

Демонстрационная программа к лабораторной работе

"Исследование каналов ввода-вывода микроконтроллеров семейства MCS-196"

```
; Программа спроектирована для оценочного модуля 80C196KD Development Board
;
; Программа формирует ШИМ-сигналы на 2-х каналах HSO:
;
; - HSO.0 - опорный, с постоянной скважностью Q=2, период опорного сигнала
; определяется результатом аналого-цифрового преобразования канала АСН0 АЦП,
; который подключен к каналу 1 регулируемого источника постоянного напряжения
; (блок "АНАЛОГОВЫЕ ВЫХОДЫ" стенда).
;
; - HSO.1 - ШИМ-модулированный сигнал с периодом HSO.0 и со скважностью,
; определяемой результатом аналого-цифрового преобразования канала АСН1 АЦП,
; который подключен к каналу 1 регулируемого источника постоянного напряжения
; (блок "АНАЛОГОВЫЕ ВЫХОДЫ" стенда).
;
; Результат аналого-цифрового преобразования канала АСН0 АЦП поступает на
; выводы порта P1.
;
; Результаты работы программы контролируют
; по осциллограмме "В", которая соответствует опорному сигналу с выхода HSO.0,
; по осциллограмме "А" - соответствующей ШИМ-сигналу с выхода HSO.1;
; по сигнальному и десятичному индикаторам блока "ЦИФРОВЫЕ ВХОДЫ",
; на которых отображается результат аналого-цифрового преобразования АСН0.
```

```
$include (c:\intel\projbldr\include\80c196kd.inc)
```

```
;----- Регистровый сегмент -----
```

```
rseg at 0020h
```

```
ADres:      dsb 1; Регистр для хранения результата АЦ преобразования
hso0:       dsb 1; Регистр, младший бит которого определяет текущее состояние HSO.0
hso1:       dsb 1; Регистр, младший бит которого определяет текущее состояние HSO.1
shso1:      dsb 1; Временный регистр, использующийся для работы с IOS0
count:      dsw 1; Номер текущего отсчета
acc:        dsb 1; Временный регистр для выполнения арифметических операций
acr:        dsw 1; Регистр для хранения текущего состояния АЦП (ad_result)

count0:     dsw 1; Регистры, определяющие количество отсчетов для
count1:     dsw 1; сигналов на HSO.0 и HSO.1

chan:       dsb 1; Содержит номер канала, для которого будет производиться
              ; АЦ преобразование (#18h для 1-го или #19h для 2-го)
```

;----- Сегмент кода -----

cseg at 0C000h

ld SP,#100

mainproc: ;ПРЕДВАРИТЕЛЬНЫЕ УСТАНОВКИ

andb acc,ioc2,#00111111b; Разрешение HSO и АЦП
ldb wsr,#15
stb acc,ioc2

ld count, #0; Очистка рабочих регистров
ldb hso0, #0
ldb hso1, #0

ldb wsr,#00; Выполнение АЦ преобразования для канала
ldb ad_command, #18h

ld count0, #06000h; Скважность ШИМ (HSO.1) равна
ld count1, #06000h; скважности опорного сигнала (HSO.0)

start: ;ОСНОВНОЙ ЦИКЛ

add count, #1; Инкремент счетчика отсчетов опорного сигнала
cmp count, count0
jne setch1; Если count не равен count0, то переход на setch1

ld count, #0; Если count=count0, то count обнуляется, а hso0 = 1-hso0
ldb acc, #1
sub acc, hso0
ldb hso0, acc

ldb hso1, hso0; hso1 = hso0

ldb chan, #18h; АЦ преобразование производится:
addb chan, hso0; - для первого канала, когда HSO.0 = 0 (chan=#18h)
ldb wsr,#00; - для второго канала, когда HSO.0 = 1 (chan=#19h)
ldb ad_command, chan

setch1: cmp count, count1; Если count count1, то переход на hsowork
jne hsowork

cmpb hso0, #1; Если hso0 # 1, то переход на hsowork
jne hsowork

ldb acc, #1; hso1 = 1 - hso1
subb acc,hso1
ldb hso1, acc

hsowork: ;

УСТАНОВКА ЗНАЧЕНИЙ НА HSO.0 И HSO.1

ldb shso1, hso1;	shso1 = hso1
shlb shso1, #1;	Сдвиг shso1 на 1 бит влево для установки 2-го бита регистра IOS0 в состояние hso1
ldb acc, ios0;	
andb acc, #11111100b;	Очистка 1-го и 2-го бита IOS0 (состояния HSO)
orb acc, hso0;	1-ый бит IOS0 устанавливается в состояние hso0
orb acc, shso1;	2-ой бит IOS0 устанавливается в состояние hso1
ldb wsr, #15	
ldb ios0, acc	
ldb wsr, #0	
ld acp, ad_result;	АЦП завершил преобразование?
and acp, #8;	Если 4-ый бит регистра ad_result установлен в 0
cmp acp, #0;	то преобразование завершено.
jne start;	Если нет, то возвращаемся на начало цикла
ldb ADres, ad_result_hi;	Сохраняем результат АЦ преобразования в регистре ADres
;	
cmpb hso0, #0;	Если hso0 не равно 0, то преобразование было для второго канала, переход на doch1
jne doch1;	
ldbze count0, ADres;	Результат преобразования определяет количество отсчетов для первого канала
;	count0 = ADres*32
shl count0, #5;	
ldb ioport1, ADres;	Выводим результат в порт 1.
sjmp start;	Возвращаемся на начало цикла
doch1: ldbze count1, ADres;	Результат преобразования определяет количество отсчетов для второго канала
;	count1 = ADres*32
shl count1, #5;	
sjmp start;	Возвращаемся на начало цикла
end	

Диаграмма работы программы

