# Audio Framework Training

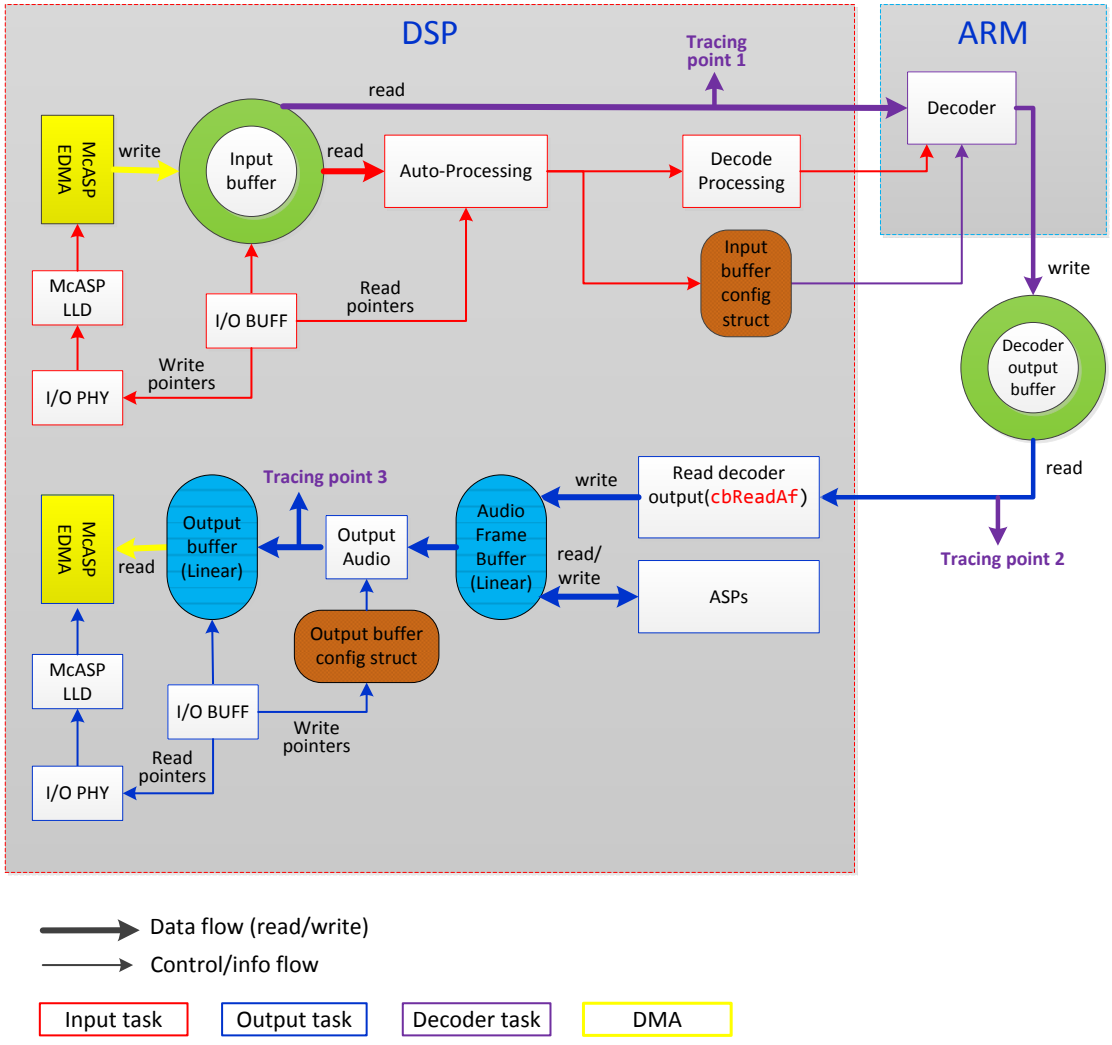
**Framework Overview**

**11/21/2018**



Texas Instruments

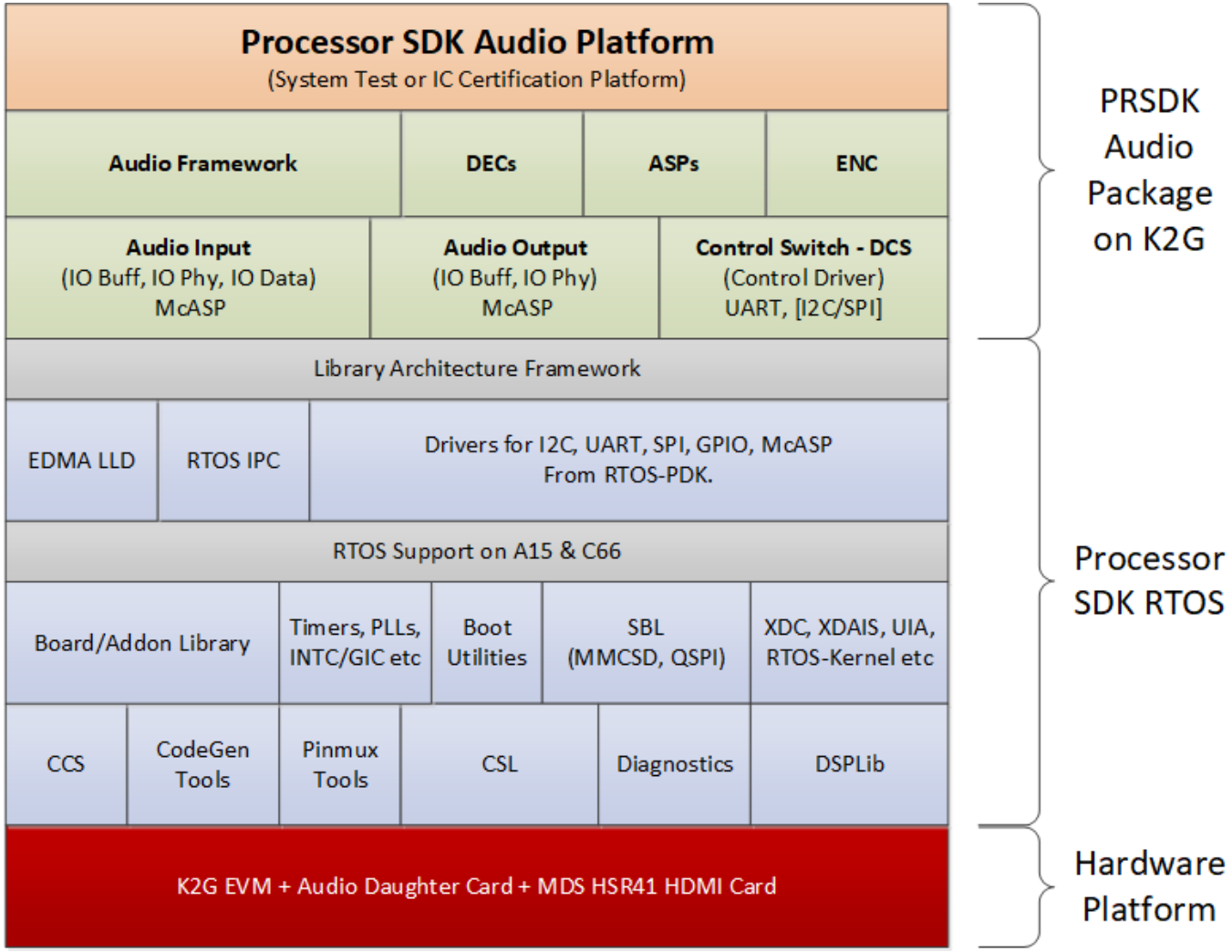
# System Overview






Texas Instruments

# Audio Software



# Framework History, DA8x Performance Audio Framework (PAF)

- Single Audio Stream Task (AST) for audio processing
  - Essentially audio output clock driven
  - Modified SIO/DEV driver used for audio IO

- Two DA8x C674x DSP @ 456 MHz for DTS:X solution
  - 1st DSP for decoding, 2nd DSP for PCM post-processing
  - Each DSP contains AST with entire PAF signal chain (IB, DEC, ASP, ENC, OB)
  - PCM transport between DSPs using McASP, metadata packed in PCM samples

