

НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЕ УЧРЕЖДЕНИЕ
ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ им. Г.И. Будкера СО РАН

В.Р. Мамкин

ИНТЕЛЛЕКТУАЛЬНЫЙ КОНТРОЛЛЕР
В СТАНДАРТЕ VME

ИЯФ 2002-33

НОВОСИБИРСК
2002

Интеллектуальный контроллер в стандарте VME

В.Р. Мамкин

Институт ядерной физики им. Г.И.Будкера СО РАН
630090, Новосибирск, СО РАН

В работе приведено техническое описание контроллера на базе процессора Motorola. Контроллер выполнен в стандарте VME. Приводится описание структуры, карты памяти и регистров. Рассматривается системное программное обеспечение, разработанное для контроллера.

The intelligent controller in VME standard

V.Mamkin

Budker Institute of Nuclear Physics
630090, Novosibirsk, SB RAS

There is technical description of VME controller, based on Motorola processor, provided in this paper. The controller is developed in VME standard. Architecture, memory map and registers are considered. System software, developed for the controller, is described.

© Институт ядерной физики им. Г.И.Будкера СО РАН

Введение

В настоящее время в ИЯФ СО РАН разработан и используется контроллер в стандарте VME, базирующийся на микропроцессоре MC68360 фирмы Motorola. При разработке контроллера ставились задачи добиться небольшой себестоимости и совместимости с современными операционными системами. Дополнительно к контроллеру был разработан крейт VME на 7 слотов.

1. Крейт VME

В таблице приведены основные характеристики крейта.

| | |
|----------------------------------|---|
| Конструктивное исполнение | Евромеханика, 6 U Schroff, Europack Lab |
| Механический стандарт | IEC 60297-3/DIN 41494, Part 5. |
| Габариты | Ширина 437 мм (без боковых кронштейнов) Высота 264 мм Глубина 225 мм |
| Материал корпуса | Алюминиевый каркас, Передние и задние панели из стали |
| Количество VME слотов | 7 |
| Магистраль VME J1 | Изготовитель - фирма Schroff, с автоматическими дейза-цепочками |
| Магистраль VME J2 | Изготовитель – ИЯФ |
| Напряжения питания на магистрали | +5V, +12V, -12V, +5Vstby |

Для питания крейта используется стандартный источник АТХ фирмы Powerman, модель FSP145-60SP.

Входные характеристики источника:

Voltage: 90-132 VAC or 180-264 VAC switchable;

Frequency: 47Hz to 63Hz;

Input current: 4.0A(RMS) for 115 VAC, 2.0A(RMS) for 230 VAC.

Выходные характеристики:

| Напряжение | Макс. Ток | Погрешность | Пульсации |
|------------|-----------|-------------|--------------|
| +5 V | 18 A | ±5% | 50 mv P – P |
| +12 V | 4.2 A | ±5% | 100 mv P – P |
| -12 V | 0.3 A | ±10% | 120 mv P – P |
| +5 Vstby | 0.8 A | ±5% | 50 mv P – P |

Включение питания крейта осуществляется двумя выключателями – на передней панели крейта и на источнике питания. Напряжение +5 Vstby сохраняется только при использовании выключателя на передней панели крейта.

2. Контроллер шины VME

2.1. Назначение

Контроллер BVME-1 разработан для эксплуатации в составе магистрально-модульных систем стандарта VME. Контроллер выполняет функции системного контроллера и VME-мастера. Наличие встроенного Ethernet интерфейса позволяет легко интегрировать VME систему в состав современного компьютерного комплекса. Программная поддержка включает наиболее популярные операционные системы VxWorks и Linux.

Возможные области применений:

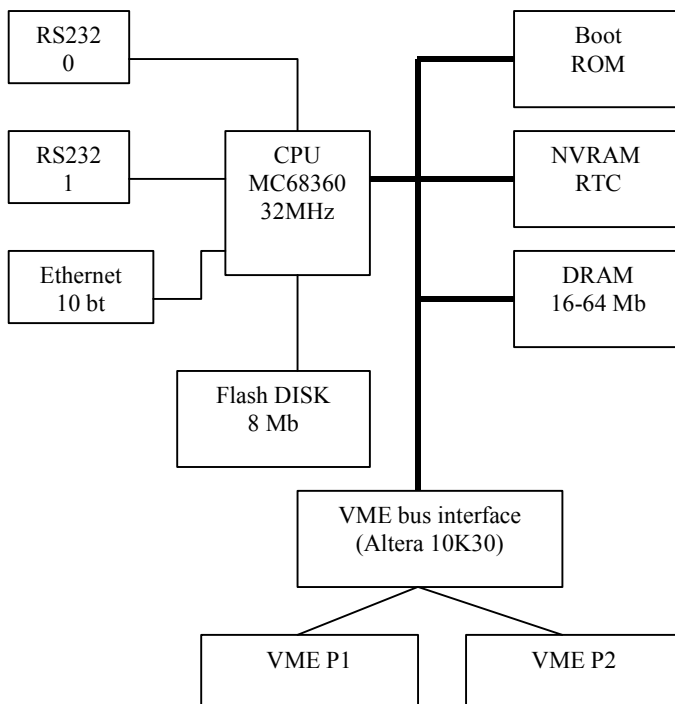
- Управление промышленными установками;
- Системы сбора данных;
- Контроллер ввода-вывода.

