

Keystone Design Consideration

Agenda

- **Multi-core Design Overview**
- Inter-core communications
 - Shared Memory
 - Inter-core Interrupt
 - Semaphore
 - EDMA3
 - Navigator
- Multi-core Software Develop Kit (MCSDK) Introduction

Why Multi-core DSPs?

- Increase of data rate
 - Think about Ethernet, from 10Mbps to 10Gbps
- Increase in algorithm complexity
 - Think about typical face recognition, finger prints
- Increase in development cost
 - Hardware and software development
- Multicore SOC devices are a solution
 - Fast peripherals part of the device
 - High performances, fixed point and floating point processing power. Parallel data movement.
 - Off the shelf devices
 - Elaborate set of software tools

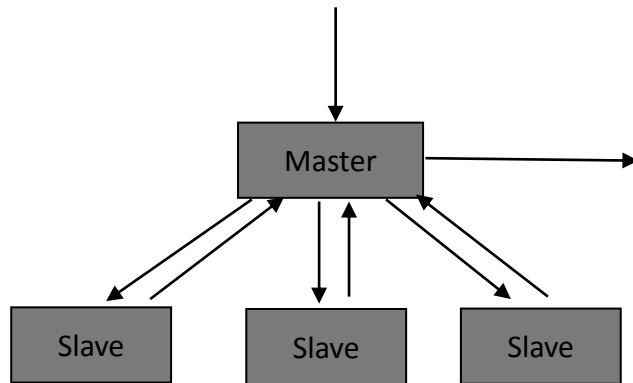
Common Usage Cases

- Network gateway, speech/voice processing
 - Typically hundreds or thousands of channels
 - Each channel consumes about 30 MIPS
- Large, complex, floating point FFT
- Video processing
- Medical imaging
- LTE, WiMAX, other wireless physical layers
- Scientific processing (Oil explorations)
 - Large complex matrix manipulations
- Your applications?

Parallel Processing Models

Master Slave Model

- Centralized control and distributed execution
- Master responsible for execution scheduling and data availability
- Required fast and cheap (in terms of CPU resources) messages and data exchange between cores
- Application consists of many small independent threads

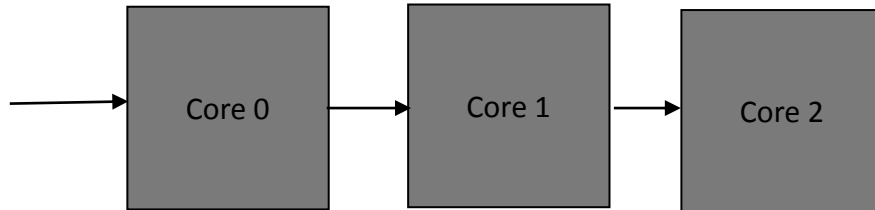


- Typical Applications
 - Multiple media processing
 - Video encoder slice processing
 - JPEG 2000 – multiple frames
 - VLFFT

Parallel Processing Models

Data Flow Model

- Distributed Control and execution
- The algorithm is partitioned into multiple block, each block is processed by a core, and the output of one core is the input to the next core
 - Exchange data and messages between any cores
- Big challenge – partition blocks to optimize performances
 - Requir loose link between cores (queue system)



- Typical Applications
 - High quality video encoder
 - Video decoder
 - Video transcoder
 - LTE physical layer

Agenda

- Multi-core Design Overview
- **Inter-core communications**
 - Shared Memory
 - Inter-core Interrupt
 - Semaphore
 - EDMA3
 - Navigator
- Multi-core Software Develop Kit (MCSDK) Introduction

Core Local Memory

- For each core, L1/L2 memories have two entries in the memory map.
 - **Global addresses:** accessible to all masters in the chip
 - **Local (aliased) addresses:** accessible only to the local core and IDMA
 - The eight most significant bits are masked to zero
 - E.g. 0x10800000 and 0x00800000 are the same memory for core 0.
 - Allows for common code to be run unmodified on multiple cores
 - Not beneficial for un-shared code.
- Each core has a private configuration space
 - Local core control registers (cache, TSC, IDMA, INTC) are not visible to other masters in the chip.
- Core number
 - software can verify the core on which it is running through register (DNUM) that holds the DSP core number (0, 1, or 2...)
 - The core number can be used during run-time to conditionally execute code, update pointers, create a global address, etc.

