SPI-DirectC v2.1 & DirectC v4.1 Release Notes

Contents

What's New in this Release

DirectC V4.1 has enhanced RTG4™ programming algorithm to perform additional checks after programming every frame. In addition, M2S/M2GL "Device Info" action has been modified to display fabric and eNVM bitstream digests on both DirectC and SPI-DirectC solutions.

This version of DirectC operates on DAT files generated by Designer v8.6 or later for programming IGLOO, IGLOO nano, IGLOO PLUS, ProASIC3, ProASIC3L (excluding M1 variants), SmartFusion, and Fusion. Libero version 11.4 or later should be used to generate DAT files for IGLOO2 and SmartFusion2 devices.

Full plain text and encrypted programming support is provided for array, FlashROM, eNVM and security components. The DirectC software supports 8-, 16-, and 32-bit microprocessors.

JTAG-DirectC SARs Fixed:

SAR 93419: RTG4 - Enhance programming algorithm to check the status of every frame after programming.

SAR 91528: M2S/M2GL – Enhance programming algorithm to address a false failure when entering programming mode.

SAR 93836: M2S/M2GL – Enhance Device Info action to export bitstream fabric and eNVM digest values.

SPI-DirectC SARs Fixed:

SAR 91528: M2S/M2GL – Enhance programming algorithm to address a false failure when entering programming mode.

SAR 93836: M2S/M2GL – Enhance Device Info action to export bitstream fabric and eNVM digest values.

Supported Families

DirectC v4.1 is available to support IGLOO® nano—the world's lowest power FPGA, IGLOO, IGLOO PLUS, IGLOO2, ProASIC®3, ProASIC nano, ProASIC3L (excluding M1 variants), PolarFire™, RTG4™, SmartFusion2, SmartFusion, and Fusion.

SPI-DirectC v2.1 is available to support the PolarFire™, IGLOO2 and SmartFusion2 families of devices.

Product Family	Device
IGLOO	AGL015, AGL030, AGL060, AGL125, AGL250, AGL600, AGL1000, AGLE600, M1AGL250, M1AGL600
IGLOO nano	AGLN010, AGLN015, AGLN020, AGLN060, AGLN125, AGLN250, AGLN030Z, AGLN060Z, AGLN125Z, AGLN250Z
IGLOO PLUS	AGLP030, AGLP060, AGLP125
IGLOO2	M2GL005, M2GL010, M2GL025, M2GL050, M2GL090, M2GL1050
ProASIC3	A3P015, A3P030, A3P060, A3P125, A3P250, A3P400, A3P600, A3P1000, A3PE1500, A3PE3000, M7A3P400, M7A3P1000, M1A3P250, M1A3P600, M1A3P1000, M1A3PE1500
ProASIC3 nano	A3PN010, A3PN015, A3PN020, A3PN060, A3PN125, A3PN250, A3PN030Z, A3PN060Z, A3PN125Z, A3PN250Z
ProASIC3L	A3P250L, A3P600L, A3P1000L
SmartFusion2	M2S005, M2S010, M2S025, M2S050, M2S090, M2S150
PolarFire	MPF300
RTG4	RT4G150
SmartFusion	A2F200, A2F500
Fusion	AFS090, AFS250, AFS600, AFS1500, M7AFS600, M1AFS250,M1AFS600, M1AFS1500, P1AFS600, P1AFS1500

DirectC v1.3 must be used for programming ProASICPLUS (APA) devices.

Security programming is enabled. Take care when using this feature. DirectC is designed for remote reprogramming via a microprocessor. Security programming should only take place in a trusted environment. In a non-secure environment, the communications line between the PC and the remote equipment must be secured by the end customer. After altering the security settings, remote upgrades using DirectC with an encrypted STAPL file (matching the AES key programmed during altering the security settings) can be safely carried out over a non-secure communications line by the user.

Known Issues and Workarounds

SAR 18887: When using the compiler option, ENABLE_CODE_SPACE_OPTIMIZATION, programming the CORE or FROM of a previously secured device, using a DAT file with a different security key will erroneously pass. The device remains in the same state prior to the operation. The CORE or FROM programming is not executed on the device. This SAR does not apply to the SmartFusion2 or IGLOO2 families of devices.