2.2. Основные характеристики контроллера

| | |
|--------------------|--|
| Процессор | Motorola MC68EN360 |
| Тактовая частота | 32 МГц |
| ОЗУ | 32 bit, 8 или 16 Мбайт |
| ПЗУ | 128 Кбайт Bootrom |
| Flash | 8 Мбайт флэш - диск |
| Порты ввода-вывода | 2 интерфейса RS-232 |
| | 1 Ethernet 10base-T |
| Прочее | Часы реального времени и 2 Кбайт памяти с бесперебойным питанием |
| VME – шина | |

| | |
|----------------|---|
| Типы адресации | A16, A24, A32 Программируемый адресный модификатор |
| Форматы данных | D8, D16, D32 |
| VME запросчик | Программируемая линия запроса, освобождение по запросу |
| VME арбитр | Фиксированный и круговой приоритеты |
| Прерывания | IRQ1..IRQ7 на VME шине Вызов прерывания по AC_FAIL |

2.3. Блок-схема контроллера



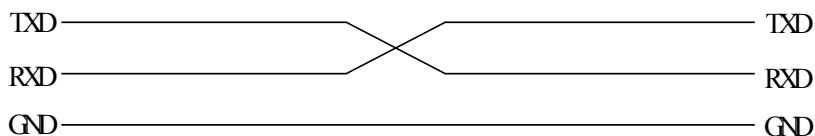
2.4. Подключение контроллера

На передней панели контроллера расположены:

1. Кнопка сброса (Reset). При нажатии кнопки, если установлен режим системного контроллера, сбрасывается вся VME система. Если BIVME-1 не является системным, сбрасывается только локальный процессор.
2. Индикатор локальной сети (LAN led). Загорается при передаче пакета сетевым контроллером.
3. Индикатор VME шины. Загорается при обращении процессора к VME шине.
4. Разъемы последовательных портов (com0, com1). Назначение выводов:

| конт | цепь | Тип |
|------|------|-------|
| 1 | CD | Вход |
| 2 | RXD | Вход |
| 3 | TXD | Выход |
| 4 | DTR | Выход |
| 5 | GND | Земля |
| 6 | – | – |
| 7 | RTS | Выход |
| 8 | CTS | Выход |
| 9 | – | – |

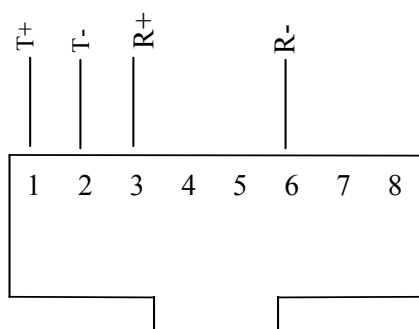
Порт com0 является консольным, т.е. используется по умолчанию загрузочным монитором и операционными системами для ввода-вывода. Для подключения консоли используется перекрещенный кабель:



На компьютере обслуживания запускается терминальная программа с параметрами: 38400/8bit/1S/NP.

5. Гнездо для подключения локальной сети (LAN socket). Через этот разъем контроллер подключается стандартным кабелем к сетевому концентратору.

Разъем VME шины P2 используется при 32-х разрядном доступе к данным и 32-х разрядном адресе. На некоторых кросс-платах VME этот режим может не поддерживаться, и разъем P2 может быть не задействован.



Переключки (джамперы) устанавливаются при наладке контроллера и их положение не должно изменяться.

2.5. Загрузочный монитор

2.5.1. Общие положения

После включения питания или перезапуска контроллера управление всегда получает загрузочный монитор. В зависимости от установленных параметров, монитор сразу после запуска может начать получение сетевого адреса (BOOTP протокол), загрузку файла в динамическую память контроллера, выполнить переход по заранее введенному адресу или перейти в командный режим. Командный режим используется для конфигурации монитора и выполнения определенных сервисных функций. В этом режиме монитор выполняет вводимые пользователем через консоль команды и выводит на консоль результат их выполнения. Набор команд зависит от текущего меню. Содержимое меню можно посмотреть нажатием в терминальной программе клавиш <h>, <Enter>.

