

Инструкция по эксплуатации
64-канального модуля дискретного ввода-вывода,
ЦАП и АЦП в конструктиве РСІ.
DD64-РСІ
ГФКП.467239.019

Содержание

1. Назначение	2
2. Технические характеристики.....	2
3. Структура DD64-PCI и контроллера DD64.....	3
4. Начало работы.....	4
5. Описание регистров общего назначения.....	4
6. Контроллер DI/DO. Работа с разовыми командами.	5
7. Выходные РК	6
8. Входные РК	7
9. Прерывания РК.....	7
10. Матрицы выдачи	9
11. «One Hot» фильтры выходных РК	10
12. Типы РК платы DD64-PCI	13
13. Описание ЦАП (тип AD5392).....	15
14. Работа с ЦАП AD5392.....	18
15. Типовые алгоритмы работы с ЦАП AD5392.....	19
16. Описание АЦП AD7328.....	20
17. Работа с АЦП AD7328.....	21
18. Типовые алгоритмы работы с АЦП AD7328.....	24
19. Работа с таймером.....	26
20. Рекомендации по эксплуатации и техническому обслуживанию	29
Приложение 1. Регистры DD64-PCI	32

1. Назначение.

Модуль DD64-PCI предназначен:

1. для сопряжения шины PCI с системами и устройствами по 64 независимым гальванически развязанным/связанными дискретным каналам (разовым командам (РК));
2. для преобразования аналоговых сигналов, представленных в виде напряжения, в цифровой код, передаваемый на шину PCI;
3. Для преобразования цифровых кодов, передаваемых в изделие по шине PCI в аналоговые сигналы, представленные в виде напряжения.

2. Технические характеристики.

- ▶ Шина PCI;
- ▶ Количество 16-ти разрядных портов ввода/вывода и прерываний: 8 и 1;
- ▶ до 64 гальванически развязанных/связанных вх./вых. разовых команд:
 - Количество входных $N_{вх}$ = от 0 до 64 входных РК;
 - Количество выходных $N_{вых}$ = от 0 до (64 - $N_{вх}$) выходных РК;Тип и гальваническая развязка определяются опционально при заказе;
- ▶ Время срабатывания выходного каскада РК: $T_{тур} = 0,18$ мс, $T_{мах} = 0,5$ мс;
- ▶ Время срабатывания входного каскада РК: $T_{тур} = 10$ мкс, $T_{мах} = 80$ мкс;
- ▶ Типы и характеристики РК :
 - Гальванически развязанные:
 - Параметры Вых. РК тип «сухой контакт»: Напряжение $U < 80В$; Ток $I_{мах} < 500mA$;
 - Гальванически НЕ развязанные:
 - Параметры Вых. РК тип «Ключ на корпус»: Напряжение $U < 80В$; Ток $I_{мах} < 500mA$;
 - Параметры Выходных РК типы: +5В/обрыв; +5В/Gnd; Gnd/+5В:
Напряжение $U = U_{ист} = 5В$ (Определяется значением источника питания +5В)
Ток $I_{мах} < 500mA$ для 1 каскада РК. (сумм. ток, определяется мощностью источника питания +5В).
 - Параметры Входных РК всех типов:
Минимальный диапазон напряжения срабатывания: 2 – 4 В*;
По умолчанию значение входного напряжения РК: логический «0» $0В \leq U_{вх} \leq 16В$
логическая «1» $20В \leq U_{вх}$
Минимальный Ток срабатывания не более $\sim 0,5mA$ *;
- ▶ до 16 каналов АЦП** с гальванической развязкой.
 - разрядность АЦП: 13 бит (12 + 1 знак).
 - время преобразование по 1 каналу 2,5 мкс
 - диапазон напряжений для каждого канала: +/-10, +/-5, +/-2.5, 0...10 В
 - входное сопротивление 10 Мом.
- ▶ до 8 каналов ЦАП** с гальванической развязкой.
 - разрядность ЦАП: 14 бит.
 - время преобразование по 1 каналу 10 мкс
 - диапазон напряжений для каждого канала: +/-10, +/-5, 0...10 В
- ▶ 16-ти разрядный Таймер, тактируемый от 16-ти разрядного делителя частоты 1МГц.
- ▶ Прерывание от любой РК и/или от регистра сравнения Таймера.
- ▶ Поддержка мультизадачного управления (запись регистров по маске).
- ▶ Встроенный источник питания +30В / 30мА.
- ▶ Возможность одновременного изменения состояния всех выходных РК по 1 команде.

* - значение может быть задано индивидуально для каждого канала при заказе.

** - суммарное количество каналов АЦП и ЦАП не может быть больше 16.

3. Структура DD64-PCI и контроллера DD64

Внутренняя организация модуля представлена на рис.:

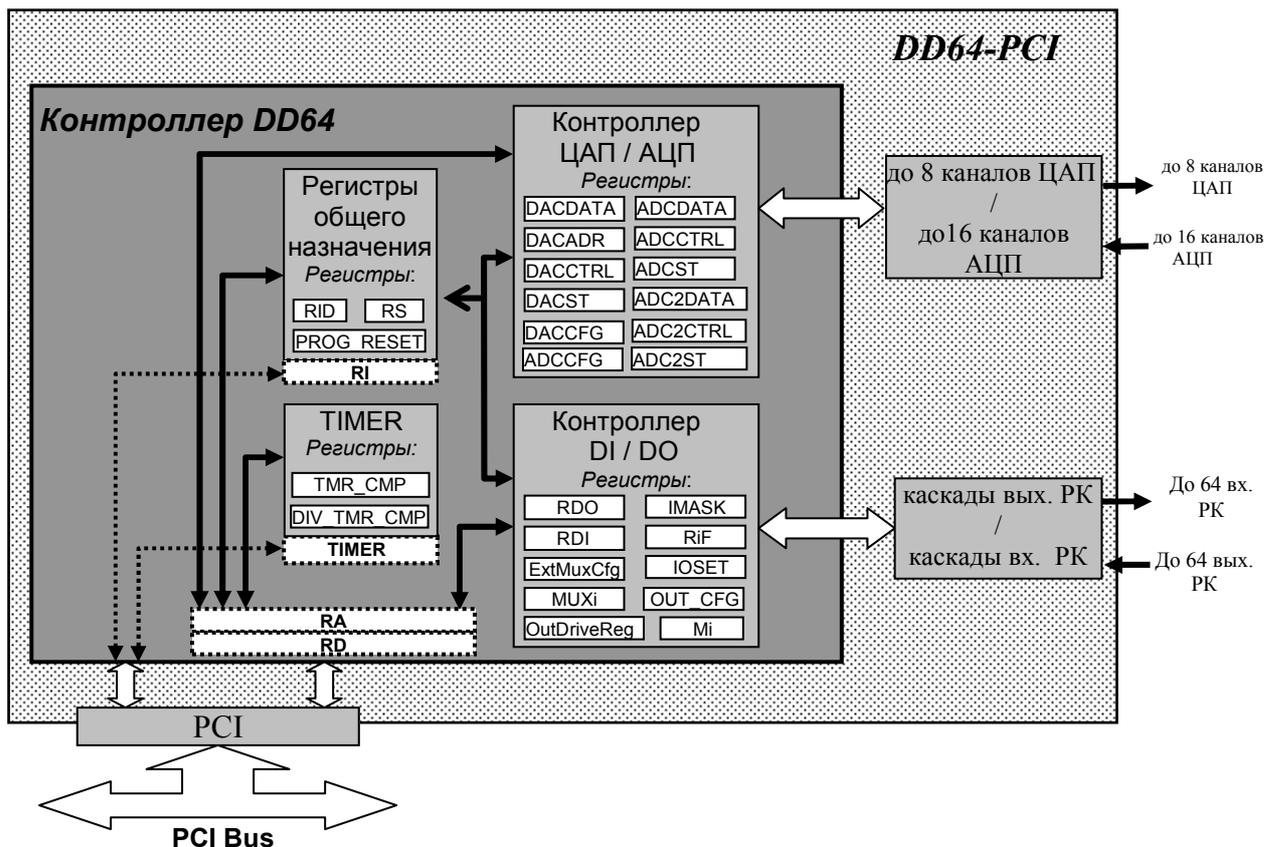


Рис. Структурная схема модуля DD64-PCI и контроллера DD64

Обмен данными модуля с шиной PCI осуществляются через 16-разрядные регистры-порты модуля, входящие в адресное пространство портов шины PCI. Доступ к внутренним регистрам организован через прямую и косвенную адресацию. В зоне прямой адресации находятся регистры с быстрым доступом (одна транзакция по PCI) **RI** – регистр прерываний и **TIMER** – регистр таймера, а также регистры доступа к зоне косвенной адресации **RA** – регистр адреса и **RD** – регистр данных. Доступ к остальным внутренним блокам происходит через регистры **RA** и **RD** (две транзакция со стороны PCI). В зоне косвенной адресации находятся регистры настройки таймера, регистры общего назначения, регистры управления каналами разовых команд (**PK**) и регистры ЦАП/АЦП.

Смещение	Регистр прямой адресации	RA	Регистры косвенной адресации
+0h	Reserved	00h	Регистры контроллера вх/вых PK: RDOi, RDli, OutDriveReg, IMASKi, RiFi, EXT_MUX_CFG, MUXi, IOSETi, OUT_CFGi, Mi
+2h	Reserved	.	.
+4h	Reserved	.	Регистры общего назначения :
+6h	Reserved	.	RID, RS, PROG_RESET
+8h	RI	.	Регистры контроллера ЦАП/АЦП:
+Ah	TIMER	.	DACDATA, DACADR, DACCTRL, DACST, ADCDATA, ADCCTRL, ADCST, ADC2DATA, ADC2CTRL, ADC2ST
+Ch	RA	.	Регистры таймера :
+Eh	RD	9Fh	TMRCMP, DIV_TMR_CLK

4. Начало работы

После установки модуля и запуска ЭВМ, на этапе работы программы BIOS происходит распределение ресурсов пространства PCI. В результате плата DD64-PCI получает свой базовый адрес BASE и номер прерывания IRQ. Дальнейшая работа с устройством осуществляется с использованием этих значений.

Обращение к регистрам устройства, находящихся в зоне косвенной адресации, производится через регистры адреса RA и данных RD:

Формат регистра адреса RA (Adress = BASE+Ch)

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	ADR ₁₅	ADR ₁₄	ADR ₁₃	ADR ₁₂	ADR ₁₁	ADR ₁₀	ADR ₉	ADR ₈	ADR ₇	ADR ₆	ADR ₅	ADR ₄	ADR ₃	ADR ₂	ADR ₁	ADR ₀
Доступ	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr

Формат регистра данных RD (Adress = BASE+Eh)

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	DAT ₁₅	DAT ₁₄	DAT ₁₃	DAT ₁₂	DAT ₁₁	DAT ₁₀	DAT ₉	DAT ₈	DAT ₇	DAT ₆	DAT ₅	DAT ₄	DAT ₃	DAT ₂	DAT ₁	DAT ₀
Доступ	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr

Для чтения или записи регистра косвенной адресации производится запись значение адреса в регистр RA, а затем чтение либо запись регистра RD соответственно.

5. Описание регистров общего назначения

В зоне регистров общего назначения находятся регистры, объединяющие управление и контроль внутренними функциональными блоками контроллера DD64. Рассмотрим их по отдельности:

Формат данных регистра идентификатора модуля RID (RA = 00h)

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	Hver ₃	Hver ₂	Hver ₁	Hver ₀	Fver ₇	Fver ₆	Fver ₅	Fver ₄	Fver ₃	Fver ₂	Fver ₁	Fver ₀	Rrev ₃	Rrev ₂	Rrev ₁	Rrev ₀
Доступ	rd															

Код в регистре содержит информацию об аппаратной версии модуля и о версии встроенного ПО, где:

Hver[3...0] – Аппаратная версия (Hardware version)

Fver[7...0] – Версия встроенного программного обеспечения (Firmware Version)

Rrev[3...0] – Редакция встроенного программного обеспечения (Firmware Revision)

Формат данных регистра программного сброса PROG RESET (RA = 75h)

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	x	x	x	x	x	x	x	x	x	x	x	x	Res ₃	Res ₂	Res ₁	Res ₀
Доступ	wr	wr	wr	wr												

При записи в биты Res[3..0] значения Ah происходит программный сброс модуля. Запись кода отличного от xxxAh, а также изменения остальных бит регистра значения не имеет.

При формировании программного сброса все управляющие регистры и таймер переходят в default-состояние, т.е. 0x0000, если не указано иное.

Формат данных регистра настроек **RS (RA = 01h)**

№ бита	15	14	13	12	11	10	9	8
Название бита	Tst_Pwr_On	IRQ_timer_mode	Global_int_ena	DO_Default_Set	Timer_Up_Down	Timer_mode	Timer_ena	Timer_int_ena
Доступ	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr

№ бита	7	6	5	4	3	2	1	0
Название бита	Int64_49_ena	Int_48_33_ena	Int_32_17_ena	Int_16_1_ena	DI64_49_ena	DI48_33_ena	DI32_17_ena	DI16_1_ena
Доступ	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr

Default = 1000h.

Задание режимов модуля, управление прерываниями, таймером, а также входными и выходными разовыми командами осуществляется через регистр настроек **RS**:

RS[0] = DI16_1_ena = 1/0 - разрешение/запрет приема вх-х ПК DI1 – DI16;
RS[1] = DI32_17_ena = 1/0 - разрешение/запрет приема вх-х ПК DI17 – DI32;
RS[2] = DI48_33_ena = 1/0 - разрешение/запрет приема вх-х ПК DI33 – DI48;
RS[3] = DI64_49_ena = 1/0 - разрешение/запрет приема вх-х ПК DI49 – DI64;

RS[4] = Int_16_1_ena = 1/0 - разрешение/запрет формир. прерываний от вх-х ПК DI1 – DI16.
RS[5] = Int_32_17_ena = 1/0 - разрешение/запрет формир. прерываний от вх-х ПК DI17 – DI32.
RS[6] = Int_48_33_ena = 1/0 - разрешение/запрет формир. прерываний от вх-х ПК DI33 – DI48.
RS[7] = Int_64_49_ena = 1/0 - разрешение/запрет формир. прерываний от вх-х ПК DI49 – DI64.

RS[8] = Timer_interrupt_ena = 1/0 - разрешение/запрет формирование прерываний от таймера
RS[9] = Timer_ena = 1/0 - запуск /остановка таймера
RS[10] = Timer_mode = 1/0 - однократный / циклический режим работы таймера
RS[11] = Timer_Up_Down = 1/0 - счет таймера с декрементом / инкрементом

RS[12] = DO_Default_Set = 1/0 - управление выходными ПК от матриц состояния (TTL, OutDriveReg) / от регистров выходных ПК (DO8_1...DO64_57).
 По умолчанию DO_Default_Set=1

RS[13] = Global_int_ena = 1/0 - режим разрешения / запрещения глобального прерывания.
RS[14] = IRQ_timer_mode = 1/0 - задает режим формирования прерывания от таймера: с отбоем по чтению таймера / без отбоя.

RS[15] = Test_Power_On = 1/0 - включение / отключение тестового источника питания TST_PWR 30 В мощностью 1 Вт, предназначенного для проверки разовых команд.

Формат данных регистра прерываний **RI (Adress= BASE+08h)**:

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Название бита	0	0	0	0	0	0	0	0	0	0	0	0	TMR	FLAG RiF64_49	FLAG RiF48_33	FLAG RiF32_17	FLAG RiF16_1
Доступ	rd	rd	rd	rd	rd												

Default = 0010h.

Код в регистре содержит вектор прерываний модуля. После возникновения прерывания первым должен быть прочитан и проанализирован вектор регистра **RI**, в котором устанавливаются флаги источников прерывания:

TMR – флаг прерывания от таймера (Default-значение TMR=1. Сбрасывается чтением регистра TIMER.)