# Framework History, K2Gx

- PASDK 1.2.x
  - Three ASTs for audio processing
    - Audio Stream Input Task (ASIT): manage audio input, driven by audio input clock
    - Audio Stream Decode Task (ASDT): decode slave, driven by decode messages from ASIT
    - Audio Stream Output Task (ASOT): PCM post-processing, audio output clock driven
  - Single K2Gx for DTS:X solution
    - ARM A15 @ 600 MHz: ASDT (DEC)
    - TI C66x DSP @ 600 MHz: ASIT (IB) & ASOT (ASP, ENC, OB)
    - PCM & metadata transport from ASDT to ASOT using Circular Buffer (CB) in Shared Memory
  - SIO/DEV port used for audio IO, dMAX replaced with EDMA3
  - PRSDK drivers used for all but audio IO, e.g. UART LLD

- PRSDK 5.x Audio Framework
  - Audio I/O completely refactored, SIO/DEV removed
  - McASP LLD used for low-level audio IO, full use of PRSDK drivers
  - ASTs fully Event driven, removed multiple Pend locations

# Hardware Interfaces

- Audio Input:
  - ADC: PCM, up to 8 channels
  - S/PDIF: PCM, DTS, DD
  - HDMI: PCM, DTS, DD, DDP, TrueHD, MAT
- Audio Output:
  - DAC: PCM, up to 16 channels
- Audio I/O connected via McASP (0, 1, or 2)
- Control I/O: UART, SPI, I2C
  - UART0 used on K2Gx EVM during development/testing

**McASP CONFIGURATION**

**(K2G EVM with Daughter Card)**

| McASP PORT | DEVICES | DEVICE MODE |
|---|---|---|
| McASP0 | DAC- PCM1690 x2 | SLAVE |
| | I2S HEADER | MASTER |
| McASP1 | ADC- PCM1865 x2 | SLAVE |
| McASP2 | HDMI TRANSMITTER (DEFAULT) | SLAVE |
| | AUDIO CODEC AIC3106 | |
| | DIR9001(DEFAULT) | MASTER |
| | COM8 | |

