

## АППАРАТУРА ДЛЯ ИССЛЕДОВАНИЙ И МЕТОДЫ ИЗМЕРЕНИЙ

УДК 681.326.34

А.П. Иванов, А.П. Ростов

### МНОГОКАНАЛЬНЫЙ АППАРАТНО-ПРОГРАММНЫЙ СИНХРОННЫЙ ДЕТЕКТОР С УСРЕДНЕНИЕМ РЕЗУЛЬТАТОВ

Разработан аппаратно-программный комплекс для высококачественного многоканального синхронного детектора с усреднением результатов.

Современные экспериментальные установки для изучения атмосферы почти всегда оснащены персональными компьютерами (ПК), которые позволяют следить за ходом эксперимента, регистрировать полученные результаты на те или иные носители цифровой информации или даже полностью его автоматизировать. Для этого сигналы от аналоговых приборов установок преобразуются в цифровую форму многоканальным аналого-цифровым преобразователем (АЦП), установленным в системный блок ПК или в экспериментальную установку [1].

Синхронные детекторы и сглаживающие фильтры широко применяются в различной аппаратуре для экспериментальных исследований атмосферы – это содары, нефелометры, резонансные спектрометры и т.д. Обычно амплитудные детекторы и сглаживающие фильтры для диапазона частот от 1 – 10000 Гц выполняются на электронных компонентах той или иной степени интеграции. Изготовить одноканальный качественный синхронный детектор с фильтром на одну частоту особых сложностей не представляет, но в случае многоканального и много-частотного – это уже целая проблема. Чтобы получить одинаковые амплитудно-фазовые характеристики, приходится долго подбирать электронные компоненты устройства, а для стабилизации их характеристик в зависимости от условий эксплуатации вводить температурную стабилизацию. В целом это получается сложная и громоздкая система.

В настоящее время с появлением у экспериментаторов персональных компьютеров с тактовой частотой более 20 МГц стало возможным программно реализовать в этом частотном диапазоне некоторые аналоговые устройства экспериментальных установок с достаточно высокими и стабильными характеристиками [2].

В качестве примера решения такой задачи ниже приводится описание аппаратно-программного комплекса (АПК), предназначенного для регистрации сигналов от оптико-акустического спектрометра [3, 6].

#### *Характеристики АПК:*

Динамический диапазон	72 дБ
Частотный диапазон	1–10000 Гц
Максимальная амплитуда входных сигналов	+ / -5 В
Диапазон времени усреднения	0,1–1000 мс
Интерфейс	шина IBM-PC/AT

Сигналы переменного тока от трех линейных усилителей датчиков и сигнал задающего генератора преобразуются в цифровой код, а синхронное детектирование и сглаживание делается программно для всех трех каналов в реальном масштабе времени. Синхронизирующим сигналом в этом случае является сигнал задающего генератора.

Структурная схема аппаратной части комплекса представлена на рис. 1. Она состоит из 12-разрядного АЦП среднего быстродействия (время преобразования 5 мкс), четырех ус-

ройств выборки и хранения, 4-канального мультиплексора, регистра данных и состояния, генератора, контроллера всего устройства в стандарте IBM-PC и вторичного источника питания. Все устройство работает в циклическом режиме с периодом 10 мкс.

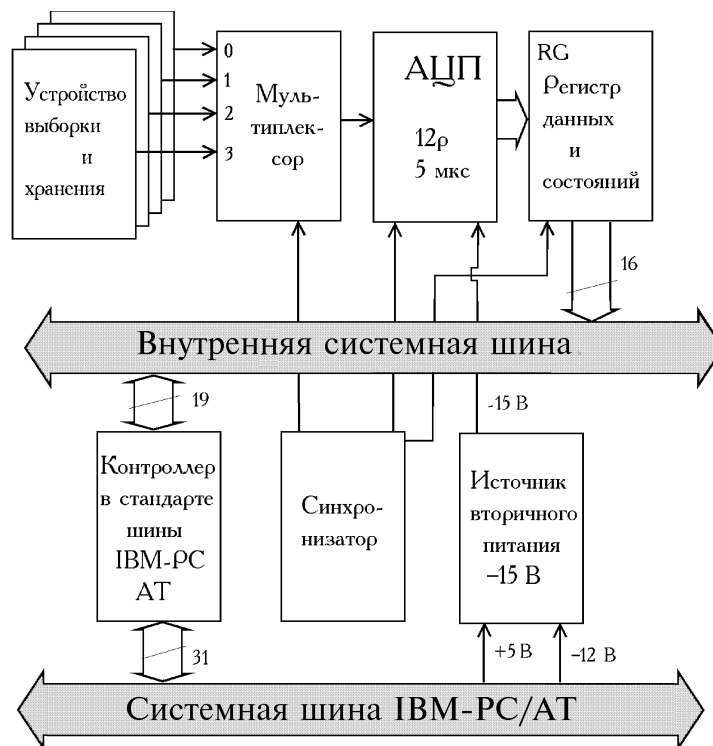


Рис. 1. Структурная схема аппаратной части комплекса

Порядок цикла следующий. Входные сигналы запоминаются на время оцифровки в четырех устройствах выборки и хранения и затем по очереди преобразуются в цифровой код АЦП. Оставшиеся от времени преобразования 5 мкс используются для переключения каналов мультиплексора и подготовки АЦП к следующему запуску, а также для считывания результатов оцифровки в память компьютера.