FLAG RiF64_49 – флага от ПК 49-64, регистра RiF64_49

FLAG RiF48_33 – флага от ПК 33-48, регистра RiF48_33

FLAG RiF32_17 – флага от ПК 17-32, регистра RiF32_17

FLAG RiF16_1 – флага от ПК 1-16, регистра RiF16_1

Более подробно описание регистра RI рассмотрено в разделе «Прерывания ПК» и «Работа с таймером».

Формат данных регистра таймера **TIMER (Adress= BASE+0Ah)**:

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	TMR15	TMR14	TMR13	TMR12	TMR11	TMR10	TMR9	TMR8	TMR7	TMR6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0
Доступ	rd/wr															

где **TMR[15..0]** – значение таймера.

Более подробно описание регистра TIMER рассмотрено в разделе «Работа с таймером»

6. Контроллер DI/DO. Работа с разовыми командами.

Структурная схема Контроллера DI/DO и вх./вых. каскадов разовых команд:

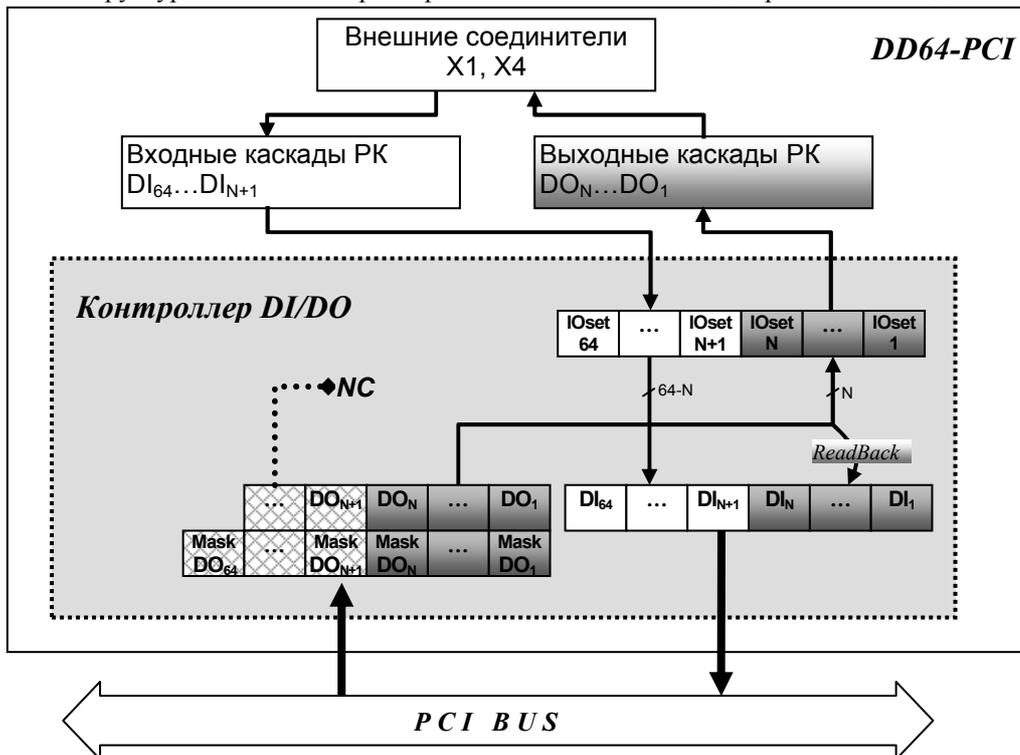


Рис. Структурная схема ПК, где:

RDO – глобальный 64-разрядный регистр **выходных** ПК.

RDI – глобальный 64-разрядный регистр **входных** ПК.

IOset – глобальный 64-разрядный регистр **конфигурации** входных / выходных ПК.

Программная модель модуля в части ПК является универсальной, т.е. не зависит от того, какая конфигурация ПК (инверсная или прямая логика, входная или выходная ПК) реализована (собрана) при заказе конкретного модуля.

Состояние конфигурации входных/выходных ПК отражено в регистре IOset, доступном только для чтения. Каждой ПК, вне зависимости от того, сконфигурирована она на вход или на выход, соответствует один бит в глобальном регистре вх-х ПК RDI и один бит в глобальном регистре выходных ПК RDO. Если n-я ПК сконфигурирована при сборке модуля **на выход**, то соответствующий ей n-й бит в регистре **входных** ПК RDI отражает текущее состояние выходного каскада ПК (функция readback на структурной схеме). Если n-я ПК реализована при сборке модуля как **вход**, то соответствующий ей n-й бит регистра входных ПК RDI отражает текущее состояние линии ПК, а состояние n-ого бита регистра **выходных** ПК RDO не имеет значения.

7. Выходные ПК

Значение, записанное в регистр RDO, устанавливает выходные линии ПК в состояние включено/выключено.

Структура глобального 64-разрядного регистра выходных разовых команд **RDO**

№ вых ПК	57-64	49-56	41-48	33-40	25-32	17-24	9-16	1-8
Название регистра	RDO64_57	RDO56_47	RDO48_41	RDO40_33	RDO32_25	RDO24_17	RDO16_9	RDO8_1
Адрес в RA	0Fh	0Eh	0Dh	0Ch	0Bh	0Ah	09h	08h

Как видно из таблицы, глобальный регистр RDO состоит из восьми регистров RDO64_57..RDO8_1, каждый из которых имеет формат данных, представленный ниже:

Формат данных регистров выходных ПК RDO_{n+7_n} (RA= 08h ... 0Fh):

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	mask DO _{n+7}	mask DO _{n+6}	mask DO _{n+5}	mask DO _{n+4}	mask DO _{n+3}	mask DO _{n+2}	mask DO _{n+1}	mask DO _n	DO _{n+7}	DO _{n+6}	DO _{n+5}	DO _{n+4}	DO _{n+3}	DO _{n+2}	DO _{n+1}	DO _n
Доступ	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr						

Где n – это номер первой ПК в регистре, принимает значения:

n = 1 (RDO8_1), 9 (RDO16_9), 17 (RDO24_17), 25 (RDO32_25), 33 (RDO40_33), 41 (RDO48_41), 49 (RDO56_49), 57 (RDO64_57);

DO_{n+x} –бит, отвечающий за состояние ПК с номером n+x (где X от 0 до 7);

mask DO_{n+x} – маска ПК с номером n+x (где X от 0 до 7).

Запись в 16-разрядные регистры RDO_{n+7_n} осуществляется по маске mask DO_{n+x}, где биты младшего байта определяют состояние выходных ПК, а соответствующие биты старшего байта разрешают/запрещают изменение битов младшего байта регистра RDO_{n+7_n} ('1'-запись разрешена, '0'-изменение запрещено). Данная возможность реализована для поддержки мультизадачного управления, т.е. использования одного и того же регистра ПК в различных независимых приложениях. Используя данный механизм, разные программы могут независимо использовать ПК, управление которыми осуществляется через один и тот же регистр RDO_{n+7_n} без необходимости отслеживания за битами ПК «чужих» приложений.

8. Входные ПК

Прием разовых команд осуществляется в глобальный регистр входных ПК RDI.

Структура глобального 64-разрядного регистра входных разовых команд **RDI**:

№ вх ПК	49-64	33-48	17-32	1-16
Название регистра	RDI64_49	RDI48_33	RDI32_17	RDI16_1
Адрес в RA	0Fh	0Dh	0Bh	09h

Как видно из таблицы, регистр RDI состоит из четырех регистров RDI64_49 .. RDI16_1, каждый из которых имеет формат данных, представленный ниже:

Формат данных регистров входных ПК **RDI_{m+15_m}** (RA= 09h, 0Bh, 0Dh, 0Fh):

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	DI_{m+15}	DI_{m+14}	DI_{m+13}	DI_{m+12}	DI_{m+11}	DI_{m+10}	DI_{m+9}	DI_{m+8}	DI_{m+7}	DI_{m+6}	DI_{m+5}	DI_{m+4}	DI_{m+3}	DI_{m+2}	DI_{m+1}	DI_m
Доступ	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd

где m – номер первой ПК в регистре, принимает значения:

m = 1 (RDI16_1), **17** (RDI32_17), **33** (RDI48_33), **49** (RDI64_49).

DI_{m+y} – бит, отвечающий за состояние ПК с номером m+y (где Y от 0 до 15);

Текущие состояния ПК, сформированных в глобальном регистре **RDO**, отражаются в соответствующих битах глобального регистра входных ПК **RDI (ReadBack)**.

Пример. Если выдать, например, 5-ю ПК, записав в регистр RDO8_1 (RA=0x08h) значение 0xFF10 (запись по маске FFh) и например 15-ю ПК, записав в регистр RDO16_9 (RA=0x09h) значение 0xFF40 (запись по маске FFh), то в регистре RDI16_1(09h), доступном по чтению, можно будет прочитать значение 0x4010.

9. Прерывания ПК

В модуле реализован механизм формирования прерывания от любой ПК, в том числе и от выходных ПК через регистр RDI посредством функции **ReadBack**.

Работа с прерываниями модуля осуществляется через регистры **RI** (зона прямой адресации), **RiF**, **iMASK**, а также битов **13, 7..4** регистра **RS**.

Разрешение глобального прерывания производится записью значения '1' в бит RS[13] (Global interrupt enable). Разрешение прерывания от ПК устанавливается в регистре RS[7..4], каждый из 4-х битов которого отвечает за определенную группу ПК. Значение '1' записанное в RS[4]- разрешает прерывания от группы ПК 1-16, RS[5] – от ПК 17-32, RS[6] – от ПК 33-48, а RS[7] – от ПК 49-64. Побитовое разрешение прерываний от ПК производится в 64-разрядном глобальном регистре **iMASK**.

Информация о принятых прерываниях отображается в регистре **RI**, а также в глобальном 64-разрядном регистре **RiF**, состоящем из 4-х регистров RiF64_49, RiF48_33, RiF32_17, RiF16_1. В модуле реализован механизм быстрого анализа прерываний, при котором нет необходимости читать все четыре регистра **RiF_{i+7_i}** (где i=1, 17, 33, 49), определяя наличие флагов прерываний от ПК. Достаточно прочитать регистр **RI**, содержащий информацию о том, какой из четырех регистров **RiF_{i+7_i}** был модифицирован, и далее прочитать соответствующий регистр **RiF**, чтобы понять, от какой именно ПК поступило прерывание.

Примечание. Выходная ПК также может являться источником прерываний, если прерывания были разрешены соответствующими этой ПК битами регистра iMASK.

Формат данных регистра прерываний **RI (Adress= BASE+08h)**:

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	0	0	0	TMR	FLAG RiF64_49	FLAG RiF48_33	FLAG RiF32_17	FLAG RiF16_1
Доступ	rd	rd	rd	rd	rd											

TMR – флаг прерывания от таймера

FLAG RiF64_49 – флага от ПК 49-64, регистра RiF64_49

FLAG RiF48_33 – флага от ПК 33-48, регистра RiF48_33

FLAG RiF32_17 – флага от ПК 17-32, регистра RiF32_17

FLAG RiF16_1 – флага от ПК 1-16, регистра RiF16_1

Настройка прерываний от входных ПК осуществляется в глобальном регистре масок прерываний **ПК iMASK**.

Структура глобального 64-разрядного регистра масок прерываний **iMASK**:

№ вых ПК	57-64	49-56	41-48	33-40	25-32	17-24	9-16	1-8
Название регистра	iMASK_64_57	iMASK_56_47	iMASK_48_41	iMASK_40_33	iMASK_32_25	iMASK_24_17	iMASK_16_9	iMASK_8_1
Адрес в RA	1Fh	1Eh	1Dh	1Ch	1Bh	1Ah	19h	18h

Глобальный регистр iMASK состоит из восьми регистров iMASK_64_57 .. iMASK_8_1, каждый из которых имеет формат данных, представленный ниже:

Формат данных регистров масок прерываний PK iMASK_{k+7..k} (RA = 18h ... 1Fh):

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	M ₂ DI _{k+7}	M ₁ DI _{k+7}	M ₂ DI _{k+6}	M ₁ DI _{k+6}	M ₂ DI _{k+5}	M ₁ DI _{k+5}	M ₂ DI _{k+4}	M ₁ DI _{k+4}	M ₂ DI _{k+3}	M ₁ DI _{k+3}	M ₂ DI _{k+2}	M ₁ DI _{k+2}	M ₂ DI _{k+1}	M ₁ DI _{k+1}	M ₂ DI _k	M ₁ DI _k
Доступ	wr	wr	wr													

где k – номер первой PK в регистре, принимает значения:

k = 1 (iMASK_8_1), 9 (iMASK_16_9), 17 (iMASK_24_17), 25 (iMASK_32_25), 33 (iMASK_40_33), 41 (iMASK_48_41), 49 (iMASK_56_49), 57 (iMASK_64_57).

Каждой разовой команде DI_{k+z} (где z от 0 до 7) отводится по 2 бита – M₂ и M₁ со следующими значениями:

M₂M_{1(k+z)} = 00 – прерывание запрещено от (k+z)-й входной PK;

M₂M_{1(k+z)} = 01 – прерывание по фронту от (k+z)-й PK;

M₂M_{1(k+z)} = 10 – прерывание по спаду от (k+z)-й PK;

M₂M_{1(k+z)} = 11 – прерывание по фронту и по спаду от (k+z)-й PK.

Глобальный регистр флагов прерываний разовых команд RiF:

Структура глобального 64-разрядного регистра RiF:

№ вх PK	49-64	33-48	17-32	1-16
Название регистра	RiF64_49	RiF48_33	RiF32_17	RiF16_1
Адрес в RA	2Fh	2Dh	2Bh	29h

Глобальный регистр флагов прерываний RiF состоит из четырех регистров RiF64_49 .. RiF16_1, каждый из которых имеет формат данных, представленный ниже:

Формат данных регистров флагов прерываний RiFi+7..i (RA = 29h, 2Bh, 2Dh, 2Fh):

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	RiFi+15	RiFi+14	RiFi+13	RiFi+12	RiFi+11	RiFi+10	RiFi+9	RiFi+8	RiFi+7	RiFi+6	RiFi+5	RiFi+4	RiFi+3	RiFi+2	RiFi+1	RiFi
Доступ	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr

где i – номер первой PK в регистре и соответствующего ему флага, принимает значения:

i=1 (RiF16_1), 17 (RiF32_17), 33 (RiF48_33), 49 (64_49).

RiFi+q – флаг прерывания от входной PK DI_{i+q}.

В случае если в iMASK прерывание разрешено, в регистре RiF при соответствующем событии возникает флаг, а при разрешенном бите RS[7..4] () будет сгенерировано прерывание. Для правильной обработки прерываний ПО пользователя, предусмотрен механизм корректного обнуления обработанных флагов: чтобы отбить прерывания необходимо записать в RiF прочитанное из него значение и тем самым обнулить только обработанные флаги. Если же за время, которое ПО тратит на обработку значений прочитанных из RiF возникнут новые флаги, то после записи последнего прочитанного значения обнулятся только обработанные флаги, а новые флаги сохранят линию прерывания в активном состоянии, что снова сгенерирует прерывание при выходе из обработчика.

10. Матрицы выдачи

Матрицы выдачи представляют собой восемь 64-х разрядных массивов, записанных в ПЗУ модуля и предназначенных для одномоментной выдачи на выходные каскады РК по команде ПО пользователя, либо путем задания переключателей на плате. С помощью матриц выдачи пользователь может установить/предустановить состояние выходных РК в обход регистра RDO, а так же в момент подачи питания, когда плата ещё не находится под управлением ПО пользователя. Обращаем внимание на то, что количество разрядов матрицы может отличаться от количества выходных РК модуля. Те биты матрицы, которые соответствуют входным РК не имеют значения. Такая структура создана для обеспечения универсальной программной модели, т.е. независимости аппаратной реализации модуля от внутренней структуры регистров: номер бита в матрице всегда соответствует номеру РК, не зависимо от того вход это или выход. Структурная схема управления выходными РК представлена на рис.

