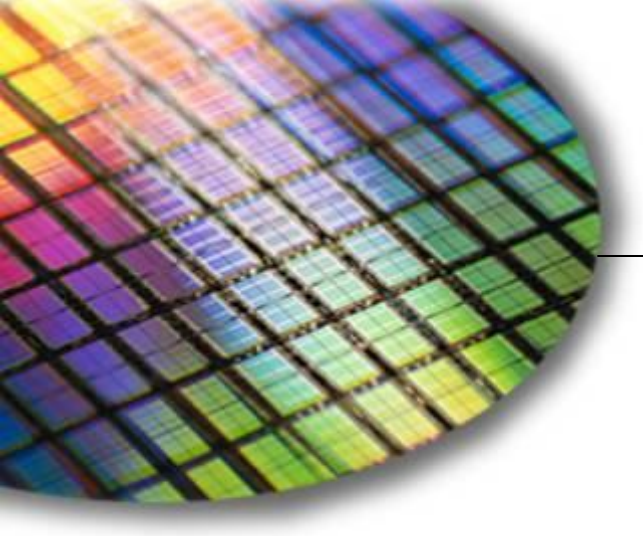

The World Leader in High Performance Signal Processing Solutions

System Services and Device Drivers: Migrating from VisualDSP++® to CrossCore® Embedded Studio

Craig Gilchrist

Embedded Systems Products &Technologies Group

Analog Devices Inc.

ANALOG DEVICES

Training Module Outline

Migrating from VisualDSP++® and SSL1.0 to CrossCore® Embedded Studio and SSL2.0

- **Feature Comparison Table**
- **Significant Changes**
- **Simple Migration Example**

Demo: Porting VisualDSP++® Project to CrossCore® Embedded Studio

- **Remove Redundant Services**
  - Unnecessary Includes
  - Memory Allocations
  - API Calls
- **Add Memory Definitions for Device Drivers**
- **Replace Device Manager API (adi_dev_xxxx) with Device-Specific API**
- **Modify or Remove Callback Functions**

Migrating from VisualDSP++® to CrossCore® Embedded Studio: Feature Comparison Table

Feature	SSL 1.0 VisualDSP++®	SSL 2.0 CrossCore® Embedded Studio
Device Management	Central Device Management Service and API	No Central Device Manager, Per-Device API
Interrupt Management	Central Interrupt Manager Service	Device Drivers Manage Own Interrupts
Callbacks	Yes	Yes (discouraged)
Deferred Callbacks	Central Deferred Callback Manager	No Deferred Callbacks
DMA Management	Central DMA Management Service	Drivers Manage Own DMA
Power Service	Yes	Yes (simplified)
EBIU Service	Yes	No
General Purpose I/O Management	Flag Service	GPIO Service
MISRA Compliant	No	Yes - MISRA-C:2004
Memory Requirement calculation	Yes (Specified per priority level)	Yes (Specified per task)

Migrating from VisualDSP++® to CrossCore® Embedded Studio: Summary of Significant Changes

Services Removed

- Device Manager
- Deferred Callback Manager
- DMA Manager
- Interrupt Manager and Secondary Interrupts
- EBIU Service

Significant Changes

- No longer requires users to calculate the memory requirements
- Callbacks no longer default, nor recommended
- Power Service simplified to allow Clock Configuration only
- MISRA compliant Device Drivers
- Device-Specific control API due to Device Manager removal
- Flag Service is now the GPIO service