Multi-core Shared Memory Map

- Multi-core Shared Memory (MSM)
 - Supports following cache options
 - Shared L2 SRAM mode (L1 will cache, L2 will not cache requests to MSM SRAM)
 - Level 3 SRAM mode (L1 and L2 will both cache the MSM SRAM)
 - Dynamically shared among all cores
 - Data and program
- DDR3 SDRAM
 - 8 Gb of addressable memory for data and program

Memory allocations

- Some data that will be used out of your control can not be shared between cores even using the same image for all cores.
For example
 - vector table
 - stacks
- Other data or code can be shared but application need to ensure the mutex access.(IPC, Hardware Semaphore).
- If multiple masters access the same address, BE careful of cache coherency.
 - Only cache coherency between L1D and LL2 is maintained by hardware.
 - Set memory non-cachable.
 - Cache Invalidate and Writeback operation.

Inter-core Interrupts

- 2 Registers per core to control Inter-DSP Interrupts
 - IPCG (In IPCGRx)
 - Write '1' to IPCG triggers an interrupt to corresponding GEM
 - Write '0' and Reads have no effect
 - SRCSx (In IPCGRx)
 - SW method to tell what cause the interrupt
 - Write of '1' is sticky and is read back as '1' until cleared.
 - Write of '0' has no effect
 - Reads return the current value of the bit
 - SRCCx (In IPCARx)
 - Write of '1' clears SRCSx in IPCARx
 - Write of '0' or read has no effect

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCS27	SRCS26	SRCS25	SRCS24	SRCS23	SRCS22	SRCS21	SRCS20	SRCS19	SRCS18	SRCS17	SRCS16	SRCS15	SRCS14	SRCS13	SRCS12
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	1		0
SRCS11	SRCS10	SRCS9	SRCS8	SRCS7	SRCS6	SRCS5	SRCS4	SRCS3	SRCS2	SRCS1	SRCS0	Reserved			IPCG
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-000			R/W-0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCC27	SRCC26	SRCC25	SRCC24	SRCC23	SRCC22	SRCC21	SRCC20	SRCC19	SRCC18	SRCC17	SRCC16	SRCC15	SRCC14	SRCC13	SRCC12
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	0		
SRCC11	SRCC10	SRCC9	SRCC8	SRCC7	SRCC6	SRCC5	SRCC4	SRCC3	SRCC2	SRCC1	SRCC0	Reserved			
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0000			