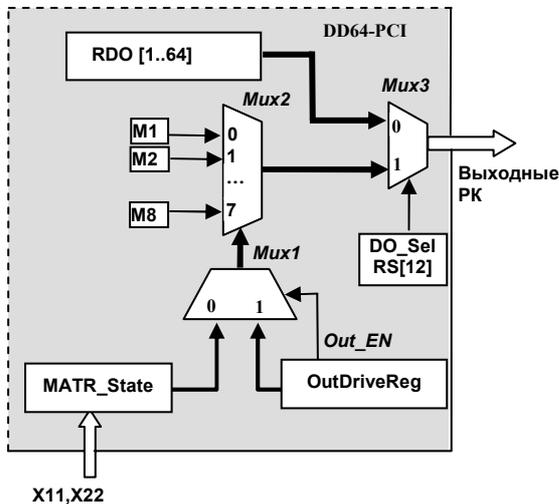


Рис. Структурная схема управления выходными РК

Mux3 – мультиплексор определяющий источник сигналов для выходных РК.
Управление: Бит в регистре **RS[12]-(DO_Sel)**, устанавливает, будут ли в каскады выходных РК передаваться значения регистра RDO или одна из матриц, определяемых мультиплексором Mux2.

Default state: **RS[12]-(DO_Sel)** = 1, т.е. управление выходами РК от матрицы Mn, где n = 1..8.

Mux2 – мультиплексор подключения матрицы Mn, где n = 1..8, которая будет подана на Mux3.

Управление: Установка номера матрицы производится от мультиплексора Mux1.

Mux1 – мультиплексор коммутирующий номер матрицы, заданный в регистрах **OutDriveReg** и **RD_MATR** (состояние переключателей X11, X22).

Управление: Бит в регистре **OutDriveReg[15]-(Out_EN)** задает источник управления выбором матрицы

Default state: **OutDriveReg[15]-(Out_EN)** = 0, т.е. управление от **RD_MATR** (X11, X22).

RDO (rd/wr) – 64-х разрядный глобальный регистр выходных РК;

M1..M8 (rd) – 64-разрядные матрицы выдачи.

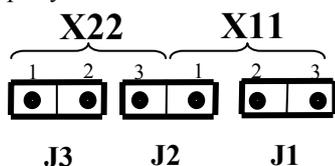
MATR_State (rd) – регистр отражающий состояние переключателей X11, X22;

OutDriveReg (rd/wr) – регистр программного задания [биты 2..0] и управления источником [бит 15] номера матрицы выдачи Mn, где n = 1..8.

Out_EN => **OutDriveReg[15]** = 0/1 – номер матрицы задается регистром **RD_MATR** (X11, X22) / программно, значением в регистре **OutDriveReg[2..0]**.

Работа матриц выдачи M1..M8:

В момент подачи питания выходные каскады РК переходят в состояния, задаваемые одной из восьми матриц выдачи. Матрицы выдачи содержат стандартные значения, жестко заданные в плате и не доступные для модификации, например, 0x0000.0000.0000.0000h, 0x0F00.0F00.0F00.0F00h или 0xFF00.FF00.FF00.FF00h (см. таблицу), эти значения доступны по чтению. Матрицы выдачи позволяют еще до загрузки ПО пользователя, которая может занять длительное время, установить выходные РК в определенные состояния. Матрица выдачи называется **активной**, если значение, содержащееся в ней, передано на выходы РК. После загрузки управляющего ПО номер активной матрицы может быть изменён программно либо управление переключено на регистр RDO. Номер активной матрицы, после подачи питания определяется состояниями джамперов J1, J2, J3 разъема X11, X22 модуля. С помощью джамперов можно установить любую из 8 комбинаций на X11, X22 и задать таким образом активную матрицу:



Наличие (1) / Отсутствие (0) джамперов			Матрица	Адреса RA	Доступ	Значение матрицы
J3	J2	J1				
0	0	0	M1	80h..83h	rd	0x0000.0000.0000.0000
0	0	1	M2	84h..87h	rd	0x0100.0100.0100.0100
0	1	0	M3	88h..8Bh	rd	0x0300.0300.0300.0300
0	1	1	M4	8Ch..8Fh	rd	0x0700.0700.0700.0700
1	0	0	M5	90h..93h	rd	0x0F00.0F00.0F00.0F00
1	0	1	M6	94h..97h	rd	0x1F00.1F00.1F00.1F00
1	1	0	M7	98h..9Bh	rd	0x3F00.3F00.3F00.3F00
1	1	1	M8	9Ch..9Fh	rd	0xFF00.FF00.FF00.FF00

Таким образом, пользователь имеет возможность задать начальные состояния РК еще до загрузки ОС.

Конфигурация джамперов на X11, X22 доступна по чтению в регистре **MATR_State**.

Формат данных регистра **MATR_State** (RA = 60h):

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Название бита	0	0	0	0	0	0	0	0	0	0	0	0	0	0	J3	J2	J1
Доступ	rd																

Для программного переключения между матрицами (установки активной матрицы), уже после загрузки пользовательского ПО, используется регистр **OutDriveReg**.

Формат данных регистра **OutDriveReg (RA = 14h)**

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	OUT_EN	x	x	x	x	x	x	x	x	x	x	x	x	mset2	mset1	mset0
Доступ	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr

mset2 ... mset0 – биты задают номер активной матрицы №№1-8

mset2	mset1	mset0	Матрицы
0	0	0	M1
0	0	1	M2
0	1	0	M3
0	1	1	M4
1	0	0	M5
1	0	1	M6
1	1	0	M7
1	1	1	M8

OUT_EN = 1 – управление выходами по активной матрице, заданной в регистре **OutDriveReg**.

0 - управление выходами по активной матрице, заданной джамперами на X11, X22.

Переключение управления между активной матрицей и регистром **RDO** ПК осуществляется установкой и снятием бита **DO_Default_Set** регистра **RS[12]** (см.стр.5).

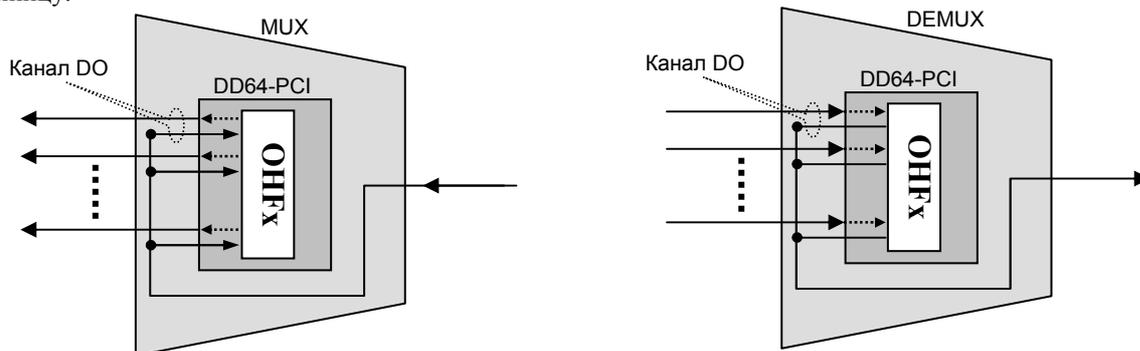
DO_Default_Set = 1 (*Default state*) – управление выходными ПК осуществляется от активной матрицы M_n (где $n=1..8$).

= 0 – управление выходными ПК осуществляется от регистра выходных ПК **RDO**.

11. «One Hot» фильтры выходных ПК

Модуль в части выходных ПК (DO) представляет собой, набор независимо управляемых линий, т.е. между линиями отсутствует взаимосвязь на аппаратном уровне. В программной модели контроллера DI/DO предусмотрена возможность задания взаимной связи между группами ПК по принципу «One Hot» (далее OH), т.е. в группе может быть активной (т.е. значение бита в **RDO** = "1") только одна ПК либо все ПК выключены. Данная возможность обеспечивает защиту на аппаратном уровне в особо ответственных задачах управления, где недопустимо совместное включение любых из группы управляющих выходных ПК, вне зависимости от состояния ПО и ОС (зависание, программный сбой, ошибка на этапе отладки ПО), под управлением которых работает модуль. Одной из таких задач является мультиплексирование и демультимплексирование сигнала (см. рис.).

ОН-фильтры устанавливаются на группу ПК. Глобальные 64-разрядные регистры **OHF1**, **OHF2**, **OHF3** позволяют организовать до 3-х ОН-фильтров для групп выходных ПК модуля. Для того, чтобы организовать фильтр группы выходных ПК, необходимо установить соответствующие биты в регистрах настройки **OHF1**, **OHF2**, **OHF3** в единицу.



Пример мультиплексирования и демультимплексирования сигналов с помощью модуля DD64-PCI с аппаратной защитой фильтром OHF

Структурная схема и пример реализации ОН-фильтров представлен на рисунке.

В группе ПК, объединенной ОН-фильтром, на выходной каскад будет поступать только старшее значение «1» из установленной для ПК группы в регистре **RDO**. Все значения ПК = «1», которые младше, будут обнулены ОН-фильтром и поступят на выходные каскады, при этом соответствующие биты регистра **RDO** останутся неизменными. Как видно из рисунка фильтры **OHF1**, **OHF2**, **OHF3** расположены таким образом, что на фильтр со старшим номером может быть подан результат, полученный от фильтра с младшим номером.

Рассмотрим пример на рисунке. В фильтре **OHF1** организована группа1 из трех произвольных ПК: RDO_a , RDO_b и RDO_c . Значения : RDO_a и $RDO_b = \langle 1 \rangle$, а : $RDO_c = \langle 0 \rangle$. Значение ПК = «1» старшей в группе установлено для RDO_b . Соответственно на выходе **OHF1** только RDO_b будет иметь значение «1».

В фильтре OHF2 организована группа 2 из трех ПК: RDO_a, RDO_b и RDO_d, т.е. таким образом, что 2 ПК из группы 1 попадают в группу 2. Значение же старшей ПК на входе группы 2 равной «1» соответствует RDO_d. Т.е. на выходе фильтра OHF2 в состоянии «1» будет только ПК RDO_d.
 011_{bin} => OHF1 => 010_{bin}; 110_{bin} => OHF2 => 100_{bin}

Простое применение фильтра показано для группы 3 из 3-х ПК. Здесь значения RDO₆₃, RDO₁ и RDO₂ = «1», при этом на выходные каскады поступит только старшее единичное значение ПК RDO₆₃. Значение RDO₁ и RDO₂ на выходе будет соответствовать «0». 111_{bin} => 100_{bin}.

Обращаем внимание, что объединение в группы ПК через ОН-фильтры никак не влияет на значения в регистре RDO. Значения же для выходных ПК прочитанных в регистре RDI (ReadBack) будут соответствовать фактическому состоянию, поступающему на выходные каскады ПК.

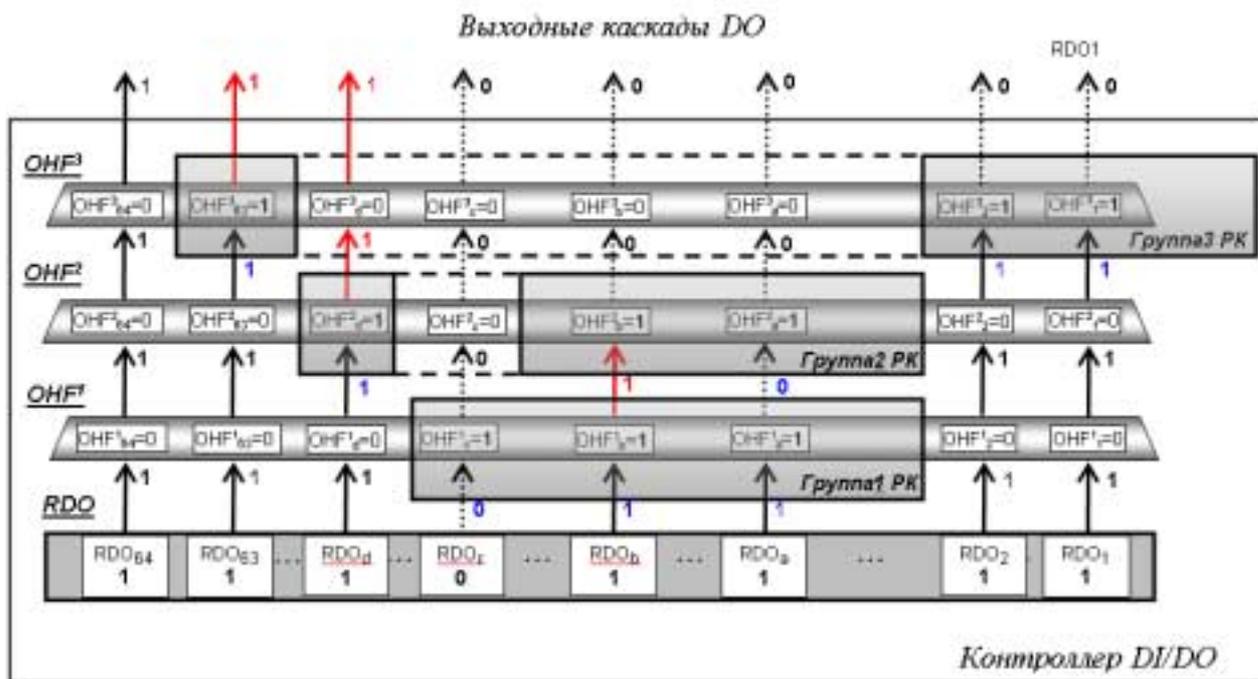


Рис. Пример реализации схемы ОН-фильтров модуля DD64-PCI.

При наличии нескольких модулей DD64-PCI имеется возможность объединения ОН-фильтров этих модулей. Для этого используется порты OHF_sm[3..1] разъема X33 (см. рис). Фильтры OHF³..OHF¹ подключены к линиям портов OHF_sm3 .. OHF_sm1 соответственно.

Любая линия может быть настроена в регистре Ext_OHF_sm как Master (выход) или как Slave (вход). Если линия настроена как Master, то на ней устанавливается активный уровень 0 или 1 в соответствии со значением подключенного к ней ОН-фильтра: если хотя бы одна ПК объединенная в группу ОН-фильтра находится в состоянии «1», то значение выхода Master будет в «1», если же все ПК группы в состоянии «0», выход Master = «0». Т.о. происходит передача состояния ОН-фильтра к фильтру другой платы. ОН-фильтр другой платы принимает через свою линию OHF_sm в режиме Slave состояние фильтра платы Master. Таким образом ОН-фильтры 2-х разных плат объединяются в один с распределением ПК от старших в плате Master к младшим в плате Slave. Возможно объединение до 4-х плат.

ВНИМАНИЕ!!! Недопустимо объединение плат и одновременная установка при этом состояния соединенных линий OHF_sm в режим Master (в регистре Ext_OHF_sm).