Монитор имеет следующие возможности:

- Получение сетевых параметров по протоколу BOOTP;
- Загрузка файлов по протоколу TFTP из внешнего сервера;
- Загрузка файлов из флэш-диска;
- Выполнение сервисных функций по команде пользователя (BOOTP запрос, ARP запрос, ICMP echo);
- Сохранение параметров загрузки в NVRAM памяти;
- Отображение и установка времени в RTC;
- Копирование, заполнение, тестирование памяти.

В энергонезависимой памяти параметры монитора занимают 512 байт от начала окна (см. карту памяти). Часть из них может использоваться драйверами операционной системы. В частности, первые 6 байт параметров занимает адрес Ethernet контроллера, следующие 256 байт - командная строка.

2.5.2. Варианты загрузки

Основная задача монитора - загрузка и запуск операционной системы. Источников загрузки (носителей загружаемого файла) может быть три: сетевой TFTP сервер, файловая система флэш-диска, неформатированный флэш-диск.

TFTP протокол разработан для загрузки бездисковых устройств. Это транспортный протокол, базирующийся на UDP/IP. Обычно после включения, бездисковое устройство получает по BOOTP запросу сетевые настройки, в том числе адрес TFTP сервера. После этого TFTP серверу посылается запрос на загрузку файла. Если файл найден, начинается копирование его в память устройства (в данном случае контроллера). TFTP сервером может служить любая рабочая станция, подключенная к сети, с запущенной программой TFTP сервера. Для Linux, например, такая программа входит в состав стандартных утилит.

Флэш-диск представляет собой область перепрограммируемой (флэш) памяти, со специально организованными данными. При использовании ОС Linux флэш-диск может быть отформатирован (в формате ext2) средствами этой системы. Монитор при этом имеет возможность только читать файловую систему. Форматирование и запись файлов может быть выполнены только под управлением ОС Linux. Если файловая система не используется, то флэш-диск организован как линейный буфер памяти. В зависимости от того, какой вариант задействован, монитор загружает из флэш-диска либо файл, либо буфер заданной длины. Тип носителя задается параметром bootsrc монитора, имя загружаемого файла - параметром file (этот параметр игнорируется при загрузке из неформатированного флэш-диска), длина загружаемого блока данных - параметром bootlen (этот параметр игнорируется при загрузке из TFTP сервера или из форматированного флэш-диска). Сама загрузка инициируется двумя способами - командой load и при рестарте монитора, если установлен флаг Auto load.

2.5.3. Командная строка

Командная строка (КС) служит для передачи данных от загрузочного монитора к операционной системе. КС находится в энергонезависимой памяти контроллера и может быть скопирована монитором в любое место основной динамической памяти. Копирование КС в динамическую память производится, если установлен флаг "Copy bootstring to RAM". При копировании к тексту КС, введенному пользователем, автоматически добавляются сетевые параметры. Это позволяет операционной системе использовать результаты BOOTP запроса, выполненного монитором. Формат этих параметров следующий: e=<adrs:mask> h=<adrs> g=<adrs>, где adrs – IP адрес в текстовом виде, например, 193.1.2.3, mask – сетевая маска, обычно fffff00. Параметр e передает IP адрес сетевого интерфейса контроллера, параметр h передает IP адрес TFTP сервера, g передает IP адрес шлюза.

КС редактируется командой `bootstr`. Содержимое строки зависит от загружаемой операционной системы.

Например, для загрузки Linux КС должна иметь вид:

```
Arg!root=/dev/vfdisk rw,
```

если в качестве файловой системы используется флэш-диск;

```
Arg!root=/dev/rom0,
```

если в качестве файловой системы используется rom-диск.

При загрузке vxWorks, КС обычно имеет вид:

```
es(0,0)host:/home u=vxworks pw=vxworks111 f=0x26 tn=vme .
```

Более подробно содержимое командной строки рассматривается в руководствах по соответствующим операционным системам.

2.5.4. Редактирование параметров

Переход в меню редактирования параметров происходит по команде `<opt>`.

Доступны следующие параметры:

myip – ввод IP адреса контроллера. Это поле может заполняться автоматически после выполнения BOOTP запроса.

servip – ввод IP адреса сервера, с которого будет загружаться файл по TFTP протоколу.

gwip – ввод IP адреса сетевого шлюза.

mask – ввод сетевой маски.

file – имя файла, загружаемого по команде `load`.

loadptr – адрес области памяти, куда будет загружаться файл.

jumpptr – адрес передачи управления по команде `go`.

bootstr – содержание командной строки, которая передается операционной системе.

bootptr – адрес, куда копируется командная строка перед запуском ОС.

bootsrc – код носителя загружаемого файла

0 – загружать по сети от TFTP сервера,

1 – загружать из флэш-диска (файловая система `ext2`),

2 – загружать из неформатированного флэш-диска.

bootlen – длина области памяти, загружаемой командой `load`. Учитывается, если `bootsrc = 2`.

list – выводит значения всех параметров.

flags – переход в меню установки флагов.