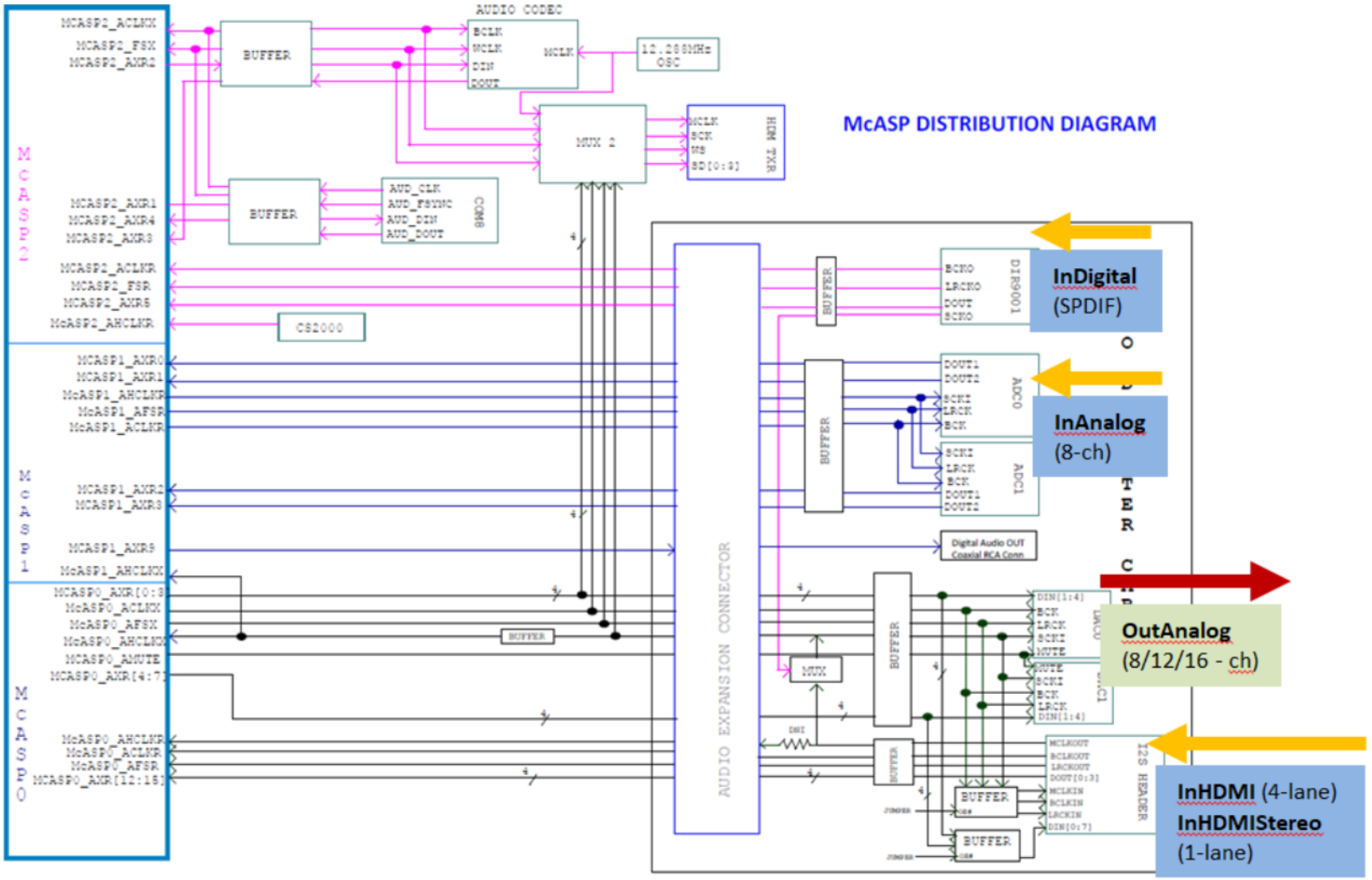
# Audio I/O – Clocking

- Analog
  - K2G equipped with on-board Audio-OSC
    - Fed by 22.5792 MHz crystal on K2Gx EVM
    - ‘AUDIO_OSCCLK’ available as internal source for all the McASP instances
  - ADCs clocked with McASP1 AHCLKR, i.e. McASP master
  - DACs clocked with McASP0 using same AUX clock source

- S/PDIF
  - DIR9001 PLL recovers input clock & generates three output clocks:
    - Master clock (DIR_SCKO)
    - Bit clock (DIR_BCKO)
    - Frame clock (DIR_LRCKO)
  - Bit & Frame clocks sufficient to master receive section of McASP2
  - Master clock used as external source for AHCLKX of McASP0, McASP masters DACs with internally generated Frame & Bit clocks

- HDMI
  - The “I2S Header” exposed on Audio DC interfaces to HDMI Repeater card, which supplies the necessary clocks
  - I2SHDR_McASP0_AFSR & I2SHDR_McASP0_ACLKR used to master the Frame & Bit clocks of the receive section of McASP0
  - I2SHDR_MCLKOUT used as external source for AHCLKK of McASP0, McASP masters DACs with internally generated Frame & Bit clocks