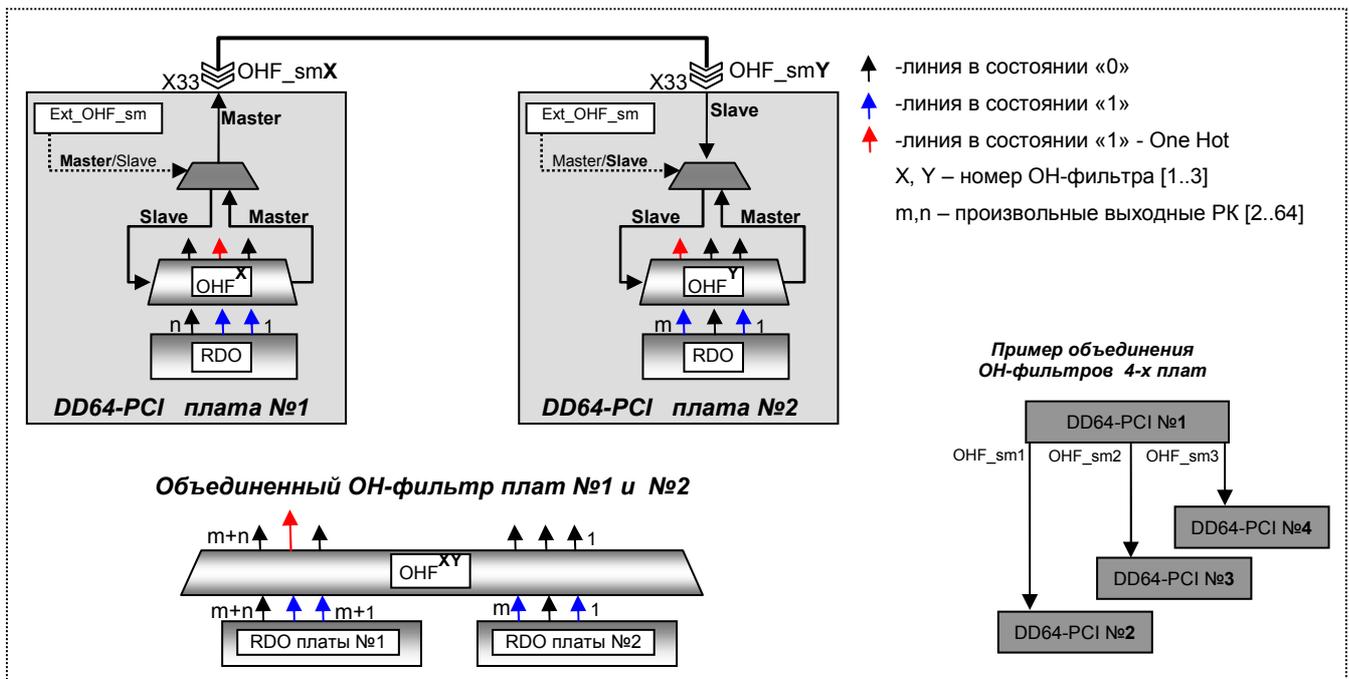


Рис. Структурная схема объединения фильтров модулей DD64-PCI.

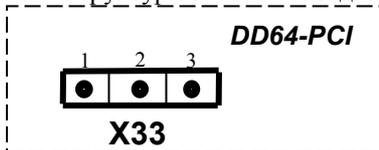


Рис. Разъем X33 портов внешнего конфигурирования ОН-фильтров
 где X33-1 – порт OHF_sm1
 X33-2 – порт OHF_sm2
 X33-3 – порт OHF_sm3

• Регистры настройки фильтров ОНФ.

Структура трех глобальных 64-разрядных регистров настройки фильтров ОНФ:

	№ вых РК	49-64	33-48	17-32	1-16
ОНФ ¹	Название регистра	ОНФ ¹ 64_49	ОНФ ¹ 48_33	ОНФ ¹ 32_17	ОНФ ¹ 16_1
	Адрес в RA	3Fh	3Dh	3Bh	39h
ОНФ ²	Название регистра	ОНФ ² 64_49	ОНФ ² 48_33	ОНФ ² 32_17	ОНФ ² 16_1
	Адрес в RA	4Fh	4Dh	4Bh	49h
ОНФ ³	Название регистра	ОНФ ³ 64_49	ОНФ ³ 48_33	ОНФ ³ 32_17	ОНФ ³ 16_1
	Адрес в RA	5Fh	5Dh	5Bh	59h

Каждый глобальный регистр ОНФ¹, ОНФ², ОНФ³ состоит из четырех регистров ОНФ^{1,2,3}64_49 .. ОНФ^{1,2,3}16_1, каждый из которых имеет формат данных, представленный ниже:

Формат данных регистров настройки фильтров: **ОНФ¹_{n+15_n} (RA = 39h, 3Bh, 3Dh, 3Fh)**
ОНФ²_{n+15_n} (RA = 49h, 4Bh, 4Dh, 4Fh)
ОНФ³_{n+15_n} (RA = 59h, 5Bh, 5Dh, 5Fh)

№ вых.РК	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	ОНФ ¹ _{n+15}	ОНФ ¹ _{n+14}	ОНФ ¹ _{n+13}	ОНФ ¹ _{n+12}	ОНФ ¹ _{n+11}	ОНФ ¹ _{n+10}	ОНФ ¹ _{n+9}	ОНФ ¹ _{n+8}	ОНФ ¹ _{n+7}	ОНФ ¹ _{n+6}	ОНФ ¹ _{n+5}	ОНФ ¹ _{n+4}	ОНФ ¹ _{n+3}	ОНФ ¹ _{n+2}	ОНФ ¹ _{n+1}	ОНФ ¹ _n
Доступ	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr

где n – номер первой РК в регистре = 1 (ОНФ¹16_1), n = 17 (ОНФ¹32_17), n = 33 (ОНФ¹48_33), n = 49 (ОНФ¹64_49);
 ОНФ^X_i = 1/0 – подключение/отключение i-й выходной РК к входу ОН-фильтра X (где n = [64..0], X = [3..1]).

Регистры конфигурации портов подключения внешних ОН-фильтров модуля EXT_OHF_sm (RA = 30h)

Формат данных регистра:

№ бита	7	6	5	4	3	2	1	0
Название бита	0	OHF_sm3 Slave	OHF_sm2 Slave	OHF_sm1 Slave	0	OHF_sm3 Master	OHF_sm2 Master	OHF_sm1 Master
Доступ	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr

Возможные конфигурации линии OHF_sm_n (где n = [1..3]):

OHF_sm _n Slave	OHF_sm _n Master	Режим	Состояние линии порта OHF_sm _n
0	0	NC	линия OHF_sm _n отключена. Внутренний фильтр OHF ⁿ не передает и не зависит от состояния внешнего порта EXT_OHF_sm.
0	1	Master	линия OHF_sm _n сконфигурирована как выход и передает состояние фильтра OHF ⁿ .
1	X	Slave	линия OHF_sm _n сконфигурирована как выход и влияет на состояние фильтра OHF ⁿ .

12. Типы ПК платы DD64-PCI

Возможные типы разовых команд платы DD64-PCI*:

Тип	Выходные	Входные
	Гальванически развязанные	
Сухой контакт		
Гальванически НЕ развязанные		
5В / Обрыв		
5В / 0		
0 / Обрыв (Ключ на корпус)		
0 / 5В		

Примечания:

R_{prot} – защитный резистор порядка единиц Ом (зависит от исполнения).

v.1.0.corr27.v.

R_{thr} - резистор, определяющий напряжение срабатывания РК

R_i - подтягивающий резистор.

1. Ограничение по току на один вывод выходной РК: $I_{max} < 500$ мА.

2. Суммарный ток на все выходы $\sum I_{max}$ ограничен мощностью источника питания системы.

3. Обозначения на схемах «GND» соответствуют значению «земли» источника питания.

* Каждая из 64 разовых команд может быть смонтирована на плате любым вышеописанным способом (опционально при заказе).

Информация о наличии и направлении каскадов разовых команд модуля отображается в глобальных 64-разрядных регистрах конфигурации IOCFG1 и IOCFG2. Каждой РК соответствует один бит в регистре IOCFG1 и один бит в регистре IOCFG2. 64-разрядные регистры IOCFG1 и IOCFG2 доступны только по чтению и отображают информацию о наличии и направлении каждой РК, которая может являться Входной, Выходной, либо быть физически не установленной на плате. Информация об этом приведена в таблице ниже, на примере произвольной i-й РК (где i может принимать значения от 1 до 64):

IOCFG1 _i	IOCFG2 _i	Наличие/направление РК
0	0	РК не установлена на плате
	1	РК установлена / Входная РК
1	x	РК установлена / Выходная РК

Структура глобальных 64-разрядных регистров IOCFG1:

№ вх РК	49-64	33-48	17-32	1-16
Название регистра	IOCFG1_64_49	IOCFG1_48_33	IOCFG1_32_17	IOCFG1_16_1
Адрес в RA	7Bh	7Ah	79h	78h

Структура глобальных 64-разрядных регистров IOCFG2:

№ вх РК	49-64	33-48	17-32	1-16
Название регистра	IOCFG2_64_49	IOCFG2_48_33	IOCFG2_32_17	IOCFG2_16_1
Адрес в RA	7Fh	7Eh	7Dh	7Ch

Каждый регистр IOCFG_x (где x=1 или 2) состоит из четырех регистров IOCFG_x_64_49 .. IOCFG_x_16_1, в свою очередь, каждый из них имеет формат данных, представленный ниже:

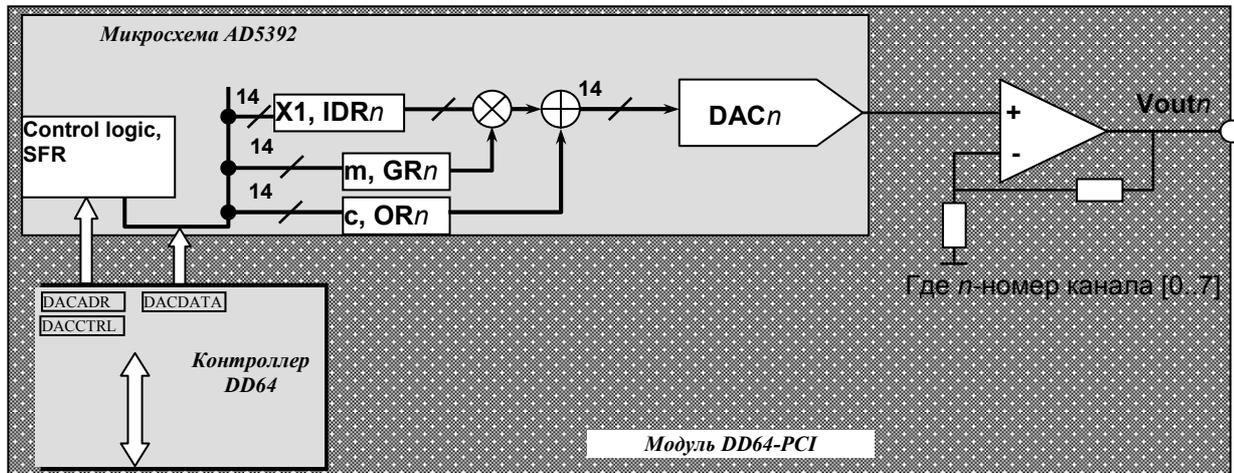
Формат данных регистров конфигурации РК **IOCFG_x_n+15_n (RA = 78h, 79h, 7Ah, 7Bh):**

№ вых РК	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	IOCFG _x _{n+15}	IOCFG _x _{n+14}	IOCFG _x _{n+13}	IOCFG _x _{n+12}	IOCFG _x _{n+11}	IOCFG _x _{n+10}	IOCFG _x _{n+9}	IOCFG _x _{n+8}	IOCFG _x _{n+7}	IOCFG _x _{n+6}	IOCFG _x _{n+5}	IOCFG _x _{n+4}	IOCFG _x _{n+3}	IOCFG _x _{n+2}	IOCFG _x _{n+1}	IOCFG _x _n
Доступ	rd	rd	rd	rd	rd	rd	Rd	rd	Rd	rd	rd	rd	rd	rd	rd	rd

где n – номер первой РК в регистре = 1 (IOCFG_x16_1), n = 17 (IOCFG_x32_17),
n = 33 (IOCFG_x48_33), n = 49 (IOCFG_x64_49);

13. Описание ЦАП (тип AD5392)

Модуль имеет до 8 каналов ЦАП в зависимости от исполнения, реализованных на базе микросхемы AD5392 с разрешающей способностью 14-разрядов. Все каналы имеют программируемый диапазон выходного напряжения (одинаковый для всех каналов). Доступны диапазоны $\pm 10\text{В}$, $\pm 5\text{В}$, $0\text{В}..+10\text{В}$. Каждый из каналов ЦАП имеет программируемый коэффициент усиления и коэффициент смещения, которые позволяют индивидуально калибровать каждый из каналов ЦАП. Схематическая структура ЦАП представлена на рисунке:



Работа с ЦАП модуля DD64-PCI осуществляется путем настройки и передачи данных в **регистры данных каналов IDR**, **регистры коэффициента усиления GR**, **регистры смещения OR**, **регистры спец.функций SFR** микросхемы AD5392. Доступ к этим регистрам осуществляется через регистры **DACADR**, **DACDATA**, **DACCTRL** контроллера DD64, которые в свою очередь доступны по PCI.

Регистры ЦАП (IDR, OR, GR, SFR), а также механизмы доступа к ним по PCI будут описаны ниже.

Рассмотрим внутреннюю структуру одного канала ЦАП (где n – номер канала от 0 до 7):
Выходное напряжение **Vout** вычисляется по формуле:

$$V_{out} = (V_{\max} - V_{\min}) \cdot \frac{\frac{m+2}{2^{14}} \cdot X1 + (c - 2^{13})}{2^{14}} + V_{\min} \quad (1)$$

Формула перевода напряжения в код для записи в IDR:

$$X1 = \left[\frac{V_{out} - V_{\min}}{V_{\max} - V_{\min}} \cdot 2^{14} + 2^{13} - c \right] \cdot \frac{2^{14}}{m+2} \quad (2)$$

где:

Vmax – верхняя граница выбранного диапазона напряжений,

Vmin – нижняя граница выбранного диапазона напряжений

[Vmin, Vmax] = **[-10В,+10В]**; **[-5В,+5В]**; **[0,+10В]**, соответственно для 3-х диапазонов.

X1 - 14-битное слово, записываемое пользователем для выдачи в ЦАП.

m - 14-битный коэффициент усиления (значение по умолчанию 3FFEh).

c - 14-битный коэффициент смещения (значение по умолчанию 2000h).

Управление каналами осуществляется путем настройки регистров AD5392:

1. **Регистр данных каналов** Input Data Register (**IDR**), содержащий **X1** – 14-битное слово, записываемое пользователем для выдачи в выходные каналы ЦАП.
2. **Регистр смещения** Offset Register (**OR**), содержащий коэффициент смещения **c**.
3. **Регистр коэффициента усиления** Gain Register (**GR**), содержащий коэффициент усиления **m**.
4. **Регистры специальных функций** Special Function Registers (SFRs).

● **Регистр данных канала IDR** содержит код, выдаваемый на аналоговый выход ЦАП в соответствии с формулой (1). В таблице ниже представлены уровни выходов ЦАП в зависимости от значения регистра **IDR** для случая, когда коэффициенты смещения и усиления имеют значения по умолчанию:

Таблица 1.

Данные D13..D0 регистра IDR X1 , bin\dec				ВЫХОДЫ ЦАП Vout (В)		
				Для диапазона $\pm 10В$	Для диапазона $\pm 5В$	Для диапазона $0В..+10В$
111111	1111	1111	16383	$20 \cdot (16383/16384) - 10$	$10 \cdot (16383/16384) - 5$	$10 \cdot (16383/16384)$
111111	1111	1110	16382	$20 \cdot (16382/16384) - 10$	$10 \cdot (16382/16384) - 5$	$10 \cdot (16382/16384)$
⋮				⋮	⋮	⋮
100000	0000	0001	8193	$20 \cdot (8193/16384) - 10$	$10 \cdot (8193/16384) - 5$	$10 \cdot (8193/16384)$
100000	0000	0000	8192	$20 \cdot (8192/16384) - 10$	$10 \cdot (8192/16384) - 5$	$10 \cdot (8192/16384)$
011111	1111	1111	8191	$20 \cdot (8191/16384) - 10$	$10 \cdot (8191/16384) - 5$	$10 \cdot (8191/16384)$
⋮				⋮	⋮	⋮
000000	0000	0001	1	$20 \cdot (1/16384) - 10$	$10 \cdot (1/16384) - 5$	$10 \cdot (1/16384)$
000000	0000	0000	0	0	0	0

● **Регистр смещения OR** содержит поправочный коэффициент смещения. Для каждого канала существует свой регистр смещения. В таблице представлены соответствие данных регистра **OR** значению коэффициентов смещения **C**.