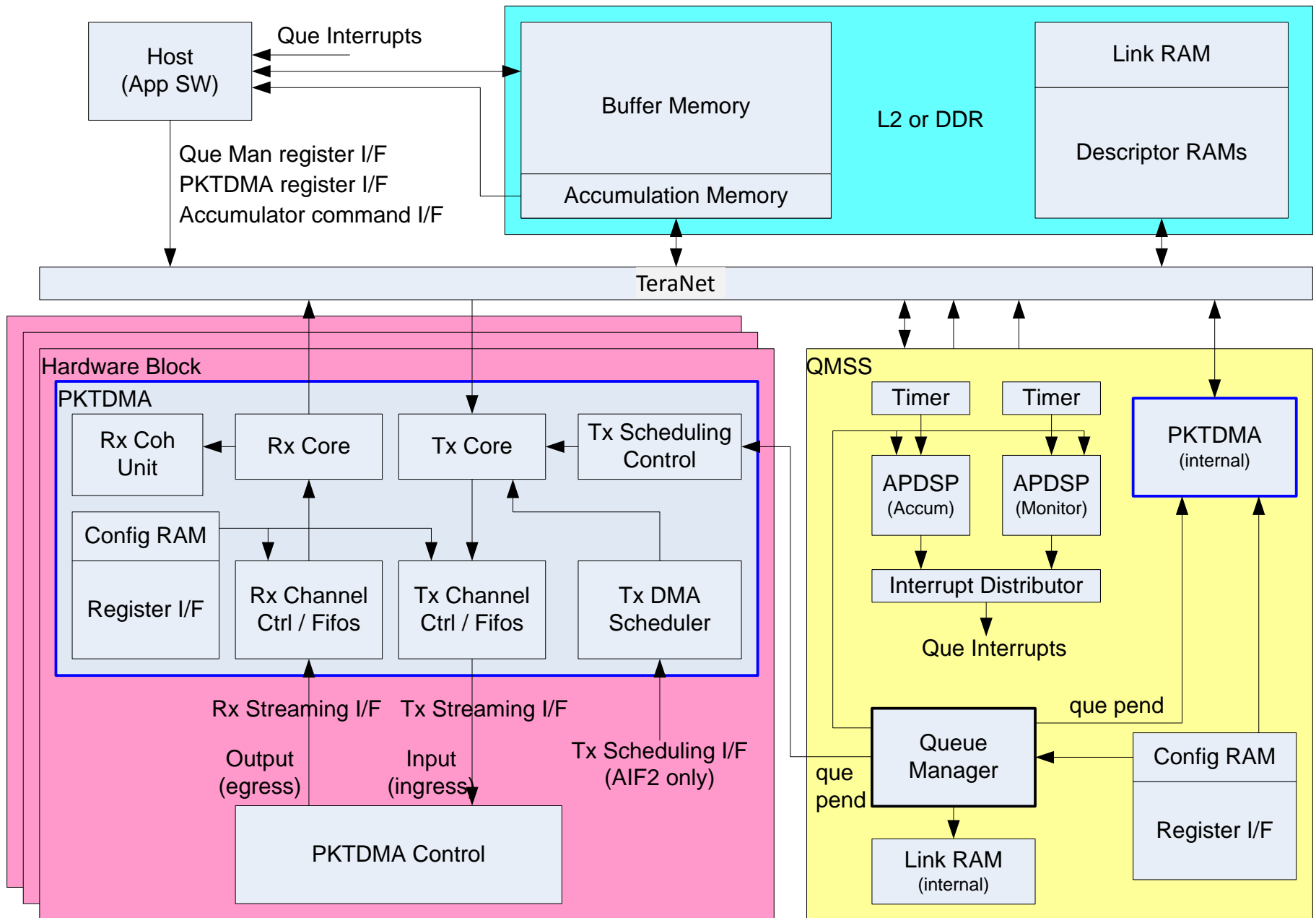
Hardware Semaphore

- Keystone supports 64 Hardware Semaphores.
- Direct Mode:
 - DSP core can get access by issuing a read command. If resource is free it will be immediately granted to the particular master.
- Indirect Mode:
 - DSP core can get access by issuing a write command to the request queue. If resource is free it will be granted to the particular master.
- Combined Mode:
 - This is the combination of both direct and indirect mode to best utilize the resource. If the resource is free then access is granted and if the resource is not free then a request will be put in the request queue.

EDMA3

- Three EDMA3 entities in Keystone DSP.
 - EDMA0 in 1/2 CPU clock and EDMA1,2 in 1/3 CPU clock
- Data moving between different memories.
 - High performance than memcpy() in large block.
 - Work independently of CPU.
- Transfer completion events for all transactions
 - Synchronize to TP-CC channels
 - Generate interrupt to any CPU

Navigator



Navigator Usage

- Exchanging messages between cores
 - Synchronize execution of multiple cores
 - Move parameters or arguments from one core to another
- Transferring data between cores
 - Output of one core as input to the second
 - Allocate memory in one core, free memory from another, without leakage
- Sending data to peripherals
- Receiving data from peripherals
- Load Balancing and Traffic Shaping
 - Enables dynamic optimization of system performance

Agenda

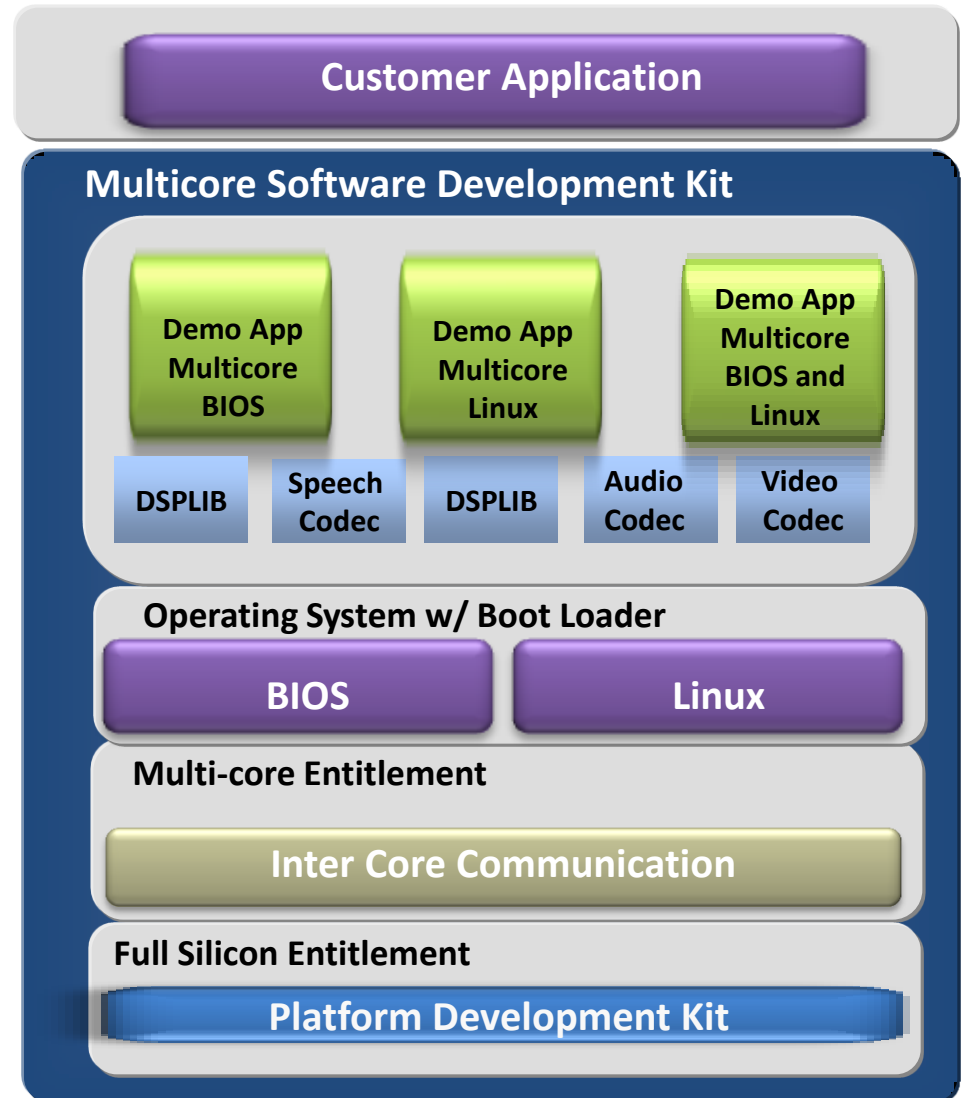
- Multi-core Design Overview
- Inter-core communications
 - Shared Memory
 - Inter-core Interrupt
 - Semaphore
 - EDMA3
 - Navigator
- **Multi-core Software Develop Kit (MCSDK)**
Introduction