Доступны следующие флаги:

- "verbose mode" – при установке в "on" монитор выводит больше текстовых сообщений при работе;
- "TFTP standalone" – при установке в "on" параметр servip не будет меняться при получении сетевых параметров по BOOTP. Таким образом, адрес TFTP сервера не будет зависеть от параметров, полученных по BOOTP;
- "Auto bootp" – выполнять BOOTP запрос при перезапуске;
- "Auto load" – выполнять загрузку файла при перезапуске (аналогично команде load);
- "Auto jump" – выполнять передачу управления при перезапуске (аналогично команде go);
- "Copy bootstring to RAM" – копировать командную строку в оперативную память при перезапуске;
- "Watchdog" – включить сторожевой таймер;
- "Local bus timer" – включить обработку ошибок шины процессором. Если этот параметр "off", ошибки шины обрабатываются микросхемой VME интерфейса (предпочтительно).

Примечания.

Если одновременно установлены флаги "Auto bootp", "Auto load", "Auto jump", "Copy bootstring to RAM", то после перезапуска очередность операций следующая: выполняется BOOTP запрос, загружается файл, копируется командная строка, передается управление.

update - сохранить все параметры (включая флаги).

2.5.5. Сервисные команды

pings – монитор переходит в режим ICMP эхо сервера и отвечает на ping запросы от других станций в сети. Режим служит для проверки соединения локальной сети. Перед выполнением команды должны быть установлены сетевые настройки (IP адрес, маска, шлюз).

bootp – монитор запрашивает сетевые параметры у BOOTP сервера.

arp – монитор делает ARP запрос и получает Ethernet адрес сервера по его IP адресу. Предварительно должен быть введен IP адрес сервера (параметр servip).

load – монитор загружает файл в оперативную память контроллера. Источник загрузки (TFTP сервер или флэш-диск) задается параметром bootsrc. Адрес памяти задается параметром loadptr.

go – монитор передает управление по адресу.

rtc – монитор выводит показания часов реального времени.

- rtcset – установка часов реального времени.
- mdump – распечатка содержимого памяти.
- mfill – заполнение области памяти кодом.
- mtest – тестирование области памяти, содержимое при этом теряется.
- fdump – распечатка содержимого флэш памяти.
- fdprog – программирование флэш памяти. В качестве источника данных обычно указывается участок оперативной памяти контроллера, предварительно загруженный по сети.

2.6. Карта памяти контроллера

| Адреса | Устройство |
|----------------------|-------------------------------|
| 00000000..007FFFFFFF | Модуль динамической памяти |
| 80000000..BFFFFFFF | VME шина, адресация A32 |
| F0000000..F0FFFFFFF | VME шина, адресация A24 |
| F8000000..F801FFFF | Boot rom |
| FE000000..FE001FFF | Блок регистров процессора |
| FFF40000..FFF4FFFF | Блок регистров VME-интерфейса |
| FFFC0000..FFFC07FF | NVRAM, RTC |
| FFFF0000..FFFFFFFF | VME шина, адресация A16 |

Большинство ресурсов контроллера отображается в 32-битное адресное пространство. Хотя прямое обращение к областям памяти не защищено аппаратно, рекомендуется для управления ресурсами использовать драйверы операционной системы.

2.7. VME интерфейс

2.7.1. Общее описание

VME интерфейс выполняет следующие функции :

- Трансляция циклов от локальной шины процессора на VME шину в соответствии с картой адресов и форматом данных;
- Арбитраж VME : запрос и освобождение шины, системный арбитр (в режиме системного контроллера);
- Обработка прерываний: передача прерывания от VME на локальный процессор в соответствии с приоритетами и маской прерывания, выдача циклов подтверждения прерывания.

VIVME-1 может работать как в режиме системного контроллера, так и совместно с другим системным контроллером. Режим системного контроллера отличается наличием следующих функций:

- Включен арбитр шины;
- Включена задержка на дейза-цепочке подтверждения прерываний;
- Контроллер является источником тактовых импульсов по линии SYSCLK;
- Локальный сброс контроллера приводит к системному сбросу на VME магистрали.

VME интерфейс состоит из следующих функциональных модулей:

- Адресный декодер;
- VME запросчик;
- VME арбитр;
- Обработчик прерываний.

2.7.2. Управляющие регистры VME – интерфейса

VME интерфейс имеет следующие регистры:

| Адрес | Название | Тип |
|----------|----------|-----|
| FFF40000 | Reg0 | RW |
| FFF40004 | Reg1 | RW |
| FFF40008 | Reg2 | R |
| FFF40104 | Inth1 | R |
| FFF40108 | Inth2 | R |
| FFF4010C | Inth3 | R |
| FFF40110 | Inth4 | R |
| FFF40114 | Inth5 | R |
| FFF40118 | Inth6 | R |
| FFF4011C | Inth7 | R |

Примечания.