Таблица 2.

Данные D13..D0 регистра OR	Коэффициент смещения C
111111 1111 1111	+8192
111111 1111 1110	+8191
⋮	⋮
100000 0000 0001	+1
100000 0000 0000 (default)	+0
011111 1111 1111	-1
⋮	⋮
000000 0000 0001	-8191
000000 0000 0000	-8192

● **Регистр коэффициента усиления GR**. Для каждого канала существует свой регистр коэффициента усиления. В таблице представлено соответствие данных регистра **GR** коэффициентам усиления **m**

Таблица 3.

Данные D13..D0 регистра GR	Коэффициент усиления m
11 1111 1111 1110 (default)	1
⋮	⋮
10 1111 1111 1110	0,75
⋮	⋮
01 1111 1111 1110	0,5
⋮	⋮
00 1111 1111 1110	0,25
⋮	⋮
00 0000 0000 0000	0

● **Зона регистров специальных функций SFRs**

Микросхема ЦАП AD5392, используемая в модуле, содержит SFR регистры специальных функций, как показано в таблице 4. Данные для записи в зону SFRs устанавливаются в регистр DACDATA, а затем доступ к регистрам/командам осуществляется по адресу, установленному в регистре DACADR:

Таблица 4.

rd/wr	DACA3	DACA2	DACA1	DACA0	Операция
wr	0	0	0	0	NOP (нет операции)
wr	0	0	0	1	Запись в регистр CLR code Register
wr	0	0	1	0	Программная очистка (Soft CLR)
wr	1	1	0	0	Запись в регистр Channel range A register
wr	1	0	1	0	Запись в регистр Channel range B register
wr	1	1	1	1	Программный сброс (Soft reset)

Описание специальных функций SFR:

1. NOP. (Нет операций.)

DACADR= *0000h*

Не выполняется никаких операций.

Во время выполнения операции бит DACSTAT[7]= **BUSY**= «0».

2. Запись в регистр CLR Code (Write CLR Code register)

DACADR= *0001h*

DACDATA= **CLR Code**.

Назначение регистра CLR Code – хранение данных, которые потом при выполнении функции программной очистки Soft CLR могут быть загружены в каналы ЦАП. Таким образом, запись значения в регистр **CLR Code** используется в сочетании с функцией программной очистки Soft CLR (см. ниже п.3.). Выполнение функции *программной очистки (Soft CLR)* загружает содержимое регистра **CLR Code** в выходные регистры DAC и устанавливает соответствующие напряжения Vout0..Vout7 на выходах ЦАП.

Запись **CLR Code** регистров может служить для быстрой калибровки ЦАП. Для выполнения быстрой калибровки пользователь может загрузить полную шкалу или нулевую шкалу в регистр **CLR Code** и затем выполнить *программную очистку* для загрузки кода во все каналы ЦАП, избежав, таким образом, индивидуальной записи во все каналы, которая может занять продолжительное время. Затем в соответствии с данными, полученными на АЦП произвести настройку значений коэффициента усиления и смещения для каждого канала.

По умолчанию в **CLR Code** регистр записаны нули.

3. Программная очистка (Soft CLR).

DACADR= *0002h*

DACDATA = *не важно*

При выполнении этой операции осуществляется *программная очистка* ЦАП. При этом на выходы ЦАП подаются данные, содержащиеся в регистре **CLR Code**. Время, требуемое для выполнения этой команды – 15мкс.

Во время выполнения операции бит DACSTAT[7]= **BUSY**= «1».

4. Настройка диапазонов каналов Channel Range A (Write Channel range A register)

DACADR= *000Ch*

DACDATA = **CRA**

CRA = *1600h* – диапазон [-10В ... +10 В].

CRA = *0600h* – диапазон [-5В ... +5 В] или [0В ... 10 В].

5. Настройка диапазонов каналов Channel Range B (Write Channel range B register).

DACADR= *000Ah*

DACDATA = **CRB**

CRB = *2400h* – диапазон [-10В ... +10 В] или [-5В ... +5 В].

CRB = *2500h* – диапазон [0В ... 10 В].

Сочетание значений в регистрах **CRA** и **CRB** однозначно определяют диапазоны работы ЦАП. Значения, которые необходимо записать в эти регистры для задания диапазона сведены в таблицу:

Значения регистров диапазона	Диапазон		
	-10 В ... +10 В	- 5 В ... +5 В	0 В ... +10 В
Channel range A [DA13..DA0]	<i>1600h</i>	<i>0600h</i>	<i>0600h</i>
Channel range B [DA13..DA0]	<i>2400h</i>	<i>2400h</i>	<i>2500h</i>

6. Программный сброс (Soft Reset).

DACADR= *000Fh*

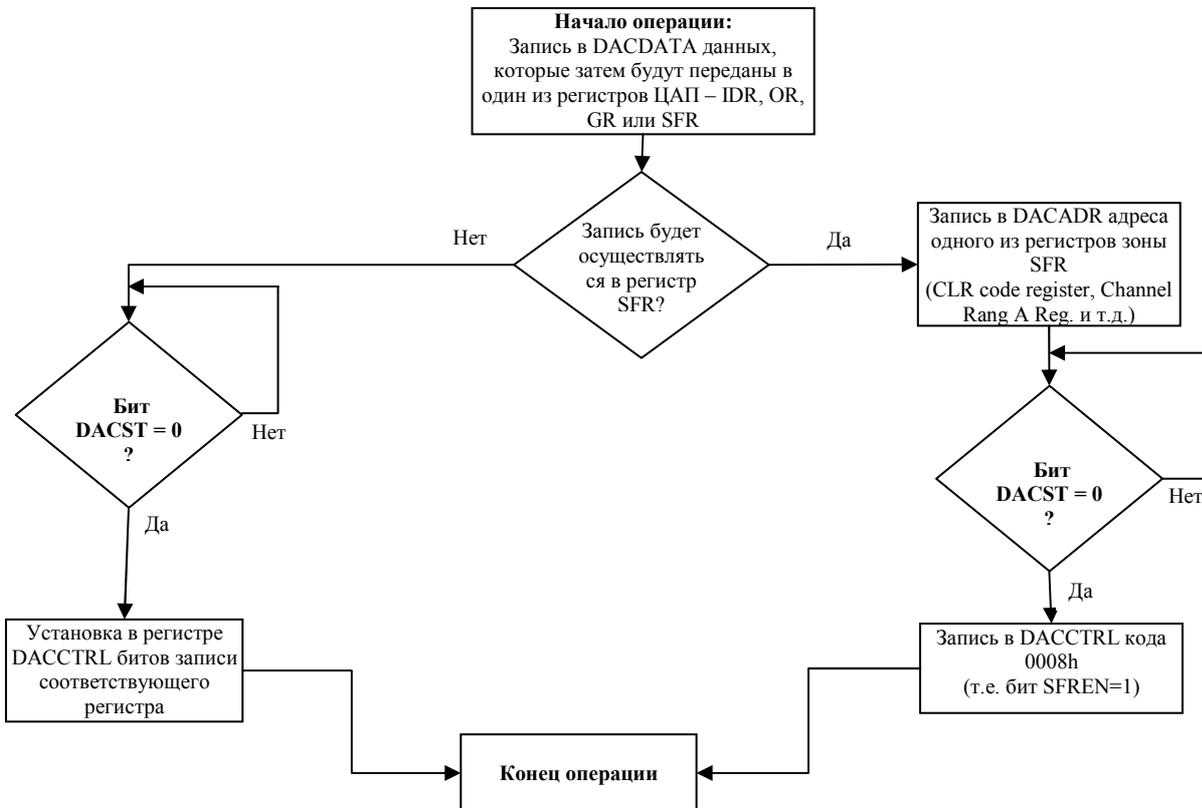
DACDATA = *не важно*

Команда необходима для выполнения *программного сброса*. Все внутренние регистры ЦАПа сбрасываются в значения по умолчанию. Содержимое DAC регистров очищается, на аналоговых выходах устанавливается напряжение 0V. Время исполнения команды программного сброса $\leq 135\mu\text{с}$.

14. Работа с ЦАП AD5392

Работа с регистрами ЦАП AD5392 осуществляется через регистры модуля **DACDATA**, **DACADR**, **DACCTRL** и **DACSTATE**.

Общий механизм записи данных в ЦАП выглядит следующим образом:



Регистр **DACCFG** содержит справочную информацию о количестве доступных для работы каналов ЦАП:

Формат регистра данных **DACCFG (RA = 20h)**:

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	DACH₇	DACH₆	DACH₅	DACH₄	DACH₃	DACH₂	DACH₁	DACH₀
Доступ	rd	rd	rd	rd	rd	rd	rd	rd								

Если **DACH_i** = «1» («0»), то это означает, что i-й канал ЦАП установлен (не установлен) на плате, где $i = [0...7]$.

Запись данных в ЦАП реализована через регистр данных **DACDATA**:

Формат регистра данных **DACDATA (RA = 03h)**:

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	DA₁₃	DA₁₂	DA₁₁	DA₁₀	DA₉	DA₈	DA₇	DA₆	DA₅	DA₄	DA₃	DA₂	DA₁	DA₀
Доступ	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr

DA[13..0] - данные для загрузки в регистры выбранного канала ЦАП.

Регистр **DACADR** используется только для задания кодов операций

Выбор адреса регистра/команды SFR производится записью значения в регистр адреса **DACADR**:

Формат регистра адреса **DACDADR (RA = 04h)**:

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	0	0	0	0	DACA3	DACA2	DACA1	DACA0
Доступ	wr	wr	wr	wr												

Битами **DACA[3..0]** задается адреса регистра/команды SFR в соответствии с таблицей 4.

Выбор регистра ЦАП (SFRs, **GR**, **OR** или **IDR**) осуществляется через регистр **DACCTRL**

Формат регистра управления **DACCTRL (RA = 05h)**:

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	DAEN	OFEN	GFEN	DALD	SFREN	DACH2	DACH1	DACH0
Доступ	wr	wr	wr	wr	wr	wr	wr	wr								

При установке «1» в соответствующие флаги регистра, происходит запись данных в регистры канала ЦАП:
DACH[2..0] - номер канала ЦАП.

DAEN= «1» - записать данные **DA[13..0]** в регистр **IDR** данных канала **DACH[2..0]**.

GFEN= «1» - записать данные **DA[13..0]** в регистр **GR** коэффициента канала **DACH[2..0]**.

OFEN= «1» - записать данные **DA[13..0]** в регистр **OR** смещения канала **DACH[2..0]**.

SFREN= «1» - записать данные **DA[13..0]** в SFR регистр ЦАП с адресом **DACA[3..0]** (в регистре **DACADR**).

DALD = «1» - обновлять выходы каналов ЦАП.

Примечание. При установке бита **DALD = «1»** на выходы ЦАП передается значение напряжения, код которого содержится в регистрах **IDR**, каждого из каналов. Пользователю может быть необходимо изменить напряжения на всех каналах ЦАП одновременно. В этом случае сначала требуется записать во все регистры **IDR[2..0]** необходимые коды напряжений, каждый раз устанавливая **DALD = «0»**, а при последней записи установить **DALD = «1»**. Напряжения при этом изменятся одновременно на всех каналах.

Контроль занятости ЦАП осуществляется через регистр статуса **DACST**

Формат регистра статуса **DACST (RA = 05h)**:

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	BUSY	0	0	0	0	DACH2	DACH1	DACH0
Доступ	rd	rd	rd	rd	rd	rd	rd	rd								

BUSY= «1» - признак занятости ЦАП (формируется последовательная диаграмма записи в регистры ЦАП или отсчитывается задержка для установки выхода ЦАП).

Перед каждой записью в ЦАП необходимо либо проверять, что бит **BUSY= «0»** (ЦАП не занят), либо делать гарантированную задержку между записями в ЦАП.

DACH[2..0] – отображение текущего состояния соответствующих битов регистра **DACCTRL**.

15. Типовые алгоритмы работы с ЦАП AD5392

Для начала работы с ЦАП модуля обязательна инициализация: установка диапазона ЦАП в регистрах Channel Range A (CRA) и Channel Range B (CRB) регистров специальных функций зоны SFRs.

Основные функции работы с ЦАП.

1. Инициализация каналов ЦАП:

1. Записать в регистр **DACDATA** данные регистра **CRA** = 1600h ($\pm 10V$), или 0600h ($\pm 5V$, 0..+10V)
2. Записать в регистр **DACDADR** адрес регистра **CRA** = 000Ch
3. Дождаться обнуления бита **BUSY** регистра **DACST**
4. Записать в регистр **DACCTRL** код записи данных в регистр зоны SFRs :

0	0	0	0	SFREN=1	0	0	0
---	---	---	---	---------	---	---	---

5. Записать в регистры **DACDATA** данные регистра **CRB** = 2400h ($\pm 10V$, $\pm 5V$), или 2500h (0..+10V)
6. Записать в регистр **DACDADR** адрес регистра **CRB** = 000Ah
7. Дождаться обнуления бита **BUSY** регистра **DACST**
8. Записать в регистр **DACCTRL** код записи данных в регистр зоны SFRs :

0	0	0	0	SFREN=1	0	0	0
---	---	---	---	---------	---	---	---

2. Одиночная запись данных в канал ЦАП:

1. Записать в регистр **DACDATA** данные регистра **IDR**
2. Дождаться обнуления бита **BUSY** регистра **DACST**
3. Записать в регистр **DACCTRL** код записи данных в регистр **IDR**:

DAEN=1	0	0	DALD	0	DACH2	DACH1	DACH0
--------	---	---	------	---	-------	-------	-------

DACH[2..0]=номер канала ЦАП (0..7)

DALD – обновлять - «1»/не обновлять - «0» выходы каналов ЦАП

Примечание. Пользователю необходимо учитывать время между записью данных в канал ЦАП и установкой соответствующего напряжения. Согласно описанию микросхемы AD5392 время изменения напряжения на выходах ЦАП со значения в $1/4$ диапазона до значения в $3/4$ диапазона составляет порядка 10 мкс.

Дополнительные функции работы с ЦАП.

1. Запись в регистр смещения канала ЦАП:

1. Записать в регистр **DACDATA** значение смещения для регистра **OR**
2. Дождаться обнуления бита **BUSY** регистра **DACST**
3. Записать в регистр **DACCTRL** код записи данных в регистр **OR**:

0	OFEN=1	0	0	0	DACH2	DACH1	DACH0
---	--------	---	---	---	-------	-------	-------

DACH[2..0]=номер канала ЦАП (0..7)

2. Запись в регистр коэффициента усиления канала ЦАП:

1. Записать в регистр **DACDATA** значение смещения для регистра **GR**
2. Дождаться обнуления бита **BUSY** регистра **DACST**
3. Записать в регистр **DACCTRL** код записи данных в регистр **GR**:

0	0	GFEN=1	0	0	DACH2	DACH1	DACH0
---	---	--------	---	---	-------	-------	-------

DACH[2..0]=номер канала ЦАП (0..7)

3. Запись регистра зоны SFRs ЦАП:

1. Записать в регистр **DACDATA** данные регистра зоны SFRs
2. Записать в регистр **DACDADR** адрес регистра/команды зоны SFRs
3. Дождаться обнуления бита **BUSY** регистра **DACST**
4. Записать в регистр **DACCTRL** код записи данных в регистр зоны SFRs :

0	0	0	0	SFREN=1	0	0	0
---	---	---	---	---------	---	---	---

16. Описание АЦП

В зависимости от исполнения модуль может содержать до 16 недифференциальных входных аналоговых каналов или до 8 дифференциальных аналоговых каналов при установке 2 м/сх AD7328 АЦП1 и АЦП2 (до 8 недифференциальных входных аналоговых каналов или до 4 дифференциальных аналоговых каналов при установке 1 м/сх AD7328) и выполнять кодирование поступающей от них информации. Микросхемы типа AD7328 имеют разрешающую способность 13 разрядов (12 разрядов + знак). Каналы измерения АЦП мультиплексированы. Номер канала АЦП для преобразования задается программно. Также программно задается тип подключения аналогового сигнала – недифференциальное или дифференциальное. Для каждого канала АЦП диапазон входного напряжения может быть запрограммирован индивидуально. Доступны диапазоны входных напряжений $\pm 10\text{В}$, $\pm 5\text{В}$, $\pm 2.5\text{В}$, $0..+10\text{В}$. Максимальная допустимая частота оцифровки по одному каналу для модуля DD64-PCI составляет 400 КГц.

