

Rabbit 4000 Advanced 16-bit Mode Auxiliary I/O Bug

A bug has recently been identified in the Rabbit 4000 that can result in inadvertently executing a data byte as if it were an instruction under certain circumstances. This will only occur when the advanced 16-bit mode is enabled, and only when certain peripheral modes are being used.

Description

A multiplexer is used to select which data are presented to the processor when any of the following peripheral features are used.

- The auxiliary I/O bus is enabled on Parallel Ports A and B.
- The external I/O bus has the short read/write strobes option selected.
- Self-timed chip selects are enabled.

When the advanced 16-bit mode is enabled, it is possible for the prefetch queue to sample the data under certain circumstances from this multiplexer before it switches back to the instruction code, resulting in that data being interpreted as an instruction. The resulting behavior depends on the value of that data byte.

This bug does not occur in either the 8-bit mode or the basic 16-bit mode.

Workarounds

To avoid this bug, disable the advanced 16-bit mode before using one of the above peripherals, and re-enable the advanced 16-bit mode immediately afterwards.

Starting with Dynamic C v. 10.21, the default is to operate all memory devices in the basic 16-bit mode most of the time on boards such as the RCM4000 series that have 16-bit memories. This may impact the memory performance, depending on the required wait state settings.

As long as your application does not use any of the three peripheral features that are associated with the defect, you may add the following macro to your program to improve the memory performance.

```
#define __ALLOW_16BIT_AUXIO_DEFECT
```

When you use this macro, comment out the **#error** and **#fatal** messages associated with it in the Dynamic C `DCRab...\BIOSLIB\STDBIOS.C` program before you run your application.

NOTE: Remember that this macro will expose the defect, and should only be used when you absolutely need the improved memory performance and your application does not use any of the three peripheral features that are associated with the defect.

When you are using the advanced 16-bit mode, you can still disable the advanced 16-bit mode before using one of the three peripheral features that are associated with the defect, and re-enable the advanced 16-bit mode immediately afterwards. The following code sequence illustrates this for the external I/O bus with the short read/write strobes option selected.

```
// In this example, a 16-bit flash is on /CS0
// and a 16-bit SRAM is on /CS1

LD A, 0x24          // put /CS0 and /CS1 into the basic 16-bit mode
IOI LD (MACR), A
NOP
NOP                // ensure that the prefetch queue is flushed
NOP

LD A, (extIOdata)
IOE LD (extIOaddr), A // external I/O bus access

LD A, 0x32          // re-enable advanced 16-bit mode on /CS0 and /CS1
IOI LD (MACR), A
NOP                // allow time for advanced mode startup
NOP
```

Rabbit Semiconductor Inc.

www.rabbit.com