Операции со всеми регистрами должны осуществляться по 32-битному доступу.

После инициализации системы значения битов в записываемых регистрах = 0.

Описание Reg0:

| Биты | Назначение |
|------|--|
| 3..0 | код адресного модификатора, биты AM3..AM0 |
| 4 | код адресного модификатора, биты AM5..AM4 Reg0.4=0 AM5, AM4 зависят от типа адресации: [AM5,AM4] = [0,0] для A32 [AM5,AM4] = [1,0] для A16 [AM5,AM4] = [1,1] для A24 Reg0.4=1 всегда [AM5,AM4] = [0,1] (пользовательский адресный модификатор) |
| 6..5 | Код линии запроса шины 0..3 |
| 7 | режим системного контроллера Reg0.7=0 выключен Reg0.7=1 включен |
| 8 | тип арбитра Reg0.8=0 PRI (приоритетный арбитр) Reg0.8=1 RRS (круговой приоритет) |

Описание Reg1:

| Биты | Назначение |
|------|--|
| 6..0 | маска прерываний, бит 0 соответствует линии 1, бит 6 соответствует линии 7 =0 запрос маскирован =1 запрос разрешен |
| 7 | маска для AC_FAIL линии Reg1.7=0 сигнал AC_FAIL игнорируется Reg1.7=1 сигнал AC_FAIL вызывает прерывание локального процессора |
| 8 | управление сигналом SYS_FAIL Reg1.8=0 не активен (лог.1) Reg1.8=1 активен (лог.0) |

Описание Reg2:

| Биты | Назначение |
|------|--|
| 2..0 | текущий уровень прерывания. Выбирается наиболее приоритетный из немаскированных уровней. Нулевое значение этого поля соответствует прерыванию по сигналу AC_FAIL |
| 3 | состояние сигнала SYS_FAIL |
| 4 | состояние сигнала AC_FAIL |

Чтение регистров InthX приводит к появлению на VME шине цикла подтверждения прерывания для соответствующего уровня. При этом биты 7..0 прочитанных данных будут содержать вектор прерывания, выставленный на шину периферийным модулем. Чтение регистров InthX происходит в обработчике прерывания.

2.7.3. Адресный декодер

После захвата VME шины посредством процедуры арбитража, VME контроллер может читать и писать данные на VME шину. Для этого локальный процессор должен обратиться в одно из адресных пространств на локальной шине, декодируемых VME-интерфейсом (см. карту памяти контроллера). При этом адреса на локальной шине будут однозначно соответствовать адресам на VME шине:

| Локальный адрес | VME адрес | Тип адресации |
|--------------------|--------------------|---------------|
| FFFF0000..FFFFFFFF | 0000..FFFF | A16 |
| F0000000..F0FFFFFF | 0000000..FFFFFF | A24 |
| 80000000..BFFFFFFF | 00000000..3FFFFFFF | A32 |

Длительность цикла чтения или записи данных в периферийный модуль VME зависит от времени отклика конкретного модуля, но ограничивается 64 мкс. При длительности VME цикла более 64 мкс, контроллер выставит на шине сигнал BERR и локальный процессор получит прерывание “ошибка шины”.

2.7.4. VME – запросчик

VME запросчик производит процедуры захвата и освобождения шины. Захват шины инициируется после обращения локального процессора к VME шине, если контроллер не является мастером. На время процедуры арбитража локальный процессор будет находиться в состоянии ожидания. Время ожидания ограничено и составляет 64 мкс. При превышении этого времени процессор выходит из состояния ожидания и получает прерывание “ошибка шины”. Сигнал BERR на VME шине при этом не выставляется. Если в течение 64 мкс шину захватить не удалось, запросчик продолжает удерживать запрос шины до ее получения, хотя локальный процессор вышел из состояния ожидания.

Линия запроса программируется битами Reg0[6,5]. Например, Reg0[6,5] = [1,0] соответствует запросу по линии 2. Наиболее приоритетная линия запроса шины имеет номер 3.

Стратегия освобождения шины – “освобождение по запросу”, т.е. запросчик освобождает VME шину, если обнаруживает запрос от другого контроллера. Шина освобождается также, если обнаруживается сигнал BLCR.

2.7.5. VME – арбитр

Функция арбитра включена, когда BIVME-1 является системным контроллером, т.е. установлен в слот 0 и бит Reg0.7 = 1. В этом случае арбитр начинает предоставлять VME шину в соответствии со стратегией предоставления шины, установленной битом Reg0.8.

