

**Применение процессоров серии «Мультикор»
Использование отладчика MDB**

30.01.2015

1. ВВЕДЕНИЕ

В ряде случаев для отладки может быть предпочтительнее использование отладчика командной строки MDB, нежели графической IDE MCStudio. Типовые случаи, когда MDB удобнее – это отладка аппаратной части нового устройства, а также тестирование устройств на производстве.

В данном документе рассмотрено применение отладчика MDB в указанных выше целях. Кроме того, рассмотрена отладка ПО безотносительно отладки или тестирования аппаратной части изделия.

2. ОТЛАДКА АППАРАТНОЙ ЧАСТИ

При получении из производства первых экземпляров нового оборудования всегда существует вероятность ошибок в принципиальной электрической схеме, в топологии печатной платы, а также дефектов монтажа.

В этих условиях использование среды MCStudio для отладки менее удобно, так как для среды разработки старт отладки подразумевает в общем случае следующий последовательный и неразделимый комплекс действий:

- соединение с процессором по интерфейсу JTAG;
- подача сигнала nRST на процессор;
- перевод процессора в режим отладки (останов);
- выполнение команд предварительной настройки (окно «Startup registers»);
- загрузка программы в память процессора;
- изменение значения регистра PC на адрес точки входа в программу.

В случае ошибок в отлаживаемом устройстве, ошибка на одном из приведенных этапов приведет к тому, что MCStudio выведет довольно общее сообщение об ошибке, по которому затруднительно будет понять, на каком именно этапе появилась проблема. Кроме того, возможен более неприятный случай, когда ошибка, возникшая на первом этапе, компенсируется вторым этапом. В результате она может быть пропущена в ходе отладки, и будет выявлена только на стадии внедрения финальной версии ПО.

Отладчик MDB позволяет выполнить указанные этапы по отдельности, убедившись в корректном прохождении каждого из них. Таким образом снижается вероятность пропустить ошибку в устройстве.

2.1 УСТАНОВКА СОЕДИНЕНИЯ ПО JTAG

Для запуска отладчика необходимо вызвать его из командной строки с ключом «-и» - этот ключ указывает на то, что работа производится через эмулятор USB-JTAG. Отладчик соединяется с процессором по интерфейсу JTAG сразу после запуска. Если установить соединение удастся – отладчик выводит приглашение своей командной строки. Если нет - отладчик выводит сообщение об ошибке и завершает свою работу. Одним из частных случаев является ситуация, когда, например, не установлен драйвер эмулятора USB-JTAG, и отладчик не может его «найти». Данная ситуация, естественно, не имеет отношения к неисправностям платы. Решение подобных проблем описано в отдельном документе «Работа с эмулятором USB-JTAG для микропроцессоров серии «Мультикор», доступном на сайте <http://multicore.ru/>.

При установке соединения по JTAG отладчик запрашивает у процессора ID блока OnCD. В случае, если полученный ID не совпадает с необходимым, отладчик выдает сообщение вида «Fatal error: Unknown device idcode: ffffffff».

Появление такого сообщения свидетельствует о том, что есть проблема либо с линиями интерфейса JTAG между процессором и эмулятором USB-JTAG, либо из строя вышел один из выводов JTAG на процессоре, либо неисправен эмулятор USB-JTAG. В частности, при сгоревших входах JTAG на процессоре их сопротивление к «земле»

становится порядка сотен Ом, в результате чего уровень сигнала на выходе USB-JTAG снижается до 1-1.5 В, и процессор данного сигнала просто «не видит». Подобный выход из строя выводов JTAG процессора вызван обычно неправильной последовательностью подключения эмулятора к плате. Корректная последовательность подключения указана в этикетке на эмулятор. Этикетка доступна на сайте <http://multicore.ru/>.

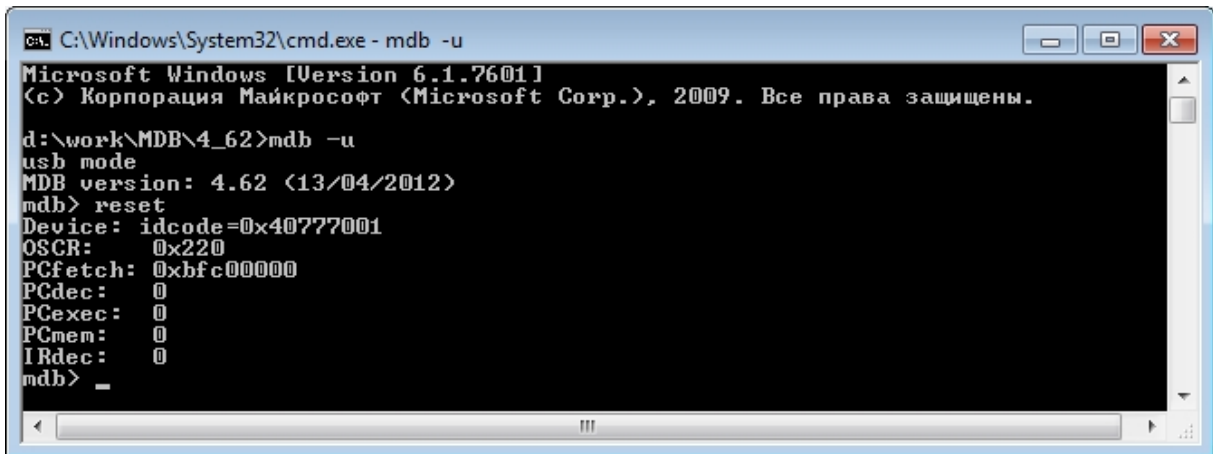
Также некорректно разведенные линии сигналов JTAG на плате могут приводить к искажению формы сигналов. В этом случае прочитанный ID будет иметь значение не 0xFFFF_FFFF, а некое произвольное.