What is MCSDK?

- The Multicore Software Development Kit (MCSDK) provides for customers to **quickly start developing embedded applications** on TI high performance multicore DSPs.
 - Uses the SYS/BIOS or Linux **real-time operating system**
 - Accelerates customer time to market by focusing on **ease of use** and **performance**
 - Provides **multicore programming** methodologies
- Available for free on the TI website bundled in one installer, all the software in the MCSDK is in source form along with pre-built libraries

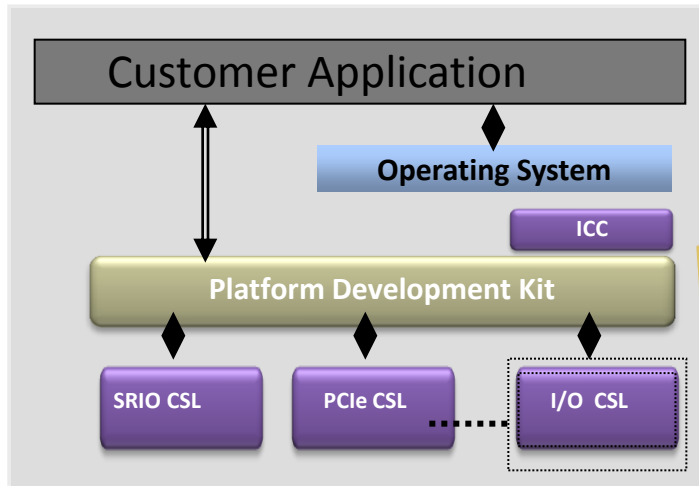
MultiCore Software Development Kit (MCSDK)

- Flexible development environment for the developer
- MCSDK will contain the following software layers
 - Demonstration applications
 - BIOS and Linux Operating System support
 - Platform Development Kit (PDK)
 - Inter Core Communication
 - Optimized DSP functions library
 - Optimized Audio, Video and Speech codecs