В соответствии с задачей пользователя возможны 4 режима измерения входов каждой из АЦП: 8 одиночных входов, 4 пары дифференциальных, 7 псевдо-дифференциальных и 4 пары псевдо-дифференциальных входов.

• Режим одиночных входов (режим по умолчанию).

На базе одной микросхемы АЦП в одиночном режиме пользователь имеет доступ к измерению до 8 одиночных входов АЦП относительно аналоговой земли AGND (до 16 одиночных входов при установке 2 м/сх AD7328 в зависимости от исполнения). Каждый аналоговый вход может быть независимо запрограммирован на один из четырех диапазонов напряжений $\pm 10\text{В}$, $\pm 5\text{В}$, $\pm 2.5\text{В}$, $0..+10\text{В}$. Аналоговые земли AGND для АЦП1 и АЦП2 объединяется с сигнальной землей для всех 8-и (16-и) входов Vin1_0..Vin1_7 (Vin2_0..Vin2_7). Измерение происходит относительно аналоговой земли AGND. В общем случае, по этой причине имеют место помехи по постоянному току и синфазные помехи, но поскольку аналоговая земля AGND АЦП модуля DD64-PCI гальванически независима, происходит погашение помех по постоянному току.

Достоинства. Наиболее простой режим (по умолчанию)

Недостатки. Наименее помехозащищенный режим.

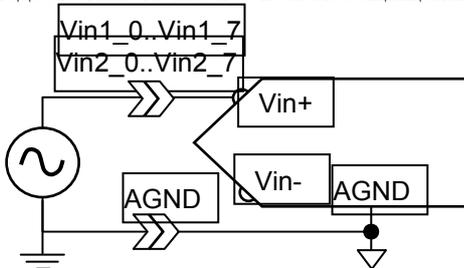


Рис. Подключение аналоговых входов к АЦП в режиме одиночного входа.

• Дифференциальный режим.

Входы Vin1_0..Vin1_7, Vin2_0..Vin2_7 могут быть оцифрованы как 8 дифференциальных аналоговых входных пар (см. рис). Пары образованы четным и нечетным каналами Vin (например, дифференциальная пара Vin1_0-Vin1_1) см. таблицу 2.

Дифференциальный режим предназначен для измерения симметричного дифференциального сигнала, т.е. сигнала имеющего сдвиг фаз на 180° , при этом уровни таких сигналов одинаковы и отсчитываются от средней точки, потенциал которой определяется как значение $(\text{Vin}+ - \text{Vin}-)/2$. Максимальный измеряемый уровень входного дифференциального сигнала определяется как разность между значениями Vin+ и Vin- ($\text{Vin}+ - \text{Vin}-$), т.е. максимальная амплитуда составляет $2 \times \text{Vin}$. К примеру для диапазона $\pm 10\text{В}$ максимальная измеряемая амплитуда дифф. сигнала составит 20В. Амплитуды же каждого из сигналов на входах Vin+ и Vin- могут достигать 20В но при условии, что сигналы сдвинуты на 180° и симметричны относительно нулевой точки, амплитуда их дифференциального значения никогда не будет превышать 20В.

Дифференциальный вход обеспечивает подавление постоянной составляющей сигнала, а также синфазной помехи.

Достоинства. Наибольшая точность и помехозащищенность, независимость измерений между парами дифференциальных сигналов.

Недостатки. Необходимость в задействовании 2 входов АЦП для измерения сигналов на 1 канале.

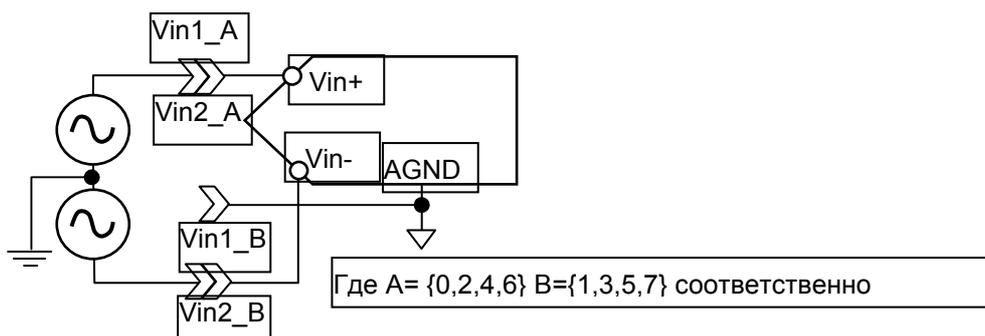


Рис. Подключение аналоговых входов к АЦП в дифференциальном режиме (один канал).

● **Псевдо-дифференциальный режим.**

АЦП модуля может иметь 4 псевдо-дифференциальных пары входов или семь одиночных псевдо-дифференциальных входов, измеряемых относительно общего входа Vin-. В общем случае, псевдо-дифференциальный режим измерения является улучшенными по сравнению с режимом одиночных входов из-за подавления помех по постоянному току. Но поскольку питание АЦП модуля DD64-PCI гальванически независимо, то преимущества связанные с подавлением этого типа помех псевдо-дифференциальный режим не имеет. В режиме с подключением 4-х пар псевдо-дифференциальных входов измерение одиночных сигналов производится независимо, т.е. каждый одиночный сигнал будет измеряться относительно своей сигнальной пары. Например Vin0 относительно Vin1. В режим с подключением 7-и псевдо-дифференциальных входов Vin0..Vin6 (Vin+) измерение происходит относительно Vin7 (Vin-). В отличие от режима дифференциального измерения в псевдо-дифференциальном режиме не происходит подавления синфазных помех.

17. Работа с АЦП AD7328

Работа с АЦП модуля DD64-PCI осуществляется путем настройки регистров управления АЦП1 и АЦП2, регистров диапазона 1 АЦП1 и АЦП2, регистров диапазона 2 АЦП1 и АЦП2 **микросхем АЦП AD7328**. Доступ к этим регистрам осуществляется через регистры ADCDATA, ADCST, ADCCTRL контроллера DD64 для работы с АЦП1 и через регистры ADC2DATA, ADC2ST, ADC2CTRL контроллера DD64 для работы с АЦП2, которые в свою очередь доступны по PCI. Механизмы доступа к регистрам м/сх АЦП AD7328 описаны ниже.

Регистры микросхемы АЦП1 и АЦП2 типа AD7328

- регистр управления – Control Register.** Содержит настройки **конфигурации аналоговых выходов** (одиночные каналы, дифференциальные каналы и т.д.), а также **способов кодирования выдаваемых данных** (прямая логика или двоичное дополнение).
- регистр диапазона 1 - Range Register 1.** Для каждого из каналов в регистре задается собственный диапазон. Регистр диапазона 1 определяет диапазоны допустимых напряжений каналов Vin0-Vin3
- регистр диапазона 2 - Range Register 2.** Для каждого из каналов в регистре задается собственный диапазон. Регистр диапазона 2 определяет диапазон допустимых напряжений каналов Vin4-Vin7. Эти регистры доступны только для записи.

● **Регистр управления АЦП Control Register.** Регистр управления используется для выбора аналогового канала входа, а также конфигурации аналоговых входов. Регистр управления доступен только для записи. Назначения битов регистра управления указаны в таблице (при подаче питания значение всех битов регистра управления = 0).

Формат регистра управления **Control Register**

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	ADD2	ADD1	ADD0	Mode1	Mode0	0	0	Coding	1	0	0	1
Доступ	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr

Таблица 1. Описание битов регистра управления.

Номер бита	Обозначение	Описание
11, 10, 9	ADD2, ADD1, ADD0	Эти биты адреса канала используется для выбора канала аналогового входа для текущего преобразования.
8, 7	Mode1, Mode0	Эти два бита режима используется для задания конфигурации аналоговых входов, номера которых заданы битами ADD2..ADD0. Эти биты используются в сочетании с битами адреса канала. Аналоговые входы модуля могут конфигурироваться как два несимметричных входа, как один истинно дифференциальный вход, или один псевдодифференциальный вход (см. табл. 2).
6, 5	-/-	0
4	Coding	Этот бит используется для выбора типа выходного кодирования AD7328. Если Coding = 0, выход кодирования – дополнительный код (двоичное дополнение). Если Coding = 1, выход кодирования - прямая двоичная логика.
3	REF	Бит должен быть установлен в 1. В этом случае он задает внутренний источник опорного напряжения. Внешний источник опорного напряжения не предусмотрен.
2, 1	Seq1, Seq2	Эти биты должны быть установлены в 0. В этом случае секвенсор АЦП не используется. Режим использования секвенсора не предусмотрен.
0	Weak/Z-сост	Если бит = 1, DOUT выход слабо подтянут к значению, записанному в бите ADD2 адреса канала.

Для каждого АЦП 8 аналоговых входных каналов могут быть сконфигурированы как 7 псевдодифференциальных входов, 4 псевдо-дифференциальных входа, 4 пары дифференциальных входов или 8 одиночных аналоговых входов.

Таблица 2. Выбор конфигурации аналогового входа.

Бит адреса канала			Mode1=1, Mode0=1		Mode1=1, Mode0=0		Mode1=0, Mode0=1		Mode1=0, Mode0=0	
			7 псевдо-дифф. входов		4 дифференц. входа		4 псевдо-дифф. входа		8 несимметр-х аналог-х входов	
ADD2	ADD1	ADD0	Vin+	Vin-	Vin+	Vin-	Vin+	Vin-	Vin+	Vin-
0	0	0	Vin0	Vin7	Vin0	Vin1	Vin0	Vin1	Vin0	AGND
0	0	1	Vin1	Vin7	Vin0	Vin1	Vin0	Vin1	Vin1	AGND
0	1	0	Vin2	Vin7	Vin2	Vin3	Vin2	Vin3	Vin2	AGND
0	1	1	Vin3	Vin7	Vin2	Vin3	Vin2	Vin3	Vin3	AGND
1	0	0	Vin4	Vin7	Vin4	Vin5	Vin4	Vin5	Vin4	AGND
1	0	1	Vin5	Vin7	Vin4	Vin5	Vin4	Vin5	Vin5	AGND
1	1	0	Vin6	Vin7	Vin6	Vin7	Vin6	Vin7	Vin6	AGND
1	1	1	xxx	xxx	Vin6	Vin7	Vin6	Vin7	Vin7	AGND

• **Регистры диапазона 1, 2 - Range Register 1,2.** Регистры диапазона используются для задания диапазона каждого аналогового входного канала. Это доступные только по записи регистры. Для задания диапазона предназначено по 2 бита на каждый канал. Допустимы 4 входных диапазона аналоговых напряжений: $\pm 10\text{В}$; $\pm 5\text{В}$; $\pm 2,5\text{В}$; $0\text{В}..+10\text{В}$. Для каждого канала аналогового входа по умолчанию выбран диапазон $\pm 10\text{В}$ (см. таблицу 2).

Регистр диапазона 1 служит для задания диапазонов каналов 0..3.

Формат регистра диапазона **Range Register 1**

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	Vin0A	Vin0B	Vin1A	Vin1B	Vin2A	Vin2B	Vin3A	Vin3B	0	0	0	0
Доступ	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr

Регистр диапазона 2 служит для задания диапазонов каналов 4..7.

Формат регистра диапазона **Range Register 2**

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	Vin4A	Vin4B	Vin5A	Vin5B	Vin6A	Vin6B	Vin7A	Vin7B	0	0	0	0
Доступ	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr

Таблица 3. Выбор диапазона входных каналов АЦП.

VINxA	VINxB	Описание
0	0	Входной диапазон $\pm 10\text{В}$
0	1	Входной диапазон $\pm 5\text{В}$
1	0	Входной диапазон $\pm 2,5\text{В}$
1	1	Входной диапазон от 0 до 10 В

Доступ к регистрам АЦП осуществляется через регистры **контроллера ЦАП/АЦП** находящихся в зоне косвенной адресации:

Регистры данных – **ADCDATA** (Read/Write, RA=06h) для **АЦП1** и **ADC2DATA** (Read/Write, RA=10h) для **АЦП2**,
Регистры управления данными – **ADCCTRL** (Write only, RA=07h) для **АЦП1**, **ADC2CTRL** (Write only, RA=11h) для **АЦП2**

Регистры состояния – **ADCST** (Read only, RA=07h) для **АЦП1**. **ADC2ST** (Read only, RA=11h) для **АЦП2**

Для того, чтобы записать данные в один из регистров АЦП (Control Register, Range Registers 1, 2) необходимо сначала записать их в регистр ADCDATA (RA=06), а затем в регистре ADCCTRL задать команду для передачи этих данных в нужный регистр АЦП. Команда запуска преобразования АЦП также записывается в регистр ADCCTRL и имеет наивысший приоритет по отношению к командам записи данных.

В том случае если была дана команда на преобразование, то результат преобразования можно будет прочитать в регистре **ADCDATA** только после снятия бита занятости АЦП в регистре **ADCST**. Таким образом, регистр ADCDATA по записи передает настройки в АЦП, а по чтению возвращает результаты преобразования.

Общий механизм обмена отображен на диаграмме:

Подробно последовательности обращения к регистрам АЦП описаны в разделе «**Типовые алгоритмы работы с АЦП AD7328**»

Регистр **ADCCFG** содержит справочную информацию о количестве доступных для работы каналов АЦП:

Формат регистра данных **ADCCFG (RA = 21h)**:

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	ADCH2	ADCH1														
Доступ	rd															

Если **ADCH1_i** = «1» («0») означает, что i-й канал АЦП1 установлен (не установлен) на плате, где i = [0...7] .

Если **ADCH2_i** = «1» («0») означает, что i-й канал АЦП2 установлен (не установлен) на плате, где i = [0...7] .

Формат регистра данных **ADCDATA (Write, RA=06h), ADC2DATA (Write, RA=10h)**

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Доступ	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	wr	Wr

AD[11..0] - данные для загрузки в регистры выбранного канала АЦП.

Формат регистра данных **ADCDATA. (Read, RA=06h) ADC2DATA. (Read, RA=10h)**

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	ADCH2	ADCH1	ADCH0	SIGN	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Доступ	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd	rd	Rd

В регистре содержится результат преобразования АЦП:

ADCH[2..0] - номер канала, **SIGN** - знаковый бит, **ADD[11..0]** - разряды данных.

Формат регистра управления данными **ADCCTRL. (Write only, RA=07h) ADC2CTRL. (Write only, RA=11h)**

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	ADEN	CREN	RR1EN	RR2EN	0	0	0	0
Доступ	wr	wr	wr	wr	wr	wr	wr	Wr								

После установки бит регистра управления происходит запись данных в регистры АЦП и/или запуск преобразования.

ADEN = “1” – запустить преобразование АЦП.

CREN = “1” – записать в **Control Register** АЦП.

