

**Применение процессоров серии «Мультикор»
Работа с памятью. Преобразование адресов и кэширование**

23.01.2015

1. ВВЕДЕНИЕ

При знакомстве с процессорами серии «Мультикор» нередко возникает ряд вопросов, связанных с особенностями адресации (физические и виртуальные адреса, диапазоны адресов для выводов nCS), а также про кэширование.

Поскольку CPU-ядро процессоров серии «Мультикор» является MIPS32-совместимым, данные вопросы подробно освещены в документации по архитектуре MIPS32, за исключением особенностей работы с портом внешней памяти и каналами DMA, так как они не являются частью процессорного ядра.

В данном документе приведена информация, служащая ответом на наиболее часто задаваемые вопросы по вышеуказанным темам.

2. ФИЗИЧЕСКИЕ И ВИРТУАЛЬНЫЕ АДРЕСА. КАРТА ПАМЯТИ ФИЗИЧЕСКИХ АДРЕСОВ

В архитектуре MIPS32 существуют понятия «физический адрес» и «виртуальный адрес». Адрес, указанный в инструкции доступа к памяти (LW/SW/...), в регистре PC, в инструкции перехода – это всегда виртуальный адрес. Он, будучи декодирован процессором из инструкции (или извлечен из регистра), преобразуется в физический адрес – для этого служит устройство управления памятью (Memory management unit – MMU). После преобразования происходит определение, к какой области памяти он относится, и тогда уже данный адрес передается либо на внутреннюю шину процессора для получения значения из внутренней памяти, либо в порт внешней памяти для получения значения из памяти, подключенной к процессору снаружи.

В картах памяти, приведенных в документации на процессоры серии «Мультикор», а также в документации на отладочные модули, указаны именно физические адреса.

В диапазоне физических адресов есть жестко закрепленные области. Например, область физических адресов 0x1800_0000-0x1801_FFFF всегда указывает на память SRAM в процессорах 1892BM12T. Физические адреса 0x182F_xxxx – это адреса периферийных регистров микросхемы, а область 0x1C00_0000-0x1FFF_FFFF – это всегда внешняя память, подключенная к выводу nCS[3]. Другие области, закрепленные за внешней памятью, по умолчанию не имеют соответствия другим выводам nCS. Соответствие конкретного диапазона физических адресов конкретному выводу nCS задается полями CSMASK и CSBA в соответствующем регистре CSCON. Так, например, на отладочном модуле NVCom-02TEM-3U (для процессора 1892BM10Я) к выводу nCS[0] подключена память объемом 64 Мбайт. По умолчанию в проектах, предназначенных для работы на данном модуле, значение регистра CSCON0 задается равным 0x3000FC. То есть, в поле CSMASK заносится значение 0xFC, а в поле CSBA – значение 0x00. В этом случае выводу nCS[0] соответствует диапазон физических адресов 0x0000_0000-0x3FFF_FFFF.

Важно также учитывать, что области памяти, задаваемые регистрами CSCON, не должны пересекаться. В противном случае, при обращении к адресу, соответствующему разным выводам nCS, в активное состояние перейдут все выводы nCS, соответствующие этому адресу. А значит, в случае чтения данных, все подключенные микросхемы памяти начнут выдавать на шину данных свои значения. Это приведет как минимум к некорректно прочитанному значению ячейки памяти.

Помимо порта внешней памяти, с физическими адресами работают каналы DMA. В регистрах IR необходимо задавать именно физический адрес.

3. ПРЕОБРАЗОВАНИЕ ВИРТУАЛЬНЫХ АДРЕСОВ В ФИЗИЧЕСКИЕ

Подробно о механизме преобразования адресов рассказано в руководстве пользователя на конкретный процессор, в разделе «Устройство управления памятью». Вкратце же стоит сказать, что для процессоров серии «Мультикор» существует два режима работы MMU. Первый – это режим фиксированного отображения (Fixed mapped, FM), в нем процессор работает сразу после снятия сигнала nRST. Второй – это режим TLB, который более необходим при реализации многозадачных операционных систем. Режим работы TLB подробно описан в соответствующей литературе и здесь затрагиваться не будет. Что же касается режима FM – он достаточно прост. Получая виртуальный адрес, устройство управления памятью анализирует его старшие три

разряда. В зависимости от их значения, определяется сегмент, к которому этот адрес относится, и физический адрес получается либо вычитанием строго заданного значения, либо отображается напрямую.