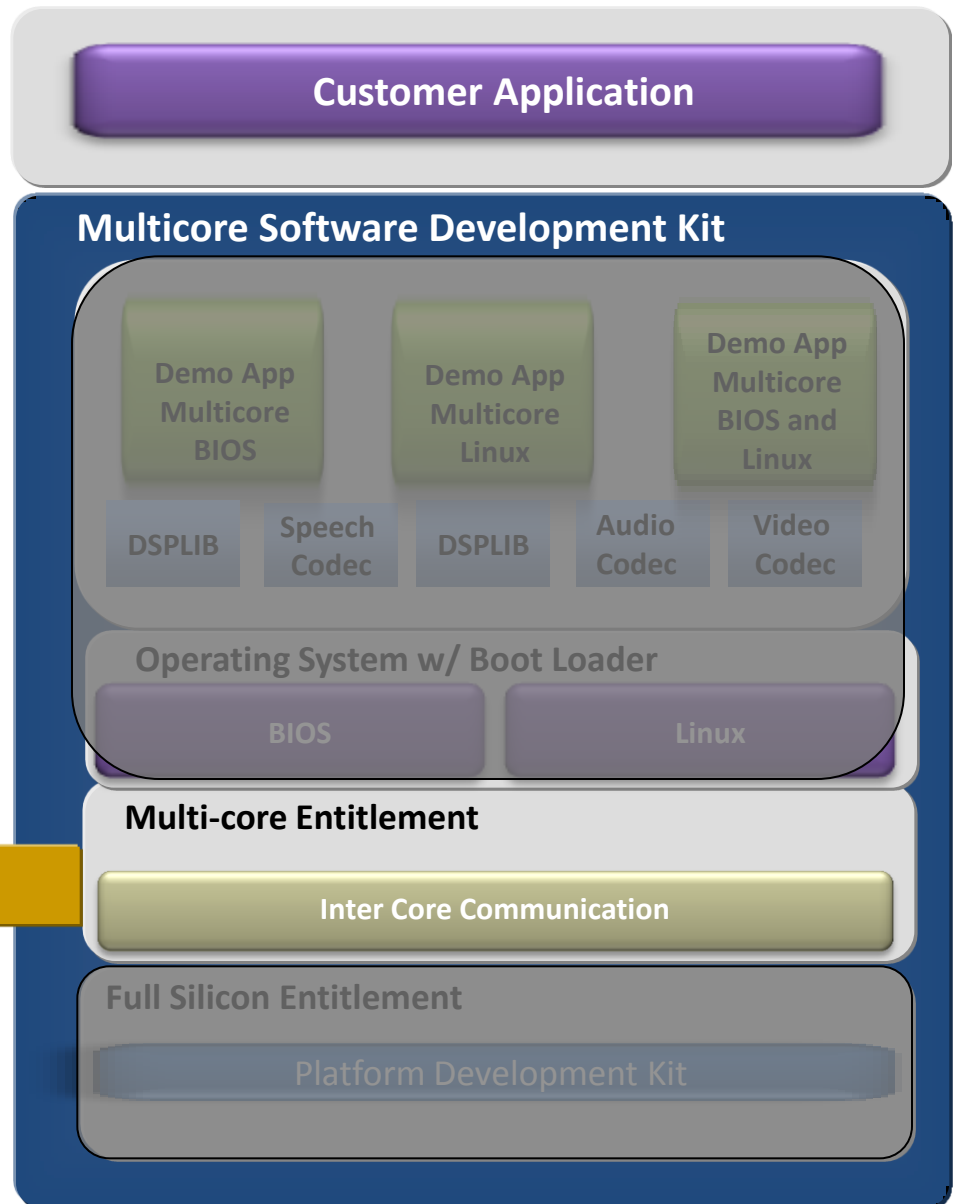
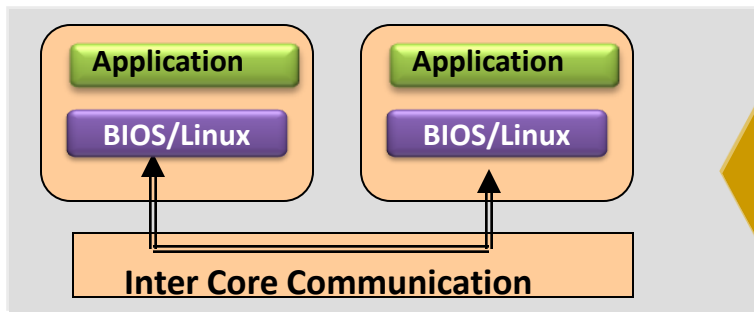
MCSDK

- PDK can be integrated with customer application using well defined “C” language APIs
- Chip Support Library (CSL) provides register layer abstraction
- Low Level Drivers (LLD) provide API support for many of the peripherals



MCSDK

- Inter Core Communication Layer provides functionality to communicate across different cores.
 - Uses on chip network switching fabric to provide efficient message based communication across different core.
 - Integrated with BIOS and Linux to facilitate easy integration with customer application
 - Provides capabilities to the application to manage load across the cores.