На рис. 2 показана структура регистра данных и состояний. Для повышения быстродействия в регистр данных введены флаги состояния устройства, которые занимают с 12-го по 15-й разряд. 12, 13, 14-й биты показывают номер канала, который оцифровывается в настоящий момент, а 15-й бит (знаковый) – на признак конца преобразования. Таким образом, удалось минимизировать количество обращений к АЦП комплекса при считывании данных информации и флагов состояния.



Рис. 2. Структура регистра данных и состояний комплекса

Программная часть комплекса написана на языке ассемблера MASM6.1 фирмы Microsoft [4]. Язык ассемблера позволяет оптимально использовать все регистры и время работы центрального процессора, что очень важно для программ реального времени. Имея эту процедуру

ру, пользователь может разрабатывать программу работы всего прибора в целом на языках высокого уровня, используя ее как стандартную внешнюю подпрограмму с конечным набором передаваемых параметров. Это становится особенно актуально при работе под управлением многозадачной операционной системы Windows-95, где доступ к портам ограничен или запрещен. Исходный текст программы, реализующей синхронное детектирование трех процессов со сглаживанием для языка C++ DOS, приведен ниже.

```

;=====
;Функция драйвер ADC4ks для программ на языке C++
;=====
;А. Ростов, Томск. Ver. 1.1 от 5.10.95
;-----
;** Вызов из Си   ADC4ks(k1,k2,k3,k4,N)
;-----
; Где k1, k2, k3 – 12-разрядные протестированные и усредненные
; за N мс значения 1, 2, 3-го каналов,
; а k4 – значение 4-го канала без всяких преобразований.
;-----
; Транслировать с помощью команды ML /c ADC4ks.ASM

.MODEL medium, c ;средняя модель распределения памяти для Си
;Опишем имя процедуры и список передаваемых параметров
ADC4KS   PROC C k1:SWORD, k2:SWORD, k3:SWORD, k4:SWORD, N:SWORD

.CODE           ; Кодовая часть
ADC4KS   PROC   k1:SWORD, k2:SWORD, k3:SWORD, k4:SWORD, N:SWORD

    mov     si, N           ;возьмем время усреднения в мс
    and     si, 3FFFh      ;число мс не должно превышать 16382
    xor     ax, ax         ;установим флаг переноса в 0
    rcl     si, 1          ;умножим на 2

;обнулим переменные
    mov     di, offset cs:chnl1
    mov     cx, 8
    xor     ax, ax

clrvar:  mov     cs:[di], ax
        inc     di
        inc     di
        loop   clrvar      ;цикл обнуления переменных

        mov     dx, 300h ;адрес АЦП
        mov     di, offset cs:chnl1

        cli

        mov     bx, 0      ;будем использовать регистр BX
                        ;для быстрого сложения с переносом

beg:     ;большой цикл
        mov     cx, 10

beg1:    ;малый цикл

beg4:    in      ax, dx
        and     ax, 3000h
        cmp     ax, 3000h ;ждем оцифровку 4-го канала
        jnz    beg4

```

```

m1n:   in    ax,dx
       test  ax,3000h      ;ждем оцифровку 1-го канала
       jnz  m1n
m12n:  in    ax,dx
       or   ax,ax
       jns  m12n          ;ждем начало оцифровки 1-го канала

m11n:  in    ax,dx
       and  ax,8FFFh      ;выделим информационные разряды
       js   m11n          ;ждем конца оцифровки 1-го канала
       sub  ax,2048        ;переведем в дополнительный код
       not  ax             ;инвертируем для получения модуля
m14n:  add  cs:[di+0],ax   ;добавим полученное число к сумме
       adc  cs:[di+2],bx   ;если был перенос, то увеличим ст.слово

m22n:  in    ax,dx
       or   ax,ax
       jns  m22n          ;ждем начало оцифровки 2-го канала
m2n:   in    ax,dx
       and  ax,8FFFh      ;выделим информационные разряды
       js   m2n           ;ждем конец оцифровки 2-го канала
       sub  ax,2048        ;переведем в дополнительный код
       jnc  m24n
m24n:  not  ax             ;инвертируем для получения модуля
       add  cs:[di+4],ax   ;добавим полученное число к сумме
       adc  cs:[di+6],bx   ;если был перенос, то увеличим ст.слово

m32n:  in    ax,dx
       or   ax,ax
       jns  m32n          ;ждем начало оцифровки 3-го канала
m3n:   in    ax,dx
       and  ax,8FFFh      ;выделим информационные разряды
       js   m3n           ;ждем конец оцифровки 3-го канала
       sub  ax,2048        ;переведем в дополнительный код
       jnc  m34n
m34n:  not  ax             ;инвертируем для получения модуля
       add  cs:[di+8],ax   ;добавим полученное число к сумме
       adc  cs:[di+10],bx  ;если был перенос, то увеличим ст.слово

;4-й канал не детектируем
m42n:  in    ax,dx
       or   ax,ax
       jns  m42n          ;ждем начало оцифровки 4-го канала
m4n:   in    ax,dx
       and  ax,8FFFh      ;выделим информационные разряды
       js   m4n           ;ждем конец оцифровки 4-го канала
       sub  ax,2048        ;переведем в дополнительный код
       mov  cs:[di+12],ax  ;запишем значение 4-го канала в служ. ;переменной

       loop beg1          ;конец малого цикла
       dec  si
       jnz  beg           ;конец большого цикла

       sti

;вычислим среднее значение протектированных сигналов
       mov  bx,[bp+6]
       mov  si,[bx]       ;возьмем число усреднений в мс