Reg0.8 = 0. PRI арбитр. В случае обнаружения запроса одновременно по нескольким линиям, шина предоставляется контроллеру, который выставил запрос по наиболее приоритетной линии. Наивысший приоритет имеет линия 3, самый низкий приоритет у линии 0.

Reg0.8 = 1. RRS арбитр. Шина предоставляется случайным образом любому контроллеру, который выставил запрос по линии, отличной от уровня запроса текущего мастера. Опрос линий при этом арбитр производит по круговой схеме.

В любом случае, если арбитру требуется передать шину другому контроллеру, он выставляет сигнал BCLR и ожидает освобождение шины текущим мастером. Ожидание освобождения не ограничено по времени.

2.7.6. Обработчик прерываний

Для обработки прерывания от VME шины программное обеспечение должно установить бит маски прерывания в регистре Reg1. Из всех разрешенных и активных прерываний VME интерфейсом выбирается наиболее приоритетное и устанавливается запрос прерывания локальному процессору. Уровень запроса прерывания от VME шины фиксированный и равен 5. Если в локальном процессоре разрешена обработка данного уровня прерывания, он переходит в подпрограмму обработки прерывания, где локальный процессор должен провести цикл подтверждения прерывания путем чтения регистра inthX. Наибольший приоритет имеет прерывание по сигналу AC_FAIL, наименьший – прерывание по линии IRQ1.

Более подробно, последовательность событий выгладит так:

- Установка нужной маски прерывания;
- Вызов подпрограммы обработки прерывания (при получении соответствующего запроса по VME шине).

В подпрограмме обработки:

- Читается уровень вызвавшего прерывания `int_level` в регистра `Reg2[2..0]`;
- Если `int_level = 0`, программа выполняет аварийные действия, связанные со сбоем питания, иначе переходит к следующему пункту;
- Читается регистр `inthX`, где `X=int_level`. Это приводит к появлению на VME шине цикла подтверждения прерывания для уровня `int_level`. После проведения цикла подтверждения прерывания периферийный модуль сбрасывает свой запрос. Прочитанные данные будут содержать вектор прерывания, выставленный периферийным модулем.

2.8. Драйверы

Операционная система управляет ресурсами контроллера через специальные программные компоненты – драйверы. Функционирование драйверов стандартных устройств контроллера (консоли, таймера, сетевого интерфейса) “закрыто” вызовами C библиотеки и описано в документации на операционную систему. Например, использование консоли для ввода-вывода сводится к использованию в прикладной программе функций `printf`, `scanf`; сетевой интерфейс доступен через обращения к TCP/IP сокетам и т.д.

Обращения прикладной программы к драйверу VME шины специфичны для данного контроллера и требуют специального описания.

2.8.1. Драйвер VME шины в ОС Linux

Драйвер VME шины выполняет следующие функции: инициализация регистров VME интерфейса; проведение операций чтения и записи на шине, выдача результата приложению; обработка прерываний VME шины; обработка сигнала `AC_FAIL`.

В ОС Linux каждому драйверу соответствует вход в файловую систему. Для VME драйвера это специальные файлы `/dev/vmeiN` (где `N=1..7`) и 8 специальных файлов `/dev/vmeN` (где `N=0..7`). Файлы `vmei` служат для управления системой прерываний, файлы `vme` служат для обращения к VME шине. Чтобы провести операцию чтения или записи, приложение должно функцией `open` открыть файл устройства. Так как прикладных процессов, работающих с VME шиной, может быть несколько, то, для избежания конфликтов, каждый из них должен работать только со своим специальным файлом `/dev/vmeN`. Например, приложение 0 открывает и работает с `/dev/vme2`, приложение 1 открывает и работает с `/dev/vme0`. Для каждого приложения драйвер хранит свою копию размера данных, текущего адреса на VME шине, флага инкрементирования адреса.

После открытия файла устройства, приложение с помощью вызовов `ioctl` должно установить параметры доступа к шине. Непосредственно ввод-вывод на VME шину осуществляется функциями `read` и `write`. Функции `read` и `write` должны вызываться с длиной буфера, кратной размеру данных, установленному `ioctl`

VME_SETSIZE, т.е. 1,2 или 4 байтам. Отрицательное значение, возвращенное read или write говорит об ошибке на VME шине.