RR1EN = “1” – записать в **Range Register 1** АЦП.

RR2EN = “1” – записать в **Range Register 2** АЦП.

Бит ADEN имеет более высокий приоритет, т.е. если одновременно дается команда на преобразование и устанавливается один из битов ADCCTRL[6..4], то их значение игнорируется.

ВНИМАНИЕ! В случае, когда **ADEN** = “0” и одновременно установлены биты **ADCCTRL[6..4]** - результат непредсказуем.

Формат регистра состояния **ADCST. (Read only, RA=07h) ADC2ST. (Read only, RA=11h)**

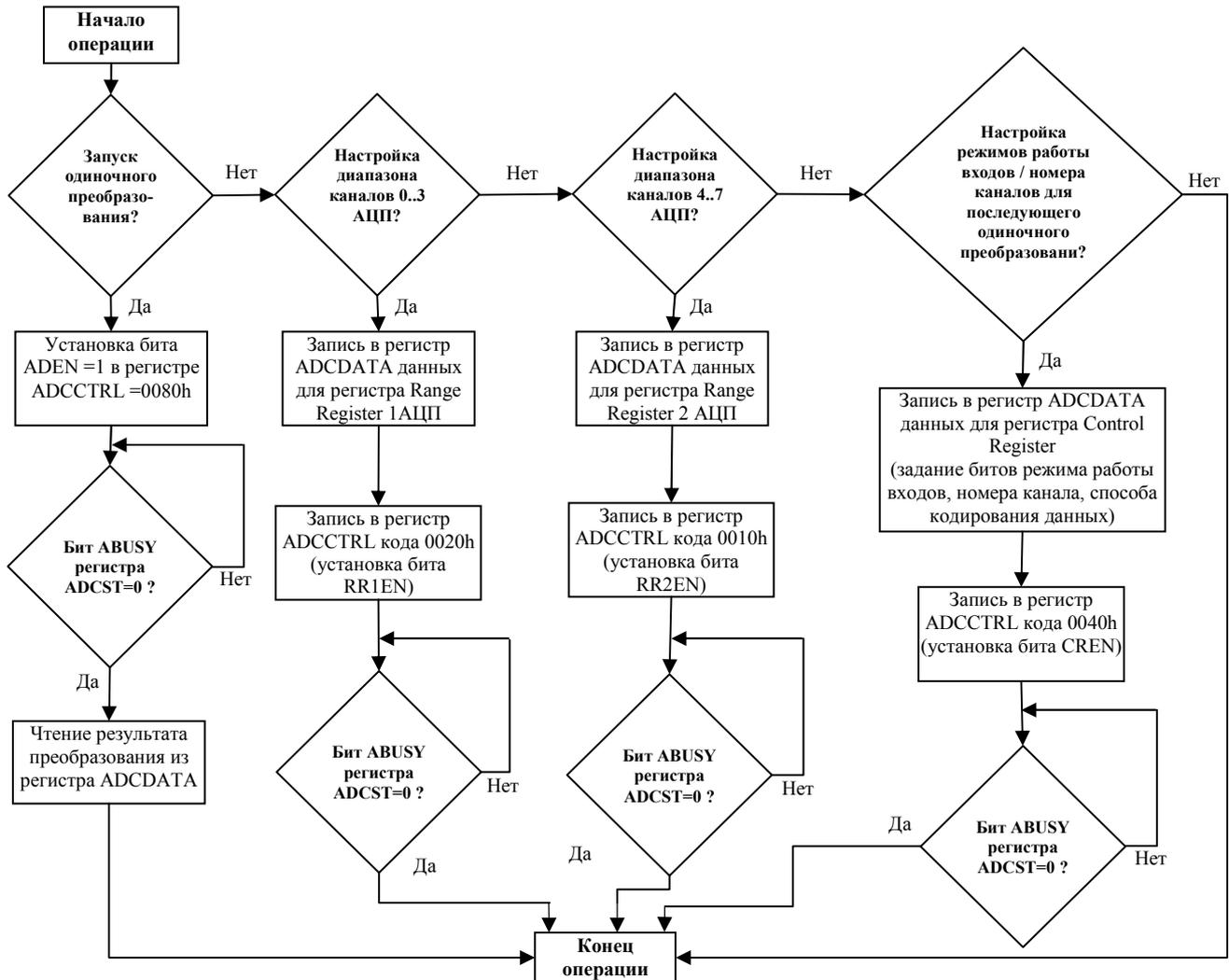
№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	ABUSY	0	0	0	0	0	0	0
Доступ	rd	rd	rd	rd	rd	rd	rd	Rd								

ABUSY=1 - признак занятости АЦП. Перед каждой записью в АЦП либо чтением результата преобразования необходимо проверять значение бита **ABUSY=0** (АЦП не занят). Другой способ корректного обмена с АЦП - это выдерживать гарантированную задержку между записью/чтением регистров.

18. Типовые алгоритмы работы с АЦП AD7328

На диаграмме представлен общий алгоритм работы с АЦП AD7328 на примере АЦП1 через регистры контроллера DD64. Алгоритмы работы для АЦП2 аналогичны с тем различием, что вместо регистров ADCDATA, ADCCTRL и ADCST используются регистры ADC2DATA, ADC2CTRL, ADC2ST.

В разделе представлены последовательности выполнения обращений к *контроллеру ЦАП/АЦП DD64* для выполнения типовых функций АЦП:



• Задание диапазонов входных каналов 0..3 АЦП1 (АЦП2):

1. Записать в регистры ADC(2)DATA, RA=06h (RA=10h для АЦП2) данные для загрузки в **Range Register 1** АЦП1 (АЦП2).

ADC(2)DATA=>

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	Vin0A	Vin0B	Vin1A	Vin1B	Vin2A	Vin2B	Vin3A	Vin3B	0	0	0	0

Значения VinxA, VinxB см. в таблице 2.

2. Записать в регистр ADC(2)CTRL, RA=07h (RA=11h для АЦП2) следующий код:

ADC(2)CTRL=>

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	0	0	RR1EN=1	0	0	0	0	0

3. Дождаться, пока не снимется признак занятости АЦП1 (АЦП2) (бит ABUSY регистра ADC(2)ST).

ADC(2)ST<=

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	ABUSY=(1 --> 0)	0	0	0	0	0	0	0

• **Задание диапазонов входных каналов 4..7 АЦП1 (АЦП2):**

1. Записать в регистры **ADC(2)DATA**, RA=06h (RA=10h для АЦП2) данные для загрузки в **Range Register 2** АЦП1 (АЦП2).

ADC(2)DATA=>

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	Vin4A	Vin4B	Vin5A	Vin5B	Vin6A	Vin6B	Vin7A	Vin7B	0	0	0	0

2. Записать в регистр **ADC(2)CTRL**, RA=07h (RA=11h для АЦП2) следующий код:

ADC(2)CTRL=>

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	0	0	0	RR2EN=1	0	0	0	0

3. Дождаться, пока не снимется признак занятости АЦП1 (АЦП2) (бит **ABUSY** регистра **ADC(2)ST**).

ADC(2)ST<=

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	ABUSY=(1 --> 0)	0	0	0	0	0	0	0

• **Установка режима работы входов АЦП и номера канала для преобразования**

1. Записать в регистры **ADC(2)DATA**, RA=06h (RA=10h для АЦП2) данные для загрузки в **Control Register** АЦП1 (АЦП2), указав номер канала **ADD[2..0]**, режим для входов **Mode[1..0]** и способ кодирования **Coding**:

ADC(2)DATA=>

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	ADD2	ADD1	ADD0	Mode1	Mode0	0	0	Coding	1	0	0	1

2. Записать в регистр **ADC(2)CTRL**, RA=07h (RA=11h для АЦП2) следующий код:

ADC(2)CTRL=>

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	0	CREN = 1	0	0	0	0	0	0

3. Дождаться, пока не снимется признак занятости АЦП (бит **ABUSY** регистра **ADC(2)ST**).

ADC(2)ST<=

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	ABUSY=(1 -> 0)	0	0	0	0	0	0	0

• **Одиночный запуск преобразования АЦП:**

(перед запуском преобразования необходимо установить диапазон входных каналов, режим работы и номер канала, напряжение на котором будет преобразовано в цифровой код)

5. Записать в регистр **ADC(2)CTRL** (RA=07h) код запуска преобразования АЦП1 (АЦП2):

ADC(2)CTRL=>

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	ADEN=1	0	0	0	0	0	0	0

6. Дождаться, пока не снимется признак занятости АЦП1 (АЦП2) (бит **ABUSY** регистра **ADC(2)ST**).

ADC(2)ST<=

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	0	0	0	0	0	0	0	0	ABUSY=(1 --> 0)	0	0	0	0	0	0	0

7. Прочитать из регистра **ADC(2)DATA**, RA=06h (RA=10h для АЦП2) результат преобразования.

ADC(2)DATA<=

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	ADCH2	ADCH1	ADCH0	SIGN	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Примечание. Если пользователю необходимо оцифровывать напряжение с одного канала, то для осуществления преобразования необходимо каждый раз запускать **Одиночный запуск преобразования АЦП**. Если пользователю необходимо оцифровывать напряжение с разных каналов, то перед **Одиночным запуском преобразования** нужно установить **Режим работы входов и номер канала для преобразования**.

19. Работа с таймером.

В модуле DD64-PCI реализован аппаратный 16-ти разрядный таймер, тактируемый от 16-ти разрядного делителя частоты, с диапазоном периода тактирования 1-65536 мкс (1 МГц – 15,259 Гц).

Основные режимы работы таймера:

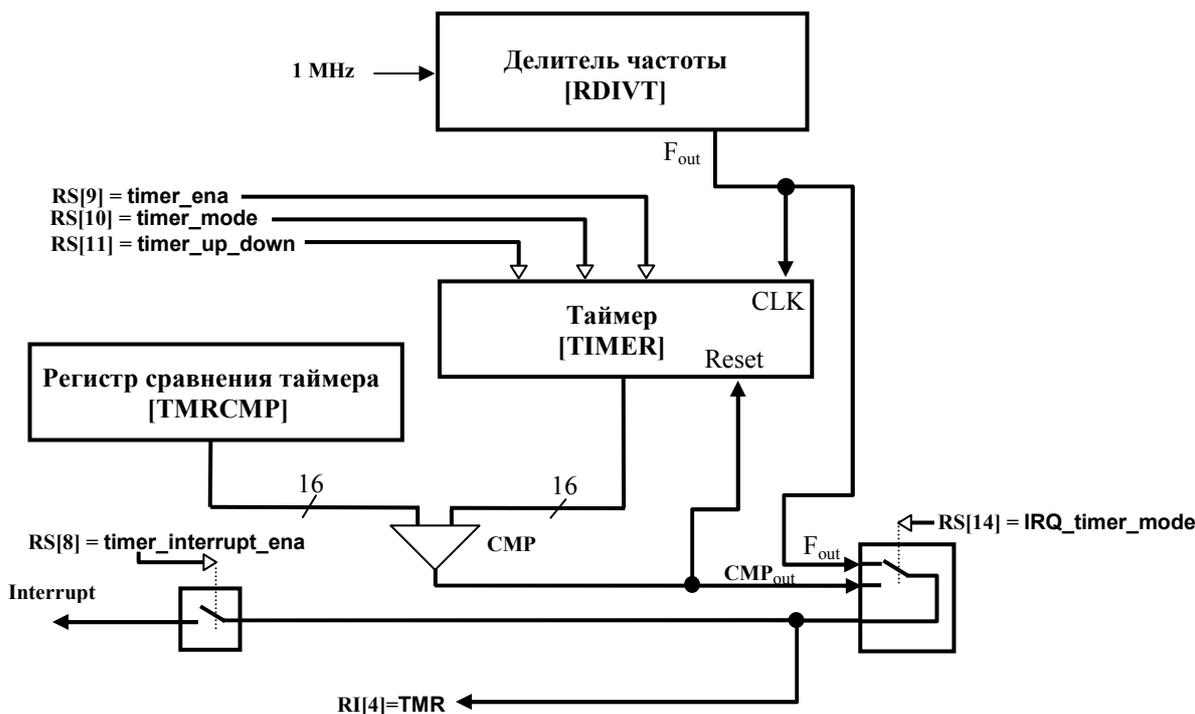
- **Режим генерации регулярных прерываний без контроля пропуска прерываний.** Генерация прерываний происходит через определенные, задаваемые пользователем интервалы времени. Такой генератор регулярных прерываний может быть использован для тактирования временных интервалов в задачах реального времени. Длительность одного интервала задается программно и состоит из N тактов, где $N = [1...65536]$, а длительность одного такта может варьироваться от 1 мкс до 65536 мкс и также задается программно. При работе таймера в этом режиме не исключена ситуация, когда прерывания будут поступать слишком часто. В этом случае программа пользователя может не успевать обработать одно прерывание, когда уже сгенерировано новое прерывание. В таком случае факт пропуска прерывания установить невозможно.

- **Режим генерации регулярных прерываний с контролем пропуска прерываний.** Также как и в первом режиме, генерация происходит через определенные, задаваемые пользователем интервалы времени. Однако в отличие от первого режима длительность одного интервала не может превышать 65536 мкс. Также в этом режиме предусмотрена возможность контролировать пропуск прерывания от таймера.

- **Режим измерения промежутков времени.** Точность измерения задается программно и варьируется от 1 мкс до 65536 мкс (настраиваемая длительность одного такта таймера). Используя только аппаратные ресурсы модуля, пользователь может измерять промежутки времени длительностью 65536 тактов таймера. Используя и программные ресурсы ЭВМ, пользователь может измерять сколь угодно длительные промежутки времени с точностью до 1 мкс.

Реализация каждого из режимов будет описана ниже.

Внутренняя организация таймера представлена на рисунке:



Работа с таймером осуществляется через регистры RS, TIMER, RDI VT, TMRCMP, RI.

Общий принцип работы таймера.

На делитель частоты поступает частота 1 МГц. В зависимости от значения, записанного в регистре делителя частоты, на выходе делителя оказывается частота F_{out} . Эта частота тактирует таймер и может быть подана на линию IRQ шины PCI в зависимости от настроек регистра RS. Запуск/остановка таймера, циклический/однократный режим работы таймера, счет с инкрементом/декрементом осуществляются установкой битов регистра RS. Таймер может работать в циклическом и однократном режиме. В однократном режиме таймер останавливается, когда достигает значения, записанного в регистре TMRCMP. В циклическом режиме при достижении таймером значения, записанного в регистре сравнения таймера TMRCMP, таймер сбрасывается, и счет тактов начинается заново.

● Таймер, регистр TIMER.

Обмен данными с таймером осуществляется через регистр TIMER. При записи в регистре устанавливается начальное значение таймера. При чтении - возвращает текущее значение.

Формат данных регистра таймера **TIMER (Address= BASE+0Ah)**:

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	TMR15	TMR14	TMR13	TMR12	TMR11	TMR10	TMR9	TMR8	TMR7	TMR6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0
Доступ	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr

где **TMR[15..0]** – значение таймера. *Default state: TMR[15..0] = 0000h*

Управление таймером организовано через регистр настроек **RS**:

RS[9] = timer_ena = 1/0 – запуск/остановка счета таймера.

RS[10] = timer_mode = 1/0 – однократный/циклический режим работы таймера

RS[11] = timer_up_down = 1/0 – счет вниз/вверх.

• Делитель частоты, регистр **RDIVT**.

Делитель частоты формирует частоту тактирования таймера. На делитель поступает частота от генератора равная 1 МГц. На выходе делителя формируется частота, определенная значением, записанным в регистр делителя частоты **RDIVT**.

Формат данных регистра делителя частоты **RDIVT (RA = 12h)**:

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Название бита	DiV15	DiV14	DiV13	DiV12	DiV11	DiV10	DiV9	DiV8	DiV7	DiV6	DiV5	DiV4	DiV3	DiV2	DiV1	DiV0
Доступ	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr	rd/wr

где **DiV[15..0]** – значение для деления частоты 1 МГц.