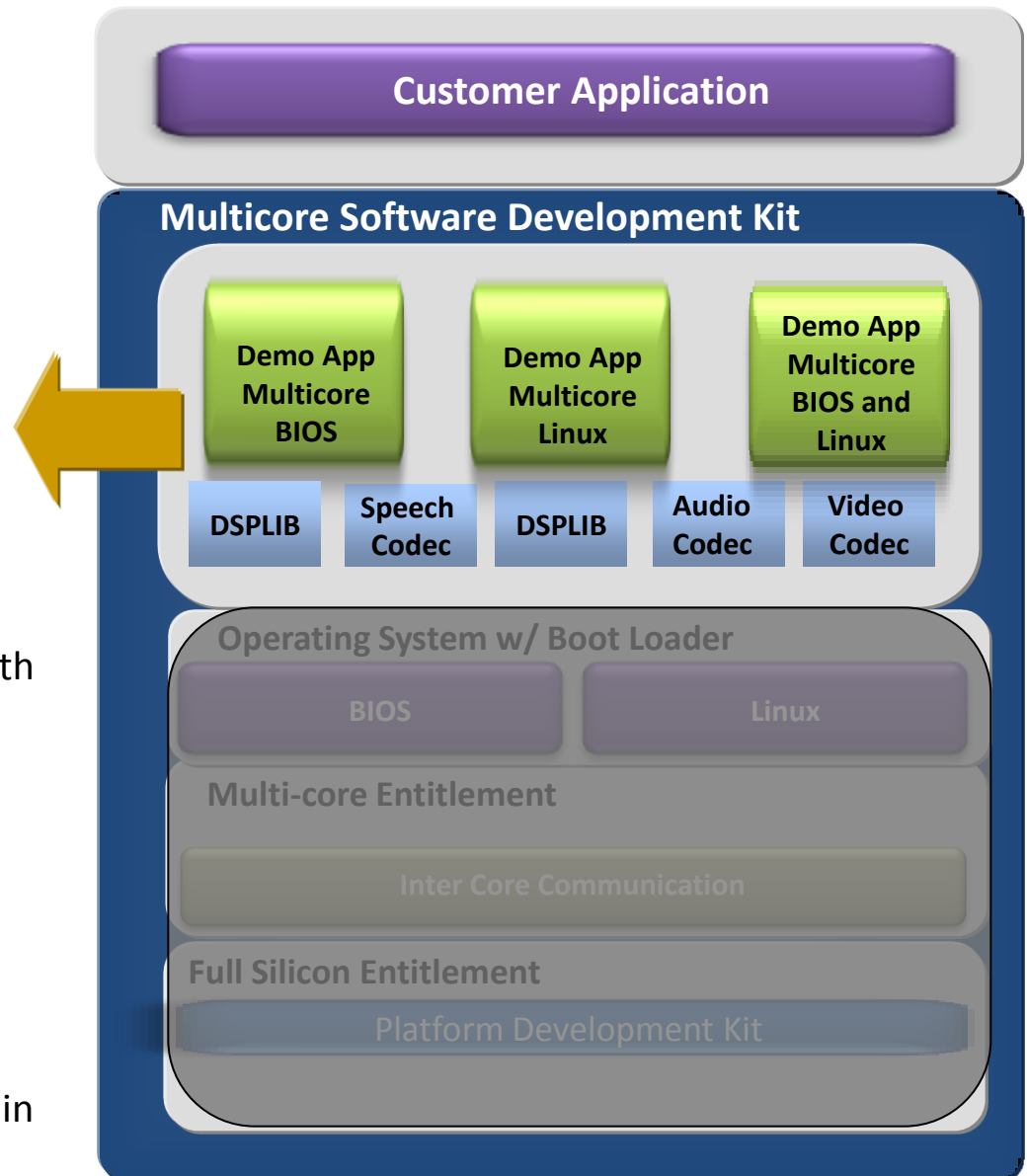
MCSDK

- BIOS
 - Ability to transport real time analysis data from multicore to external device using real time analysis.
 - Seamless interaction with inter core communication software layer.
 - Integrated and tested with platform development kit and Inter core communication layer
- Linux
 - Available from community or from TI
 - MMU less configuration
 - NAND based file system
 - Integrated and tested with platform development kit and Inter core communication layer



MCSDK

- Multicore BIOS Demonstration will show case:
 - Use of BIOS on all cores
 - Use of Inter Core Communication in BIOS framework
 - Use of Platform Demonstration Kit in BIOS framework.
- Multicore Linux Demonstration will show case:
 - Use of Linux on all cores
 - Use of Inter Core Communication in Linux framework
 - Use of Platform Demonstration Kit with Linux framework
- BIOS and Linux Demonstration will show case:
 - Use of Linux on one core and BIOS on remaining core
 - Use of inter core communication between applications using BIOS and Linux operating system
 - Use of platform development kit within Linux/BIOS framework



IMGLIB-Imaging Library

- **Features and Benefits**

- Natural C Source Code, Optimized C code with Intrinsic and associated test bench
- **Download c66x version from:** [Image and Video Processing Library \(IMGLIB\)](#)

Function Category

Subcategory

Image Analysis Functions

- Boundary and Perimeter Functions
- Dilation and Erosion Operation Functions
- Edge Detection Function
- Histogram Function
- Image Threshold Function

**~35
Functions**

Picture Filtering Functions

- Color Space Conversion Functions
- Convolution Function
- Error Diffusion Function
- Correlation Functions
- Median Filtering Function
- Pixel Expand Functions