Simple Migration Example - UART VisualDSP++ Code

```
#include <services/services.h>
#include <drivers/adi_dev.h>
#include <drivers/uart/adi_uart.h>

ADI_DMA_MANAGER_HANDLE adi_dma_ManagerHandle;
ADI_DEV_MANAGER_HANDLE adi_dev_ManagerHandle;
static ADI_DCB_HANDLE          hDCBManager;
static ADI_DEV_DEVICE_HANDLE    hUARTDriver;

#define ADI_SSL_INT_NUM_SECONDARY_HANDLERS  (4)
#define ADI_SSL_DCB_NUM_SERVERS             (4)
#define ADI_SSL_DMA_NUM_CHANNELS            (2)
#define ADI_SSL_FLAG_NUM_CALLBACKS          (0)
#define ADI_SSL_SEM_NUM_SEMAPHORES          (0)
#define ADI_SSL_DEV_NUM_DEVICES             (3)

static  u8  InterruptServiceData        [ADI_INT_SECONDARY_MEMORY *
ADI_SSL_INT_NUM_SECONDARY_HANDLERS];
static  u8  DeferredCallbackServiceData [ADI_DCB_QUEUE_SIZE *
ADI_SSL_DCB_NUM_SERVERS];
static  u8  DMAServiceData              [ADI_DMA_BASE_MEMORY +
(ADI_DMA_CHANNEL_MEMORY *  ADI_SSL_DMA_NUM_CHANNELS)];
static  u8  FlagServiceData             [ADI_FLAG_CALLBACK_MEMORY *
ADI_SSL_FLAG_NUM_CALLBACKS];
static  u8  SemaphoreServiceData        [ADI_SEM_SEMAPHORE_MEMORY *
ADI_SSL_SEM_NUM_SEMAPHORES];
static  u8  DevMgrData                  [ADI_DEV_BASE_MEMORY +
(ADI_DEV_DEVICE_MEMORY * ADI_SSL_DEV_NUM_DEVICES)];
static         u8             nDCBMgrData
                                        [ADI_DCB_QUEUE_SIZE +
(ADI_DCB_ENTRY_SIZE)*4];

adi_int_Init(InterruptServiceData, sizeof(InterruptServiceData), &i,
ADI_SSL_ENTER_CRITICAL);

adi_pwr_Init(ezkit_power);

adi_dcb_Init(DeferredCallbackServiceData,
sizeof(DeferredCallbackServiceData), &i, ADI_SSL_ENTER_CRITICAL);
```

```
adi_dma_Init(DMAServiceData, sizeof(DMAServiceData), &i,
&adi_dma_ManagerHandle, ADI_SSL_ENTER_CRITICAL);

adi_flag_Init(FlagServiceData, sizeof(FlagServiceData), &i,
ADI_SSL_ENTER_CRITICAL);

adi_dev_Init(DevMgrData, sizeof(DevMgrData), &i,
&adi_dev_ManagerHandle, ADI_SSL_ENTER_CRITICAL);

adi_dcb_Open(14U, &nDCBMgrData[ADI_DCB_QUEUE_SIZE],
(ADI_DCB_ENTRY_SIZE)*4, &nResponseCount, &hDCBManager);

adi_dev_Open(adi_dev_ManagerHandle, &ADIUARTEntryPoint,
UART_DEVICE_NUMBER, NULL, &hUARTDriver,
ADI_DEV_DIRECTION_BIDIRECTIONAL, adi_dma_ManagerHandle, hDCBManager,
UARTCallback);

ADI_DEV_CMD_VALUE_PAIR oUARTConfigTable [] = {
    { ADI_DEV_CMD_SET_DATAFLOW_METHOD, (void*)ADI_DEV_MODE_CHAINED },
    { ADI_UART_CMD_SET_DATA_BITS, (void*)8U                        },
    { ADI_UART_CMD_ENABLE_PARITY, (void*)FALSE                     },
    { ADI_UART_CMD_SET_STOP_BITS, (void*)1                         },
    { ADI_UART_CMD_SET_BAUD_RATE, (void*)57600                     },
    { ADI_DEV_CMD_END, NULL                                       },
};

adi_dev_Control(hUARTDriver, ADI_DEV_CMD_TABLE,
(void*)oUARTConfigTable);

adi_dev_Control(hUARTDriver, ADI_DEV_CMD_SET_DATAFLOW, (void*)TRUE)

adi_dev_Read(hUARTDriver, ADI_DEV_1D, (ADI_DEV_BUFFER *)pBuffer);
```

Simple Migration Example - UART CrossCore Embedded Studio Code

```
#include <drivers/uart/adi_uart.h> /* UART Driver Include */

/*UART Device Handle, Memory and Buffer */
static ADI_UART_HANDLE UARTDriverHandle;
static uint8_t gUARTMemory[ADI_UART_UNIDIR_DMA_MEMORY_SIZE];
static uint8_t RxTxBuffer[BUFF_SIZE];

/* Initialize the Power Service */
adi_pwr_Init(25000000, 500000000, 125000000, 60000000);

/* Open the UART Device */
adi_uart_Open( UART_DEVICE_NUM,
               ADI_UART_DIR_TRANSMIT,
               gUARTMemory,
               ADI_UART_UNIDIR_DMA_MEMORY_SIZE,
               &UARTDriverHandle);

/* Set the Baud Rate */
adi_uart_SetBaudRate(UARTDriverHandle, BAUD_RATE);

/* Configure the UART Device */
adi_uart_SetConfiguration( UARTDriverHandle,
                           ADI_UART_NO_PARITY,
                           ADI_UART_ONE_STOPBIT,
                           ADI_UART_WORDLEN_8BITS);

/* Enable Transmit */
adi_uart_EnableRx(UARTDriverHandle, true);

/* Write to the buffer */
adi_uart_Write(UARTDriverHandle, &RxTxBuffer[0], 1);
```

Demo
Porting VisualDSP++® Project to CrossCore® Embedded Studio

For More Information

CrossCore® Embedded Studio: www.analog.com/cces

Board Support Packages: www.analog.com/swexamples

Processors and DSP Website: www.analog.com/processors

Blackfin Documentation: www.analog.com/blackfin/manuals

Analog Devices Video Channel: videos.analog.com

EngineerZone Support Community: http://ez.analog.com/

ANALOG DEVICES