Например, к физическому адресу 0x1800_0000 можно обратиться следующими способами:

- виртуальный адрес 0x9800_0000 имеет значение старших трех разрядов, равное 100b. Это – сегмент kseg0. Физический адрес получается вычитанием значения 0x8000_0000;
- виртуальный адрес 0xB800_0000 имеет значение старших трех разрядов, равное 101b. Это – сегмент kseg1. Физический адрес получается вычитанием значения 0xA000_0000;
- виртуальный адрес 0x1800_0000 имеет значение старших трех разрядов, равное 000b. Это – сегмент kuseg. Физический адрес формируется в зависимости от состояния бита ERL регистра CP0.Status. Если ERL=1, физический адрес получается прибавлением значения 0x4000_0000. Если ERL=0, физический адрес равен виртуальному.

Несколько примеров, показывающих, в каких случаях происходит преобразование адреса в MMU, приведено ниже.

3.1 ВЫПОЛНЕНИЕ ИНСТРУКЦИИ SW/LW

Запись/чтение ячейки памяти выполняется соответственно с помощью инструкций SW/LW (для 32-разрядных слов), SH/LH (для 16-разрядных слов), SB/LB (для байтов). Также есть ряд инструкций, связанных с сопроцессором FPU, по которым обмен происходит между памятью и непосредственно регистрами сопроцессора, но механизм работы у всех этих инструкций одинаков. Далее будет рассмотрена инструкция SW, логика работы остальных инструкций полностью такая же.

SW reg1, offset(reg2)

Приведенная выше инструкция выполняет запись значения, содержащегося в регистре reg1, в адрес, сформированный из значения в регистре reg2 и 16-разрядного смещения, непосредственно указанного в коде инструкции. Значение в регистре reg2 и значение offset складываются. Сумма (reg2+offset) представляют собой виртуальный адрес, по которому будет производиться запись. Этот адрес преобразуется в MMU. На выходе MMU получается физический адрес, который выставляется на шину адреса порта внешней памяти или на внутреннюю шину адреса (в зависимости от того, в какой диапазон попадает адрес записи). В память записывается (то есть, выставляется на шину данных) значение, содержащееся в регистре reg1. Оно, разумеется, не подвергается никаким преобразованиям.

3.2 ЗАПИСЬ АДРЕСА В ИНДЕКСНЫЙ РЕГИСТР КАНАЛА DMA

Индексный регистр канала DMA (IR) – это регистр, содержащий в себе адрес данных, передаваемых с помощью DMA. Регистр IR является периферийным и отображается на адресное пространство процессора. Это значит, что доступ к нему осуществляется точно так же, как и к любой другой ячейке памяти – с помощью инструкций SW/LW.

Из этого следует, что записываемое в него значение не подвергается никаким преобразованиям. Поскольку DMA обращается к памяти минуя MMU, в регистр IR необходимо заносить сразу физический адрес. В программе это может выглядеть так:

```
unsigned int v_to_phy(unsigned int addr) {  
    return (addr-0x80000000); // предполагаем, что работаем только в kseg0  
}  
unsigned int Array[ARRAY_LEN];  
...  
void func() {  
    ...  
    IR = v_to_phy( (unsigned int) Array );  
    ...  
}
```

4. КЭШИРОВАНИЕ

Нередко возникает вопрос «как разместить программу в кэш-памяти?». Данная формулировка предполагает, что кэш-память – это некая область в адресном пространстве, в которую можно разместить код и/или данные. Данное предположение является неверным, а следовательно и формулировка – тоже некорректна.

Кэш-память не отображается на адресное пространство и содержит в себе значения отдельных ячеек памяти. В зависимости от выбранной реализации, кэширование может работать по-разному. В процессорах серии «Мультикор» используется кэш-память прямого отображения (direct mapped). Алгоритм работы данного типа кэш-памяти подробно описан в соответствующей литературе, поэтому в данном документе будут сформулированы только основные тезисы применения кэширования в процессорах серии «Мультикор».

Расположить программу в кэш-памяти нельзя. Можно расположить ее в кэшируемом сегменте виртуальных адресов и включить кэширование. Кэширование сегмента включается записью нужного значения в соответствующее поле регистра CP0.Config (или CP0.Config1, в зависимости от сегмента). Часть сегментов виртуальных адресов является некаэшируемой. Таким образом, в одном и том же диапазоне физических адресов могут быть расположены как кэшируемые данные/код, так и некаэшируемые.