Dakar Errata

Release Date: 16 March 1999
Revision 1.01

Title: F5 Dakar blink example will only work on site A				
Database Id#	**H/W Component**	**S/W Component**	**Version**	**Fixed?**
#385		WinNT SDK	v2.00	will fix in next release

Errata Description

Problem:
The LED will not blink, if the blink.out file is loaded onto a TIM module on site B, C or D.

Solution:
Only running the blink.out file form site A. The LED flashes with blink.out loaded on site A, because only site A has access to the PLX PCI9060 registers. Refer to the F5 Dakar Technical Reference Manual, doc #500-000353, section 5, PCI Bus Interface and view Table 5, Embedded C44 Memory Map.

Title: SDF and LDF files provided with the SDK examples is different from the ones created by Toolbox

Database Id#	H/W Component	S/W Component	Version	Fixed?
#290		WinNT SDK	v2.00	will fix in next release

Errata Description

Problem:
The SDF and LDF provided with the Dakar SDK reference the nodes as SITE. For example, the embedded processor is referenced as Board1:SiteA:Proc0

Toolbox references the nodes as NODE. For example, the embedded processor is Board1:NodeA:Proc0

When using toolbox to generate a new SDF, this file won't work with the examples, unless using the LDF generated by Toolbox.

Solution:
Whenever a SDF or LDF needs to be generated by Toolbox, generate both files. Ensure both LDF and SDF reference the nodes the same way.

Title: The examples fail if the SDF has incorrect settings.

Database Id#	H/W Component	S/W Component	Version	Fixed?
#292		WinNT SDK	v2.00	will fix in next release

Errata Description

Problem:
When running the example with a wrong SDF, the example will not work and DSP code will not be loaded.

Solution:
A SDF is used to define the system's hardware configuration and to compute the loading path for all processors in the system.
For a detailed description and example of a SDF, see F5 Carrier Board User Guide, Section 6.4 Defining your system's Hardware Configuration, Section 10.2 How to Run the Example Program and Appendix C: System Definition file: Description and Examples.

Title: C4x_Control function always returns ILLEGAL OPERATION when taking PCI as the value for resource parameter.

Database Id#	H/W Component	S/W Component	Version	Fixed?
#294		DSP SDK	v2.00	will fix in next release

Errata Description

Problem:
F5 Dakar DSP SDK library missing an else statement and cause the C4x_Control function always returns ILLEGAL OPERATION when taking PCI as the value for resource parameter

Solution:
Modify control.c (located in F5SDK\Dev\DSP\SRC\F5alib) by adding the missing else back as following:

ORIGINAL

```
/* Read PCI9060 mailbox
 */
if (GET_MAILBOX == Operation)
{
   if (8 > Flags)
      *(UINT32 *)Value = *(F5_PLXMAILBOX0_REG + Flags);
   else
      result = ILLEGAL_FLAGS;
}
```

CORRECTION

```
/* Read PCI9060 mailbox
 */
else if (GET_MAILBOX == Operation)
{
   if (8 > Flags)
      *(UINT32 *)Value = *(F5_PLXMAILBOX0_REG + Flags);
   else
      result = ILLEGAL_FLAGS;
}
```

Rebuild f5_c4x.lib with Makefile.c4x (provided in f5sdk\dev\dsp\src\f5alib)

Replace the original f5_c4x.lib (located in f5sdk\dev\dsp\lib & f5sdk\lib) with the new one.

Title: The PLX Interrupt control/Status register is not set by the C4x_Control function				
Database Id#	**H/W Component**	**S/W Component**	**Version**	**Fixed?**
#295		WinNT SDK	v2.00	will fix in next release

Errata Description

Problem:
The bit in the PLX Interrupt control/Status register which enables local DMA0 interrupts is not set in the C4x_Control (INTERRUPTS, …) function.

Solution:
Adding the following lines to the application code allows the PLX DMA interrupt to work.

```
#define LOCAL_DMA0_INT_ENABLE  0x00040000

/* enable local DMA0 interrupts in Int control reg */
*F5_PLXINTCONTROL_REHG |= LOCAL_DMA0_INT_ENABLE;
```