**~25
Functions**

Compression/ Decompression Functions

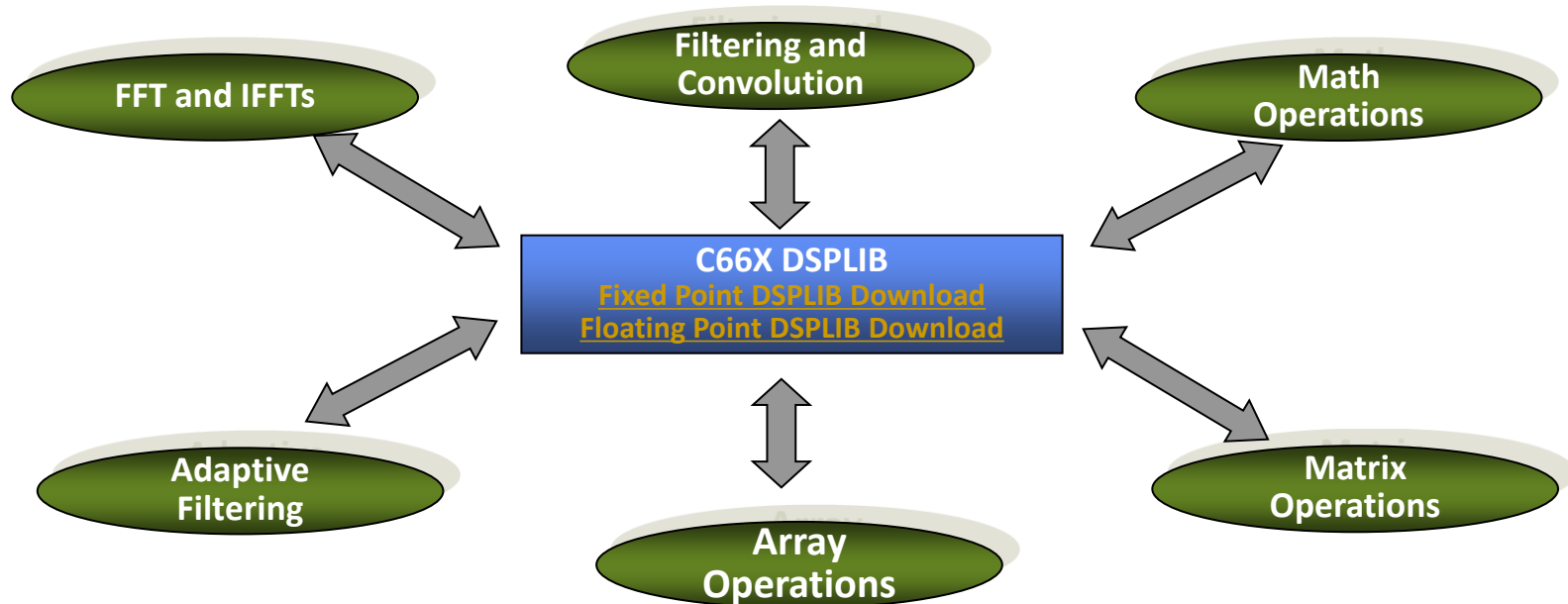
- Forward and Inverse DCT Functions
- High Performance Motion Estimation Functions
- MPEG-2 Variable Length Decoding Functions
- Quantization Function
- Wavelet Processing Functions

**~10
Functions**

DSPLIB – DSP Signal Processing Library

- **Features and Benefits**

- Natural C Source Code, Optimized C code with Intrinsics and associated test bench. Available for both Little and Big Endian
- C-callable routines, Free to Download. Both fixed point and floating point



- **C66x DSP Library Download Source Code (FREE) DSPLib**
Download [DSP Signal Processing Library\(DSPLIB\)](#)