IOCTL вызовы

Ioctl вызовы определены в файле vme_ioctl.h

- VME_SETSIZE – установить размер данных. Аргумент ioctl вызова при этом может быть установлен в один из трех вариантов – DATA8, DATA16, DATA32;
- VME_SETADDR – установить адрес на VME шине, по которому будет проведена операция чтения или записи;
- VME_INCREMENT – установить флаг инкрементирования адреса. Если аргумент ioctl установлен в 1, после каждой операции чтения или записи адрес на VME шине будет инкрементироваться (на 1,2 или 4 байта, в зависимости от размерности данных);
- VME_AMSET – установить код адресного модификатора. Аргумент должен содержать код адресного модификатора VME. Формат - см. описание регистра 0 VME интерфейса;
- VME_BUSLEVEL – установить уровень запроса на захват шины. Аргумент может быть в диапазоне 0..3. См. также описание VME запросчика;
- VME_SYSTEM – установить режим системного контроллера. Если аргумент =1, VME интерфейс будет работать в режиме системного контроллера;
- VME_INTHMASK_W – установить маску прерываний. См. описание регистра 1 VME интерфейса;
- VME_INTHMASK_R – прочитать маску прерываний. См. описание регистра 1 VME интерфейса;
- VME_FAIL_R – прочитать состояние сигналов SYS_FAIL и AC_FAIL. После выполнения ioctl вызова аргумент будет содержать в бите 0 состояние сигнала AC_FAIL, в бите 1 – состояние сигнала SYS_FAIL;
- VME_FAILHANDLER – установить процесс-обработчик сигнала AC_FAIL. Аргумент игнорируется. Процессу, предварительно вызвавшему данный ioctl, драйвер посылает программный сигнал SIG_USR1, в случае обнаружения на VME шине сигнала AC_FAIL.

Пример:

```
int arg, fd;
char buf[8];

fd = open("/dev/vme0", O_RDWR | O_EXCL);
arg = 0x1500;
ioctl(fd, VME_SETADDR, &arg);
read(fd, buf, 8);
close(fd).
```

Приведенный код установит адрес текущей операции равным 0x1500, после чего произойдет чтение 8 байтов по VME шине.

Обработка прерываний

Для обработки прерываний используются специальные файлы /dev/vmei1 .. /dev/vmei7. Каждый из этих файлов соответствует линии прерывания на VME шине. Чтобы перехватить прерывание, приложение должно открыть нужный файл /dev/vmei и вызвать функцию select. При этом приложение перейдет в режим ожидания, который кончится либо по истечении определенного времени, либо после получения аппаратного прерывания. Чтобы прочитать вектор прерывания, выставленный на VME шину периферийным модулем, после завершения функции select, приложение должно вызвать функцию read. Прочитанный байт будет являться вектором .

Описанный механизм позволяет на одной линии прерывания обслуживать несколько устройств, при условии, что каждое из них имеет свой вектор прерывания. Если приложение по какой-либо причине занято, событие буферизуется драйвером, позволяя отложить обработку.

Пример:

```
int fdi,mask;
char vec;
fd_set fdset;
struct timeval ktimeout;

fdi = open("/dev/vmei2", O_RDONLY);

mask = 0x2;
ioctl(fdi, VME_INTHMASK_W, &mask);

ktimeout.tv_sec = 30;
ktimeout.tv_usec = 0;

FD_ZERO(&fdset);
FD_SET(fdi, &fdset);
if (select(fdi+1, &fdset, NULL, NULL, &ktimeout) != 0){
    if(read(fdi, &vec, 1)>0)
        printf("Caught irq vector %x\n",vec);
}
else
    printf("Time out\n");
close(fdi);
```

В данном примере 30 секунд ожидается прерывание по линии 2 VME шины.

Обработка сигнала AC_FAIL

AC_FAIL индицирует неисправность питания VME системы, что вызывает необходимость немедленной реакции программного обеспечения. Для обработки сигнала AC_FAIL приложение должно разрешить его прохождение на аппаратном уровне и установить обработчик программного сигнала SIGUSR1. Обработчиком программного сигнала является функция, в которой приложение выполняет необходимые действия при неисправности питания.

Пример

```
void ac_handler(int sig)
{
    printf("AC fail!");
    fflush(stdout);
}

main(){
    int fd,arg;
    ...
    fd = open("/dev/vme0",O_RDWR | O_EXCL);
    signal(SIGUSR1, ac_handler);
    ioctl(fd, VME_INTHMASK_R, &arg);
    arg = arg | 0x80;
    ioctl(fd, VME_INTHMASK_W, &arg);
    arg = 0;
    ioctl(fd, VME_FAILHANDLER, &arg);
    ...
}
```

Особенность обработки сигнала AC_FAIL состоит в том, что после однократного срабатывания маска прерывания (бит 7) сбрасывается в 0. Поэтому после получения данного сигнала необходимо вновь установить маску в 1.

2.8.2. Драйвер VME шины в ОС VxWorks

Драйвер VME шины выполняет следующие функции: инициализация регистров VME интерфейса; проведение операций чтения и записи на шине, выдача результата приложению; обработка прерываний VME шины; обработка сигнала AC_FAIL.