В любом случае, каждая из вышеописанных ситуаций с некорректным ID требует изучения сигналов на линиях TDI, TDO, TMS, TCK с помощью осциллографа.

2.2 ПОДАЧА СИГНАЛА NRST И ПЕРЕВОД ПРОЦЕССОРА В РЕЖИМ ОТЛАДКИ

Для подачи сигнала nRST на процессор необходимо ввести в командной строке отладчика команду «reset». MDB, подавая сигнал nRST, также переводит процессор в режим отладки.

После этого на экран будет выведено следующее:



```
C:\Windows\System32\cmd.exe - mdb -u
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

d:\work\MDB\4_62>mdb -u
usb mode
MDB version: 4.62 (13/04/2012)
mdb> reset
Device: idcode=0x40777001
OSCR: 0x220
PCfetch: 0xbfc00000
PCdec: 0
PCexec: 0
PCmem: 0
IRdec: 0
mdb> _
```

Значения регистров PCfetch, PCdec, PCexec, PCmem должны соответствовать приведенным. Это значит, что на вход процессора пришел активный уровень сигнала nRST и процессор перешел в исходное состояние. В случае отсутствия сигнала nRST на входе процессора в регистрах PC будут некие произвольные значения.

Еще одной проблемой, проявляющейся на данном этапе, может стать отсутствие сигнала тактовой частоты на входе ХТІ процессора. В этом случае отладчик после команды «reset» ожидает ответа от процессора в течение длительного времени (десятки секунд), после чего на экран может быть выведена как вышеприведенная картина, так и сообщение об ошибке «debug request error».

2.3 ВЫПОЛНЕНИЕ КОМАНД ПРЕДВАРИТЕЛЬНОЙ НАСТРОЙКИ

Имеются в виду команды, предназначенные для настройки тактирования и порта внешней памяти микросхемы перед загрузкой программы в память. В среде разработки

MCStudio эти команды расположены в окне «Startup registers» и могут быть выполнены в отладчике MDB без изменений.

На данном этапе может проявиться ошибка, связанная с отсутствием тактирования, которая описана в предыдущем разделе. Проявляться она будет тоже в форме длительных таймаутов при выполнении каждой из команд.

2.4 ЗАГРУЗКА ПРОГРАММЫ В ПАМЯТЬ ПРОЦЕССОРА И ЗАПИСЬ АДРЕСА ТОЧКИ ВХОДА В РЕГИСТР РС

В случае использования среды разработки на данном этапе могут быть следующие проблемы:

- некорректно загруженные в память данные (код);
- поскольку происходит изменение регистра РС – может пройти незамеченным тот факт, что значение регистра РС не было корректным после подачи сигнала сброса. При использовании отладчика MDB данная проблема решается на стадии подачи команды «reset».

При загрузке программы в память процессора MCStudio производит также верификацию загруженных областей памяти на предмет соответствия секциям в файле с программой. Недостатком является тот факт, что в случае несоответствия среда разработки выдает лишь первый несовпавший байт или первое несовпавшее слово, из чего сложно сделать вывод о причинах некорректной работы памяти.

В отладчике MDB для загрузки программы удобнее всего использовать команду «loadelf», которая производит загрузку секций программы в память, читает из ELF-файла адрес точки входа и вносит этот адрес в регистр РС. По умолчанию данная команда не проводит верификации загруженных данных. Даже если включить эту функцию – после исполнения команды лишь будет выведено сообщение о соответствии или несоответствии загруженных данных файлу, но не будет указано, по каким адресам данные не соответствуют, и не будет приведена разница между записанными и считанными словами.

На случай подобной ошибки в отладчике MDB предусмотрена команда «testmem», в которой задается стартовый адрес проверяемой области памяти, ее длина и количество проходов. Заданная область памяти заполняется случайными значениями, после чего проверяется соответствие считанных и записанных данных. В случае наличия ошибок отладчик выводит в консоль количество ошибочно прочитанных ячеек памяти (в процентах от общего размера тестируемой области), а также небольшую выборку некорректных значений, где приведено записанное и считанное значение для некоторых из ячеек. По разнице между записанными и прочитанными значениями (а также по адресам ошибочных ячеек) можно сделать вывод о проблемах с памятью. В частности, если из всей тестируемой области памяти считывается одно и то же значение – можно предположить дефект монтажа вывода nCS или стробов чтения. В случае, если ошибка присутствует в одном и том же разряде данных – логично предположить, что дефект монтажа именно в этой линии данных. И так далее, исходя из общих соображений, в данном случае уже не имеющих отношения непосредственно к архитектуре микросхем серии «Мультикор».