Default state: DiV[15..0] = 0000h

Частота тактирования таймера на выходе делителя определяется по формуле:

$$F_{out} = 1\,000\,000 / (DiV[15..0] + 1) \text{ Гц,}$$

Таким образом, таймер может тактироваться от минимальной частоты 15.259 Гц (при **RDIVT** = 0xFFFF) до 1 МГц (при **RDIVT** = 0x0000).

• Регистр сравнения таймера **TMRCMP (RA = 02h)**.

Цифровой компаратор **CMP** формирует флаг в регистре прерывания **RI** по совпадению значения таймера с регистром сравнения **TMRCMP**, а также формирует сброс таймера. В случае если в регистре настройки **RS[13] = Global_int_ena = 1**, то будет сгенерировано прерывание.

Значение по умолчанию регистра сравнения таймера **TMRCMP = 0000h**.

• Прерывания таймера

Разрешение прерывания от таймера устанавливается в регистре настроек:

RS[8] = timer_interrupt_ena = 1/0 – разрешение/запрещение формирования прерывания от таймера (для получения прерываний от таймера глобальное прерывание при этом также должно быть разрешено в регистре **RS**).

Таймер имеет два источника прерываний: от компаратора при совпадении значений **TMRCMP** и **TIMER**, а также от делителя частоты **RDIVT**. Активный источник прерываний определяется битом:

RS[14] = IRQ_timer_mode = 1/0 – делитель частоты / компаратор.

Режим, когда прерывания формируются от делителя частоты, позволяет организовать с помощью модуля DD64-PCI аппаратный источник прерываний высокой частоты, который может применяться для построения программ реального времени с малым временем реакции (от 1* до 65536 мкс).

Режим, когда прерывание формируется от регистра сравнения, позволяет построить программу реального времени с большими интервалами отсчетов (до значения $T = 65535 \cdot 1/15,259 = 4294,9$ с).

***ВНИМАНИЕ!** При написании ПО программист должен учитывать возможности контроллера прерываний (**КП**) процессорной платы ЭВМ, т.к. частота поступления прерываний от модуля DD64-PCI не должна превышать максимально возможную частоту обработки прерываний **КП** процессорной платы. В случае если прерывания будут формироваться с частотами близкими к максимальным для используемого в ЭВМ **КП** – это может привести к зависанию системы либо некорректной работе программы.

Инициализация таймера. Для первоначального запуска таймера недостаточно только установить бит **RS[9]=1**. Необходимо также установить значение регистра сравнения таймера **TMRCMP** отличным от 0000h (0000h – значение по умолчанию), т.к. при значении регистра **TMRCMP** равным 0000h, значение таймера **TIMER** всегда будет равно 0000h как в циклическом, так и в однократном режимах, даже если таймер запущен.

Примечание. Пользователю необходимо учитывать, что при подаче питания или при сбросе контроллера DD64 в регистре **RI** бит **TMR** установлен в «1». Этот бит сбрасывается в «0» чтением регистра **TIMER**.

● **Реализация режима генерации регулярных прерываний без контроля пропуска прерываний.**

Длительность одного интервала времени определяется формулой

$$T = (\text{DIVT_value} + 1) \cdot (\text{TMRCMP_value} + 1) \text{ мкс. ,}$$

где **DIVt_value** – значение, записанное в регистре RDIVT,

TMRCMP_value – значение записанное в регистре TMRCMP.

Для инициализации таймера необходимо проделать следующую последовательность действий.

1. Записать в регистр делителя частоты RDIVT значение DIVT_value и в регистр сравнения таймера TMRCMP значение TMRCMP_value исходя из длительности требуемого интервала времени.
2. Установить начальное значение таймера равным нулю.
3. Записать в регистр RS код $x01x.0011.xxxx.xxxx_2$

● **Реализация режима генерации регулярных прерываний с контролем пропуска прерываний.**

Длительность одного интервала времени определяется формулой

$$T = (\text{DIVT_value} + 1) \text{ мкс. ,}$$

где **DIVt_value** – значение, записанное в регистре RDIVT,

Для инициализации таймера необходимо проделать следующую последовательность действий.

1. Записать в регистр делителя частоты RDIVT значение DIVT_value исходя из длительности требуемого интервала времени.
2. Записать в регистр сравнения таймера TMRCMP значение 0xFFFF.
2. Установить начальное значение таймера равным нулю.
3. Записать в регистр RS код $x11x.0011.xxxx.xxxx_2$
4. При поступлении прерывания в обработчике прерываний прочитать регистр таймера TMR. Если значение таймера увеличилось на $n \geq 2$ раз относительно значения предыдущего входа в обработчик прерывания, то это является признаком n-1-кратного пропуска прерываний. В этом режиме, фактически, таймер является аппаратным счетчиком прерываний. Пользователю стоит учитывать, что в этом режиме отбой блокировки прерывания обязателен и происходит по чтению значения таймера: регистра **TIMER**.

● **Режим измерения промежутков времени.**

1. Записать в регистр делителя частоты RDIVT значение DIVT_value, задающее длительность одного такта таймера TMR. В этом случае значение (DIVT_value+1) фактически будет равным точности измерения интервала времени (в мкс).
2. Записать в регистр сравнения таймера TMRCMP значение 0xFFFF.
2. Установить начальное значение таймера равным нулю.
3. Записать в регистр RS код $x11x.0011.xxxx.xxxx_2$
4. При каждом такте делителя частоты значение таймера будет инкрементироваться. При совпадении значения таймера TMR_value со значением регистра сравнения таймера TMRCMP будет генерироваться прерывание. Таким образом, если ожидаемая длительность измеряемого интервала может превысить $(\text{DIVT_value} + 1) \cdot 65536$ мкс, то для измерения таких интервалов времени пользователь может в своей программе создать переменную irq_cnt, содержащую количество вызываемых прерываний от таймера и инкрементируемую при каждом прерывании от таймера.

Длительность измеряемого промежутка времени в таком случае может быть вычислена по формуле:

$$T = \text{DIVT_value} \cdot [10000h \cdot \text{irq_cnt} + \text{TMR_value}]$$

20. Рекомендации по эксплуатации и техническому обслуживанию.

Связь платы с внешними устройствами осуществляется через кабели, изготавливаемые пользователем и подключаемые к соединителям X1, X4 платы, цепи соединителей X1, X4 приведены в таблице .

Модуль устанавливается в свободный PCI-слот ПЭВМ при выключенном питании и занимает **два** слота на задней панели ПЭВМ (в случае, если установлены все 64 РК и все каналы ЦАП-АЦП). Для получения свободного (не разделяемого) прерывания, как правило, надо занимать свободные нижние слоты PCI.

Таблица Цепи внешнего соединителя X1 модуля DD64-PCI

№ контакта	Цепь	№ контакта	Цепь	№ контакта	Цепь	№ контакта	Цепь
1	DIO1	21	GND_DIO1	40	DIO17	60	GND_DIO17
2	DIO2	22	GND_DIO2	41	DIO18	61	GND_DIO18
3	DIO3	23	GND_DIO3	42	DIO19	62	GND_DIO19
4	DIO4	24	GND_DIO4	43	DIO20	63	GND_DIO20
5	DIO5	25	GND_DIO5	44	DIO21	64	GND_DIO21
6	DIO6	26	GND_DIO6	45	DIO22	65	GND_DIO22
7	DIO7	27	GND_DIO7	46	DIO23	66	GND_DIO23
8	DIO8	28	GND_DIO8	47	DIO24	67	GND_DIO24
9	DIO9	29	GND_DIO9	48	DIO25	68	GND_DIO25
10	DIO10	30	GND_DIO10	49	DIO26	69	GND_DIO26
11	DIO11	31	GND_DIO11	50	DIO27	70	GND_DIO27
12	DIO12	32	GND_DIO12	51	DIO28	71	GND_DIO28
13	DIO13	33	GND_DIO13	52	DIO29	72	GND_DIO29
14	DIO14	34	GND_DIO14	53	DIO30	73	GND_DIO30
15	DIO15	35	GND_DIO15	54	DIO31	74	GND_DIO31
16	DIO16	36	GND_DIO16	55	DIO32	75	GND_DIO32
17	ADC1_4	37	AGND_ADC	56	DAC4/ADC2_4	76	AGND_DAC
18	ADC1_3	38	AGND_ADC	57	DAC3/ADC2_3	77	DAC2/ADC2_2
19	ADC1_2	39	-15V	58	AGND_DAC	78	DAC1/ADC2_1
20	ADC1_1			59	+15V		

Таблица Цепи внешнего соединителя X4 модуля DD64-PCI

№ контакта	Цепь	№ контакта	Цепь	№ контакта	Цепь	№ контакта	Цепь
1	DIO33	21	GND_DIO33	40	DIO49	60	GND_DIO49
2	DIO34	22	GND_DIO34	41	DIO50	61	GND_DIO50
3	DIO35	23	GND_DIO35	42	DIO51	62	GND_DIO51
4	DIO36	24	GND_DIO36	43	DIO52	63	GND_DIO52
5	DIO37	25	GND_DIO37	44	DIO53	64	GND_DIO53
6	DIO38	26	GND_DIO38	45	DIO54	65	GND_DIO54
7	DIO39	27	GND_DIO39	46	DIO55	66	GND_DIO55
8	DIO40	28	GND_DIO40	47	DIO56	67	GND_DIO56
9	DIO41	29	GND_DIO41	48	DIO57	68	GND_DIO57
10	DIO42	30	GND_DIO42	49	DIO58	69	GND_DIO58
11	DIO43	31	GND_DIO43	50	DIO59	70	GND_DIO59
12	DIO44	32	GND_DIO44	51	DIO60	71	GND_DIO60
13	DIO45	33	GND_DIO45	52	DIO61	72	GND_DIO61
14	DIO46	34	GND_DIO46	53	DIO62	73	GND_DIO62
15	DIO47	35	GND_DIO47	54	DIO63	74	GND_DIO63
16	DIO48	36	GND_DIO48	55	DIO64	75	GND_DIO64
17	ADC1_8	37	AGND_ADC	56	DAC8/ADC2_8	76	AGND_DAC
18	ADC1_7	38	AGND_ADC	57	DAC7/ADC2_7	77	DAC6/ADC2_6
19	ADC1_6	39	-15V	58	AGND_DAC	78	DAC5/ADC2_5
20	ADC1_5			59	+15V		

Таблица соединений заглушки соединителей X1 (X4) платы DD64-PCI*.

1. «Заглушка» – вилка кабельная прямая DHS-78М с корпусом.
2. Монтаж цепей выполнен в корпусе «Заглушки».

№	Цепь	Название	№	Цепь	Название
1	X 1, 9	DI1-DO9	12	X 43, 51	DI20-D28
2	X 2, 10	DI2-DO10	13	X 44, 52	DI21-D29
3	X 3, 11	DI3-DO11	14	X 45, 53	DI22-D30
4	X 4, 12	DI4-DO12	15	X 46, 54	DI23-D31
5	X 5, 13	DI5-DO13	16	X 47, 55	DI24-D32
6	X 6, 14	DI6-DO14	17	X 17, 56	ADC4_DAC_4
7	X 7, 15	DI7-DO15	18	X 18, 57	ADC4_DAC_3
8	X 8, 16	DI8-DO16	19	X 19, 77	ADC4_DAC_2
9	X 40, 48	DI17-DO25	20	X 20, 78	ADC4_DAC_1
10	X 41, 49	DI18-DO26	21	X 58, 38, 37, 76	AGND_ADAC
11	X 42, 50	DI19-D27	22	X 21,22,23,24, 25,26,27,28, 60,61,62,63, 64,65,66,67, 39	-15V
			23	X 29, 30,31,32,33,34,35,36, 68,69,70,71, 72, 73,74,75, 59	+15V

* - подходит не для всех модификаций.

Приложение 1.

Регистры DD64-PCI

Таб. Зона регистров прямой адресации:

Смещение отн. BASE	Доступ по WR	Доступ по RD
BASE+8h	---	RI регистр прер.
BASE+Ah	TIMER	регистр таймера TIMER
BASE+Ch	RA	регистр адреса ---
BASE+Eh	RD	регистр данных RD

Таб. Зона регистров косвенной адресации:

Доступ к регистрам косвенной адресации осуществляется через регистры адреса RA и данных RD.

Адрес в RA	Рег. доступные по записи (Wr)	Рег. доступные по чтению (Rd)	Адрес в RA	Рег. доступные по записи (Wr)	Рег. доступные по чтению (Rd)
00h	-	RID	30h	EXT_MUX_CFG	Ext Mux Cfg
01h	RS	RS	39h	MUX1_16_1	MUX1_16_1
02h	TMRCMP	-	3Bh	MUX1_32_17	MUX1_32_17
03h	DACDATA	-	3Dh	MUX1_48_33	MUX1_48_33
04h	DACADR	-	3Fh	MUX1_64_49	MUX1_64_49
05h	DACCTRL	DACST	49h	MUX2_16_1	MUX2_16_1
06h	ADCDATA	ADCDATA	4Bh	MUX2_32_17	MUX2_32_17
07h	ADCCTRL	ADCST	4Dh	MUX2_48_33	MUX2_48_33
08h	RDO_8_1	-	4Fh	MUX2_64_49	MUX2_64_49
09h	RDO_16_9	RDI_16_1	59h	MUX3_16_1	MUX3_16_1
0Ah	RDO_24_17	-	5Bh	MUX3_32_17	MUX3_32_17
0Bh	RDO_32_25	RDI_32_17	5Dh	MUX3_48_33	MUX3_48_33
0Ch	RDO_40_33	-	5Fh	MUX3_64_49	MUX3_64_49
0Dh	RDO_48_41	RDI_48_33	60h	-	RD_MATR
0Eh	RDO_56_49	-	70h	-	-
0Fh	RDO_64_57	RDI_64_49	75h	PROG_RESET	-
10h	ADC2DATA	ADC2DATA	78h	-	IOCFG1_16_1
11h	ADC2CTRL	ADC2ST	79h	-	IOCFG1_32_17
12h	RDIVT	RDIVT	7Ah	-	IOCFG1_48_33
14h	OUTDRIVE_REG	OUTDRIVE_REG	7Bh	-	IOCFG1_64_49
18h	iMASK_8_1	-	7Ch	-	IOCFG2_16_1
19h	iMASK_16_9	-	7Dh	-	IOCFG2_32_17
1Ah	iMASK_24_17	-	7Eh	-	IOCFG2_48_33
1Bh	iMASK_32_25	-	7Fh	-	IOCFG2_64_49
1Ch	iMASK_40_33	-	80h÷83h	-	M1_16_1÷M1_64_49
1Dh	iMASK_48_41	-	84h÷87h	-	M2_16_1÷M2_64_49
1Eh	iMASK_56_49	-	88h÷8Bh	-	M3_16_1÷M3_64_49
1Fh	iMASK_64_57	-	8Ch÷8Fh	-	M4_16_1÷M4_64_49
20h	-	DACCFG	90h÷93h	-	M5_16_1÷M5_64_49
21h	-	ADCCFG	94h÷98h	-	M6_16_1÷M6_64_49
29h	RIF_16_1	RiF16_1	98h÷9Bh	-	M7_16_1÷M7_64_49
2Bh	RIF_32_17	RiF32_17	9Ch÷9Fh	-	M8_16_1÷M8_64_49
2Dh	RIF_48_33	RiF48_33			
2Fh	RIF_64_49	RiF64_49			

ЗАО «Электронная компания «Элкус».
г. Санкт-Петербург, ул. Варшавская, д.5А, лит. Л.
тел./факс: (812) 369-0104
E-mail: mail@elcus.ru
arinc@elcus.ru

2012 г.