

END- директива завершения текста программы.

3. Составить программу формирования на выходе ЦАП1 частотно-манипулированного сигнала с одинаковой длительностью посылок «0» и «1». Использовать массив TABSIN. Логический «0» передавать одним периодом синусоиды, логическую «1» – двумя. Модулирующую информацию вводить, последовательно опрашивая входы P3.2 ... P3.5.

Соединить один или два входа с шиной «Общий». Наблюдать сигнал на выходе ЦАП1.;

ORG 4000H- адрес, указанный директивой ORG, в который размещается

MOV DPTR,#TABSIN- запись TABSIN в DPTR

MOV R7,#0- запись 0 в R7

ZXC: MOV C,P3.2 - в метке ZXC осуществляется вызов подпрограммы, пересылки в бит переноса бита P3.2

CALL ARC – вызов подпрограммы в метке ARC

MOV C,P3.3 - пересылки в бит переноса бита P3.3

CALL ARC – вызов подпрограммы в метке ARC

MOV C,P3.4 - пересылки в бит переноса бита P3.4

CALL ARC – вызов подпрограммы в метке ARC

MOV C,P3.5 - пересылки в бит переноса бита P3.5

CALL ARC – вызов подпрограммы в метке ARC

JMP ZXC - переход в метку ZXC

ARC: JC НЕКО - в метке ARC осуществляется вызов подпрограммы, перехода на метку НЕКО при 0 флага переноса переход на следующую строку;

GULA: CLR A- сброс ACC

MOVC A,@A+DPTR- пересылка в ACC байта адрес которой равен ACC+DPTR ;

MOV P1,A – запись в P1 ACC

INC DPL- инкремент DPL

INC DPL- инкремент DPL

DJNZ R7,GULA декремент R7 и переход на метку GULA если не 0 иначе следующая команда ;

RET- обеспечение выхода в монитор по окончании выполнения программы;

НЕКО: CLR A- сброс ACC

MOVC A,@A+DPTR- пересылка в ACC байта адрес которой равен ACC+DPTR ;

MOV P1,A – запись в P1 ACC

INC DPL- инкремент DPL

INC B- инкремент B

DJNZ R7, NEKO декремент R7 и переход на метку NEKO если не 0 иначе следующая команда;

RET- обеспечение выхода в монитор по окончании выполнения программы;

END- директива завершения текста программы.

4. Составить программу формирования фазоманипулированного сигнала на выходе ЦАП1. В качестве элементарной посылки сигнала использовать массив TABMOD. Модуляция – через входы P3.2 ... P3.5.

ORG 4000H- адрес, указанный директивой ORG, в который размещается

MOV DPTR,#TABMOD - запись TABSIN в DPTR

MOV R7,#0- запись 0 в R7

ZXC: MOV C,P3.2 - в метке ZXC осуществляется вызов подпрограммы, пересылки в бит переноса бита P3.2

CALL ARC – вызов подпрограммы в метке ARC

MOV C,P3.3 - пересылки в бит переноса бита P3.3

CALL ARC – вызов подпрограммы в метке ARC

MOV C,P3.4 - пересылки в бит переноса бита P3.4

CALL ARC – вызов подпрограммы в метке ARC

MOV C,P3.5 - пересылки в бит переноса бита P3.5

CALL ARC – вызов подпрограммы в метке ARC

JMP ZXC - переход в метку ZXC

ARC: JC NEKO - в метке ARC осуществляется вызов подпрограммы, перехода на метку NEKO при 0 флага переноса переход на следующую строку;

GULA: CLR A- сброс ACC

MOVC A,@A+DPTR- пересылка в ACC байта адрес которой равен ACC+DPTR ;

MOV P1,A – запись в P1 ACC

CPL A – инверсия ACC

INC DPL- инкремент DPL

DJNZ R7,GULA декремент R7 и переход на метку GULA если не 0 иначе следующая команда ;

RET- обеспечение выхода в монитор по окончании выполнения программы;

NEKO: CLR A- сброс ACC

MOVC A,@A+DPTR- пересылка в ACC байта адрес которой равен ACC+DPTR ;

CPL A – инверсия ACC

MOV P1,A – запись в P1 ACC

INC DPL- инкремент DPL

DJNZ R7, NEKO декремент R7 и переход на метку NEKO если не 0 иначе следующая команда;

RET- обеспечение выхода в монитор по окончании выполнения программы;

END- директива завершения текста программы

5. Составить программу формирования амплитудно-манипулированного сигнала с соотношением уровней амплитуд огибающей сигнала 1 : 2. В качестве элементарной посылки сигнала использовать массив TABMOD. Управление модуляцией – через входы P3.2 ... P3.5.

ORG 4000H- адрес, указанный директивой ORG, в который размещается

MOV DPTR,#TABMOD - запись TABMOD в DPTR

MOV R7,#0- запись 0 в R7

ZXC: MOV C,P3.2 - в метке ZXC осуществляется вызов подпрограммы, пересылки в бит переноса бита P3.2

CALL ARC – вызов подпрограммы в метке ARC

MOV C,P3.3 - пересылки в бит переноса бита P3.3

CALL ARC – вызов подпрограммы в метке ARC

MOV C,P3.4 - пересылки в бит переноса бита P3.4

CALL ARC – вызов подпрограммы в метке ARC

MOV C,P3.5 - пересылки в бит переноса бита P3.5

CALL ARC – вызов подпрограммы в метке ARC

JMP ZXC - переход в метку ZXC

ARC: JC NEKO - в метке ARC осуществляется вызов подпрограммы, перехода на метку NEKO при 0 флага переноса переход на следующую строку;

GULA: CLR A- сброс ACC

MOVC A,@A+DPTR- пересылка в ACC байта адрес которой равен ACC+DPTR ;

MOV P1,A – запись в P1 ACC

MOV B,#4 – запись в B 4

DIV AB – деление ACC на B частное в ACC остаток в B

ADD A,#96 – сложение ACC с 96

INC DPL- инкремент DPL

DJNZ R7,GULA декремент R7 и переход на метку GULA если не 0 иначе следующая команда ;

RET- обеспечение выхода в монитор по окончании выполнения программы;

NEKO: CLR A- сброс ACC

MOVC A,@A+DPTR- пересылка в ACC байта адрес которой равен ACC+DPTR ;

MOV P1,A – запись в P1 ACC

DIV AB – деление ACC на B частное в ACC остаток в B

ADD A,#96 – сложение ACC с 96

INC DPL- инкремент DPL

DJNZ R7,GULA

INC DPL- инкремент DPL

DJNZ R7, NEKO декремент R7 и переход на метку NEKO если не 0 иначе следующая команда;

RET- обеспечение выхода в монитор по окончании выполнения программы;

END- директива завершения текста программы

6. Вызвать бесконечном цикле подпрограмму формирования сигнала с линейно-частотной модуляцией (CAL LFM). В ячейку 30H занести коэффициент изменения формирования сигнала (3..9).

ORG 4000H- адрес, указанный директивой ORG, в который размещается

MOV 30H,#3 - запись 3 в 30H

ZXC: CAL LFM – вызов LFM

JMP ZXC – переход на ZXC

RET

END

Вывод.

В ходе лабораторной работы были написаны и запущены программы на языке Ассемблер, осуществляющие требования с пункта 3.2 до 3.7.