В ОС VxWorks каждому драйверу соответствует вход в файловую систему. Для VME драйвера это 8 специальных файлов /vmeN (где N=0..7). Чтобы провести операцию чтения или записи, приложение должно функцией `open` открыть файл устройства. Так как прикладных задач, работающих с VME шиной, может быть несколько, то, для избежания конфликтов, каждая из них должна работать только со своим специальным файлом /vmeN. Например, задача 0 открывает и работает с /vme2, задача 1 открывает и работает с /vme0. Для каждой задачи драйвер хранит свою копию размера данных, текущего адреса на VME шине, флага инкрементирования адреса.

После открытия файла устройства, задача с помощью вызовов `ioctl` должна установить параметры доступа к шине. Непосредственно ввод-вывод на VME шину осуществляется функциями `read` и `write`. Функции `read` и `write` должны вызываться с длиной буфера, кратной размеру данных, установленному `ioctl` `VME_SETSIZE`, т.е. 1,2 или 4 байтам. Отрицательное значение, возвращенное `read` или `write` говорит об ошибке на VME шине.

IOCTL ВЫЗОВЫ

- `VME_SETSIZE` – установить размер данных. Аргумент `ioctl` вызова при этом может быть установлен в один из трех вариантов – `DATA8`, `DATA16`, `DATA32`;
- `VME_SETADDR` – установить адрес на VME шине, по которому будет проведена операция чтения или записи;
- `VME_INCREMENT` – установить флаг инкрементирования адреса. Если аргумент `ioctl` установлен в 1, после каждой операции чтения или записи адрес на VME шине будет инкрементироваться (на 1,2 или 4 байта, в зависимости от размерности данных);
- `VME_AMSET` – установить код адресного модификатора. Аргумент должен содержать код адресного модификатора VME. Формат - см. описание регистра 0 VME интерфейса;
- `VME_BUSLEVEL` – установить уровень запроса на захват шины. Аргумент может быть в диапазоне 0..3. См. также описание VME запросчика;
- `VME_SYSTEM` – установить режим системного контроллера. Если аргумент =1, VME интерфейс будет работать в режиме системного контроллера;
- `VME_INTNMASK_W` – установить маску прерываний. См. описание регистра 1 VME интерфейса;

- VME_INTHMASK_R – прочитать маску прерываний. См. описание регистра 1 VME интерфейса;
- VME_FAIL_R – прочитать состояние сигналов SYS_FAIL и AC_FAIL. После выполнения ioctl вызова аргумент будет содержать в бите 0 состояние сигнала AC_FAIL, в бите 1 – состояние сигнала SYS_FAIL;
- VME_FAILHANDLER – установить функцию-обработчик сигнала AC_FAIL. Аргумент должен содержать указатель на функцию типа void name(void);
- VME_INTHANDLER – установить функцию-обработчик прерываний VME. Аргумент должен содержать указатель на функцию типа void name(int level, int vector).

Пример

```
int fd,arg;
fd = open("/vme0", O_RDWR, 0);
if (fd < 0) return(ERROR);
arg = 0xd; ioctl(fd, VME_AMSET, (int) &arg); /* supervisor space */
close(fd);
```

Данный участок кода устанавливает адресный модификатор VME равным 0xd.

Обработка прерываний

Для обработки прерывания от VME шины прикладная задача должна иметь функцию-обработчик прерывания вида

```
void function_name(int level, int vector);
```

С помощью ioctl вызова VME_INTHANDLER драйверу сообщается указатель на эту функцию. Чтобы прерывания от нужной линии проходили на аппаратном уровне, ioctl вызовом VME_INTHMASK_W необходимо также установить соответствующий бит маски прерывания в 1. При появлении сигнала прерывания, драйвер вызовет функцию с параметрами, содержащими номер линии (уровень) прерывания и вектор прерывания, выставленный на vme шине периферийным модулем.

Обработка сигнала AC_FAIL

AC_FAIL индицирует неисправность питания VME системы, что вызывает необходимость немедленной реакции программного обеспечения. Для обработки сигнала AC_FAIL приложение должно разрешить его прохождение на аппаратном уровне (вызовом VME_INTHMASK_W установить в 1 бит 7 маски прерываний) и сообщить драйверу функцию-обработчик (ioctl VME_FAILHANDLER). Обработчиком сигнала AC_FAIL является функция, в которой приложение выполняет необходимые действия при неисправности питания.

Заключение

Полученные ценовые характеристики контроллера, доступность элементной базы, его повторяемость в условиях ИЯФ позволяют рассматривать данную

разработку как одну из возможных альтернатив САМАС системам. В настоящий момент имеется опыт использования рассмотренного контроллера в составе ЯМР-магнетометра, разработанного в лаб.б. Контроллер также используется в качестве стенда для отладки электронных модулей в стандарте VME.