3. ТЕСТИРОВАНИЕ

Для автоматизации тестирования MDB подходит прежде всего за счет возможности исполнения скриптов. Скажем, в качестве простейшего теста может быть использован такой скрипт:

```
reset
testmem 0xb8000000 0x8000
testmem 0xb8440000 0x4000
testmem 0xb8400000 0x20000
```

Он производит грубый тест внутренней памяти процессора, и по результатам выводит сообщение «Test failed» или «Test passed» для каждой области памяти. Можно анализировать результат каждого теста визуально, что тоже не очень удобно для производства, особенно в случае использования большого скрипта. Поэтому для отладчика MDB может быть написана несложная обертка, анализирующая его вывод и по совокупности результатов (например, отсутствие сообщений «Test failed») делающая вывод об успешном завершении теста в целом.

Неудобство заключается в том, что в синтаксисе команд MDB не предусмотрена реализация циклов и условий, поэтому написать полноценный развернутый тест в скрипте не представляется возможным. Поэтому наиболее рациональным выглядит следующий механизм тестирования: для процессора пишется программа, осуществляющая тестирование требуемого блока. По результатам тестирования в заданную ячейку памяти или глобальную переменную записывается флаг, свидетельствующий об успешном или неуспешном прохождении теста. После этого в программе ставится инструкция останова.

Данная программа загружается и запускается в отладчике MDB. После выполнения команды «run» отладчик не выводит приглашение командной строки, а постоянно отслеживает, не перешел ли процессор в состояние останова. Только в случае останова приглашение командной строки появляется снова. Останов может произойти в случае, если процессор выполнит инструкцию BREAK (так называемая «программная точка останова»), если адрес PC будет равен адресу, заданному командой отладчика «bp» («аппаратная точка останова»), или в случае нажатия пользователем сочетания Ctrl+C – в этом случае отладчик сам остановит процессор на текущей инструкции. Таким образом, команды отладчика, расположенные в скрипте после команды «run», будут выполнены не ранее, чем тестирующая программа дойдет до точки останова. Отладчик MDB, помимо прочего, разбирает отладочную информацию в составе ELF-файла, поэтому с помощью команд show или dump можно получить состояние глобальной переменной или массива.

С учетом вышесказанного, в общем виде программа теста может выглядеть так:

```
unsigned int result;
...
int main() {
    ...
    result = Test();
    asm volatile (".word 0x4d"); // asm volatile ("break 0x1");
}
```

Скрипт для MDB, запускающий тест, будет выглядеть так:

```
reset
... #здесь могут быть настройки порта внешней памяти и PLL
loadelf test.elf
run
show result
```

Чтобы MDB сразу после запуска приступил к исполнению скрипта, необходимо запускать его с ключом «-f filename».

Возможность доступа к глобальным переменным может быть также использована для того, чтобы задавать какие-то параметры теста непосредственно в скрипте. В этом случае параметры объявляются в коде также глобальными переменными:

```
unsigned int param;
unsigned int result;
...
int main() {
    ...
    result = Test(param);
    asm volatile (".word 0x4d"); // asm volatile ("break 0x1");
}
```

В скрипте же надо учесть, что, как правило, в программах на языке Си перед входом в функцию main() происходит обнуление неинициализированных переменных, поэтому задавать параметр необходимо только после входа в main():

```
reset
... #здесь могут быть настройки порта внешней памяти и PLL
loadelf test.elf
bp 0 main # аппаратная точка останова на входе в main()
run
set param 0x123
run
show result
```

4. ОТЛАДКА ПРИЛОЖЕНИЙ

Отладчик MDB может быть также использован для отладки непосредственно ПО, без привязки к выявлению недочетов в аппаратной части устройства.

В предыдущем разделе приведен пример скрипта отладчика, реализующего загрузку теста. Скрипт, используемый для загрузки отлаживаемого приложения, будет точно таким же. За исключением, может быть, предварительно установленных точек останова в других адресах или функциях.

Неудобством при использовании отладчика MDB является отсутствие в нем возможности вывести дизассемблер исполняемого участка памяти. Однако данное неудобство легко обходится, если иметь под рукой файл с дизассемблированным кодом программы. Такой файл можно получить с помощью программы OBJDUMP:

```
mipsel-elf32-objdump -D -S filename.elf > filename.dis
```

Ключ «-D» говорит программе о необходимости дизассемблировать все секции файла, а ключ «-S» - о том, чтобы в дизассемблированном файле были приведены соответствующие строки исходного кода.