```

```

    and    si,3FFFh;число мс не должно превышать 16382
    xor    ax,ax      ;установим флаг переноса в 0
    rcl   si,1       ;умножим на 2

;1-й канал
    mov   bx,10
    mov   dx,cs:chnh1
    mov   ax,cs:chnl1
    div   si          ;сначала разделим на 10
    cwd   ;преобразуем 2-байтовое число в 4-байтовое
    div   bx          ;поделим на число заданных усреднений
    mov   k1,ax       ;занесем результат в переменную k1

;2-й канал
    mov   bx,10
    mov   dx,cs:chnh2
    mov   ax,cs:chnl2
    div   si
    cwd
    div   bx
    mov   k2,ax

;3-й канал
    mov   bx,10
    mov   dx,cs:chnh3
    mov   ax,cs:chnl3
    div   si
    cwd
    div   bx
    mov   k3,ax

;4-й канал берем прямое значение последнего измерения
    mov   ax,cs:chnl4
    mov   k4,ax

    ret 10           ;возврат из процедуры со смещением стека на 10
                    ;байт

;Служебные переменные будут находиться в составе процедуры
chnl1  dw    0      ;1-й канал младшее слово
chnh1  dw    0      ;1-й канал старшее слово
chnl2  dw    0      ;2-й канал младшее слово
chnh2  dw    0      ;2-й канал старшее слово
chnl3  dw    0      ;3-й канал младшее слово
chnh3  dw    0      ;3-й канал старшее слово
chnl4  dw    0      ;4-й канал младшее слово
chnh4  dw    0      ;4-й канал старшее слово

Сопуг  BYTE "V1.1 A.Rostov Tomsk-95"

ADC4KS  ENDP      ;конец процедуры
        END       ;конец программы

```

Из текста программы видно, что обращения к оперативной памяти сведены к минимуму. Большинство арифметических и логических операций выполняются на уровне регистров процессора. На время выполнения операции накопления данных запрещены все аппаратные прерывания процессора. Такое построение программы позволило реализовать процесс приема и детектирования сигналов с вышеуказанными характеристиками на 12-мегагерцевом компьютере IBM-PC/286. Отставание системных часов в процессе работы этой программы устраняет-

ся за счет считывания кода времени из энергонезависимых часов компьютера [5] при окончании процесса набора данных.

Приведенная выше программа легко может быть адаптирована для других языков программирования. Нужно только изменить протокол передачи параметров. Например, для включения ее в состав динамически подключаемой библиотеки (\*.dll) для среды Windows 3.\*, нужно установить «паскалевское» соглашение о связи и написать файл определений (Adc4ks.def), откомпилировать командой ml /c /w3 Adc4ks.asm, а затем скомпоновать командой Link Adc4ks MyLib.dll, LibW.lib mncrtdw.lib, Adc4ks.def.

Разработка, описанная в этой статье, выполнена при частичной финансовой поддержке Международного научного фонда и Российского правительства, грант J29100.

1. Ростов А. П. // Оптика атмосферы. 1988. Т. 1. N 3. С. 125–126.
2. Финогенов К. Г. Программирование измерительных систем реального времени. М.: Энергоатомиздат, 1990. 256 с.
3. Антипов А. Б., Капитанов В. А., Пономарев Ю. Н., Сапожникова В. А. Оптико-акустический метод в лазерной спектроскопии молекулярных газов. Новосибирск: Наука, 1984. 128 с.
4. Microsoft Corporation. Microsoft MASM Version 6.1 for MS-DOS and Windows Operating Systems, 1992. 454 p.
5. Джордейн Р. Справочник программиста персональных компьютеров типа IBM PC, XT и AT/ Пер. с англ. М.: Финансы и статистика, 1992. 544 с.
6. Капитанов В. А., Куликов Г. Э., Тырышкин В. И. // Оптика атмосферы и океана. 1995. Т. 8. N 9. С. 1411–1414.

Институт оптики атмосферы СО РАН,  
Томск

Поступила в редакцию  
13 декабря 1995 г.

A. P. Ivanov, A. P. Rostov. **Multichannel Synchronous Detector with Results Averaging.**

Software-hardware complex for high-quality multichannel synchronous detector with averaging results is developed.