SAR109901: To avoid a potential IO glitch when exiting programming mode, do the following::

SmartFusion2/IGLOO2:

Add the following line in dpG4alg.h:

```
#define G4M_EXTEST2      0x09u
```

Replace the following function in dpG4alg.c:

```
void
dp_G4M_exit(void) {
      ……..
}
```

With this one:

```
void dp_G4M_exit(void)
{
    if (pgmmode_flag == TRUE)
    {
        opcode = ISC_DISABLE;
        IRSCAN_in();

goto_jtag_state(JTAG_RUN_TEST_IDLE,G4M_STANDARD_CYCLES);
        dp_delay(G4M_STANDARD_DELAY);

        opcode = ISC_DISABLE;
        dp_G4M_device_poll(32u, 31u);
        #ifdef ENABLE_DISPLAY
        if ((error_code != DPE_SUCCESS) && (unique_exit_code
== DPE_SUCCESS))
        {
            dp_display_text("\r\nFailed to disable
programming mode.");
        }
        #endif
    }

    #ifdef ENABLE_DISPLAY
    dp_G4M_read_fsn();
    #endif

    opcode = G4M_EXTEST2;
    IRSCAN_in();
    goto_jtag_state(JTAG_RUN_TEST_IDLE,G4M_STANDARD_CYCLES);
    dp_delay(G4M_EXTEST2_DELAY);

    goto_jtag_state(JTAG_TEST_LOGIC_RESET,0u);
    return;
}
```

RTG4:

Add the following line in dpRTG4alg.h:

```
#define RTG4M_EXTEST2     0x09u
```

Replace the following function in dpRTG4alg.c:

```
void dp_RTG4M_exit(void)
{
      ….
}
```

With this one:

```
void dp_RTG4M_exit(void)
{
    if (rtg4_pgmmode_flag == TRUE)
    {
        #ifdef ENABLE_DISPLAY
        dp_RTG4M_read_dsn();
        #endif

        opcode = ISC_DISABLE;
        IRSCAN_in();

goto_jtag_state(JTAG_RUN_TEST_IDLE,RTG4M_STANDARD_CYCLES);
        dp_delay(RTG4M_STANDARD_DELAY);

        opcode = ISC_DISABLE;
        dp_RTG4M_device_poll(32u, 31u);
        #ifdef ENABLE_DISPLAY
        if ((error_code == DPE_POLL_ERROR) &&
(unique_exit_code ==
DPE_SUCCESS))
        {
            dp_display_text("\r\nFailed to disable
programming mode.");
        }
        #endif
    }

    opcode = RTG4M_EXTEST2;
    IRSCAN_in();

goto_jtag_state(JTAG_RUN_TEST_IDLE,RTG4M_STANDARD_CYCLES);
    dp_delay(RTG4M_EXTEST2_DELAY);


    opcode = RTG4M_JTAG_RELEASE;
    IRSCAN_in();
    dp_delay(RTG4M_RESET_DELAY);

goto_jtag_state(JTAG_TEST_LOGIC_RESET,RTG4M_RESET_CYCLES);
    return;
}
```

PolarFire:

Add the following line in dpG5alg.c:

```
#define G5M_EXTEST2     0x09u
```

Replace the following function in dpG5alg.c:

```
void dp_G5M_exit(void)
{
      …
}
```

With this one:

```
void dp_G5M_exit(void)
{
    if (g5_pgmmode_flag == TRUE)
    {
        opcode = G5M_ISC_DISABLE;
        IRSCAN_in();

goto_jtag_state(JTAG_RUN_TEST_IDLE,G5M_STANDARD_CYCLES);
        dp_delay(G5M_STANDARD_DELAY);

        opcode = G5M_ISC_DISABLE;
        dp_G5M_device_poll(32u, 31u);
        #ifdef ENABLE_DISPLAY
        if ((error_code != DPE_SUCCESS) && (unique_exit_code
== DPE_SUCCESS))
        {
            dp_display_text("\r\nFailed to disable
programming mode.");
        }
        #endif
    }
    #ifdef ENABLE_DISPLAY
    dp_G5M_read_fsn();
    #endif

    opcode = G5M_EXTEST2;
    IRSCAN_in();
    goto_jtag_state(JTAG_RUN_TEST_IDLE,G5M_STANDARD_CYCLES);
    dp_delay(G5M_EXTEST2_DELAY);

    goto_jtag_state(JTAG_TEST_LOGIC_RESET,5u);
    return;
}
```

System Requirements

Any development system that supports ANSI C Programming.

Microprocessor compiler for the chosen platform.

DAT file generated by Microsemi Designer Software or Microsemi Libero® Integrated Design Environment (IDE).

Download DirectC Installer v1.1

DirectC Installer v1.1 (187 MB)

Note: DirectC installer v1.1 contains the source, sample projects, and Libero design files for both DirectC V4.1 and SPI-DirectC v2.1 solutions. In addition, Windows based UARTHostLoader application used to interact with the IAR sample projects is included in this installer. Please see DirectC and SPI-DirectC user guides for more information.

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: **sales.support@microsemi.com**
www.microsemi.com

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at **www.microsemi.com**.