KeyStone Multicore Software - Codecs

Voice

- G.711, G.722
- G.723, G.729
- AMR(NB/WB), EVRC-B
- Others

Video

- H.263
- H.264
- MPEG2
- MPEG4
- VC1/WMV9 Decode
- Others

Audio

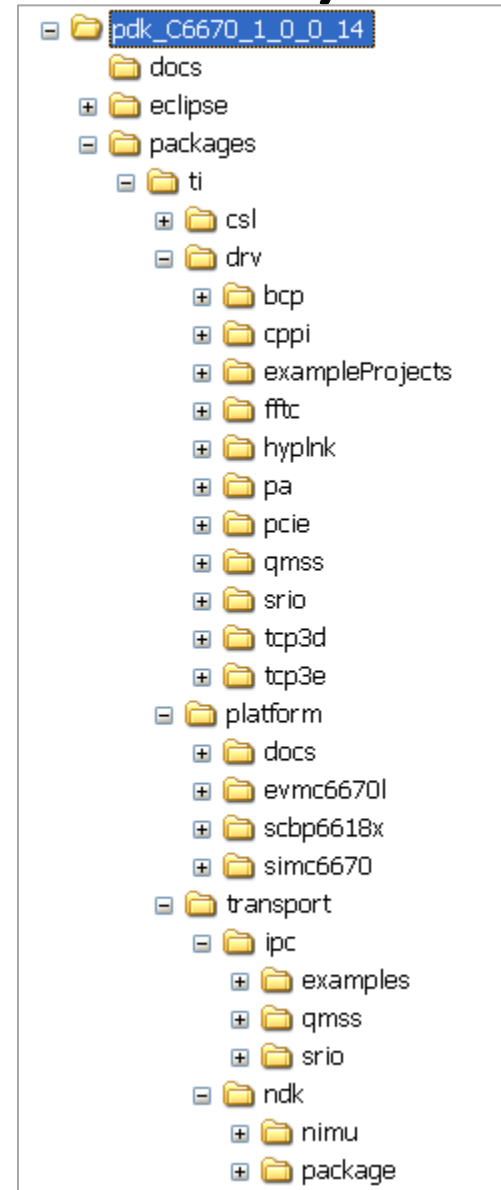
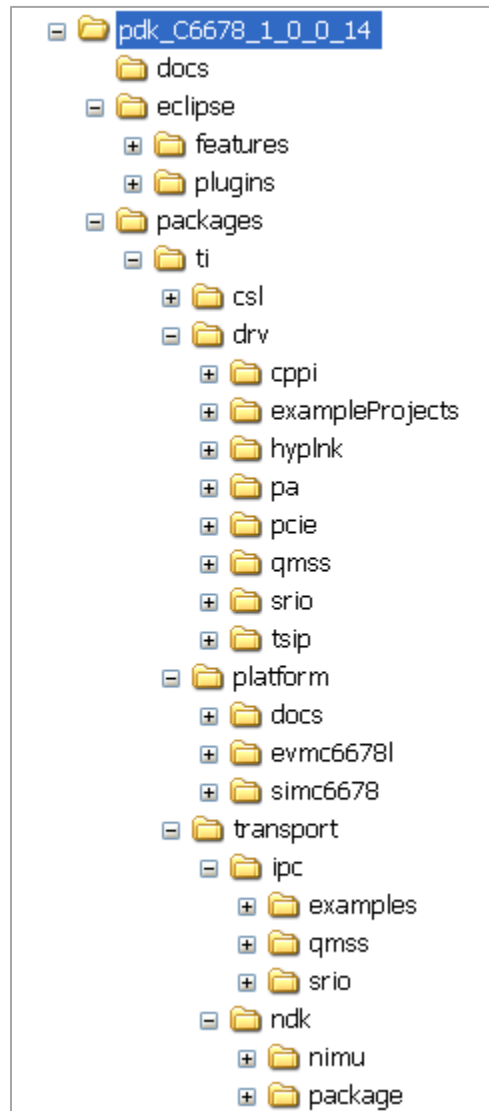
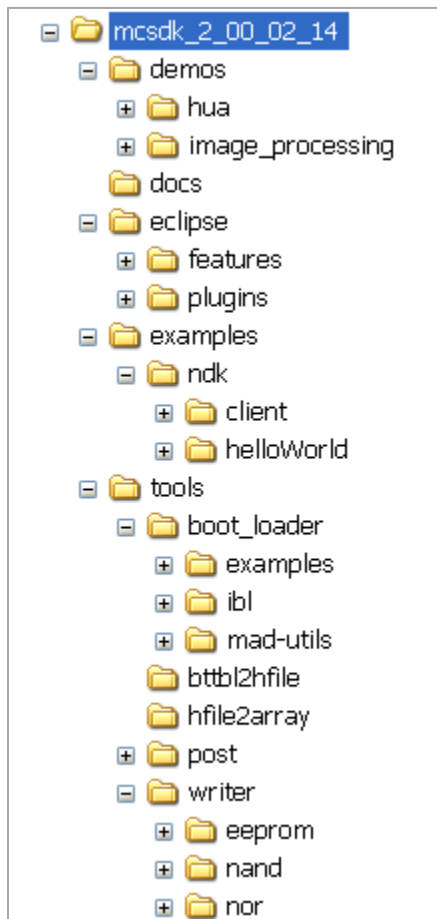
- MPEG1 Layer2
- AAC LC/HE
- AC3 2.0/5.1
- Sample Rate Conversion

Fax

- T.38
- Fax Modem

Some of the codecs require special licensing

Packaging Example (BIOS-MCSDK)



For More Information

- [Multicore Program Guide](#)
- Multicore articles, tools, and software are available at [Embedded Processors Wiki for the KeyStone Device Architecture](#).
- View the complete [C66x Multicore SOC Online Training for KeyStone Devices](#), including details on the individual modules.
- For questions regarding topics covered in this training, visit the support forums at the [TI E2E Community](#) and [德州仪器中文社区](#).