# Audio I/O – Shortcuts



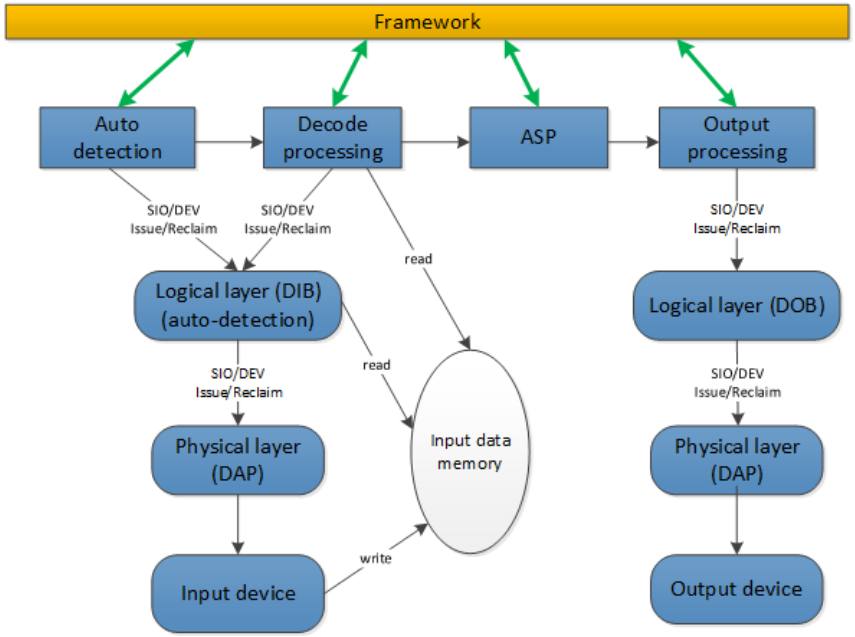
# Audio I/O – Refactoring

- "Stacked" SIO/DEV driver replaced with "flattened" driver model, audio I/O operations driven directly by ASIT/ASOT State Machines
  - Intelligence (e.g. auto-detection) in SIO/DEV moved to new audio I/O components
  - SIO/DEV calls in framework replaced with calls to new audio I/O components
  - Decision making in SIO/DEV moved to framework
  - PRSDK McASP LLD used for low-level Audio I/O instead of directly using EDMA3 LLD

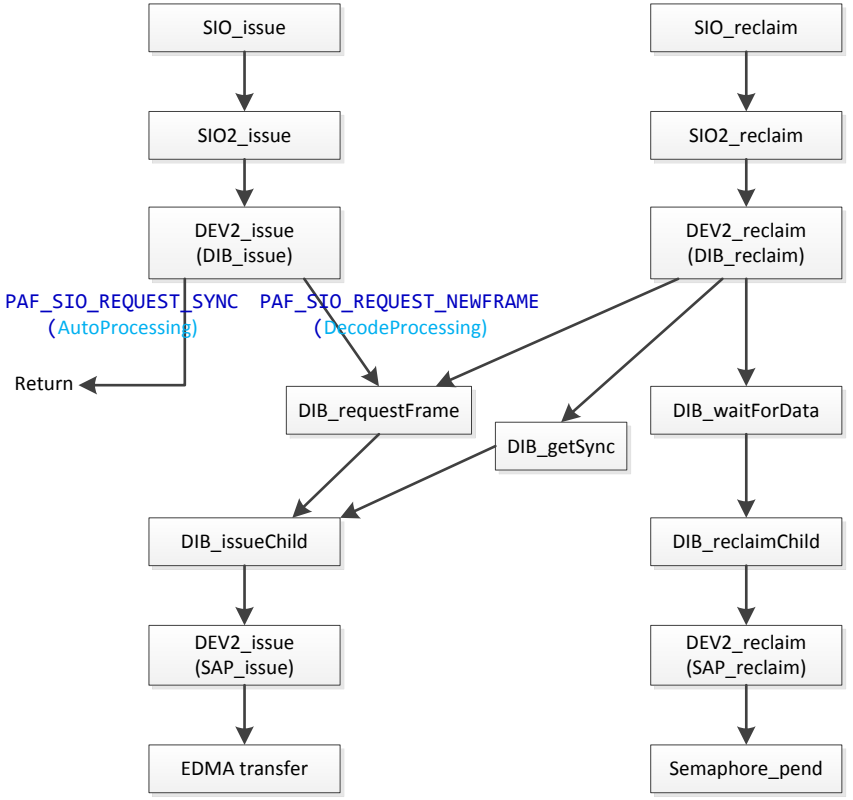
- Four main reasons for refactoring:
  1. SIO/DEV deprecated in SYSBIOS
  2. SIO/DEV blocks Task execution in multiple locations
     - Increases response time of Task to time critical operations (e.g. EDMA reconfiguration)
     - Unable to accommodate short-term peak MIPS overflow (beyond 100%)
  3. SIO/DEV stacked driver difficult to develop, maintain & debug due to multiple layers in model
  4. Too much intelligence embedded in stacked in SIO/DEV drivers, flattened model more flexible

# Audio I/O – SIO/DEV Complexity

## Block Diagram of SIO/DEV-Based I/O Drivers




## SIO_issue & SIO_reclaim in auto-detection






Texas Instruments

# Audio I/O – New I/O Components

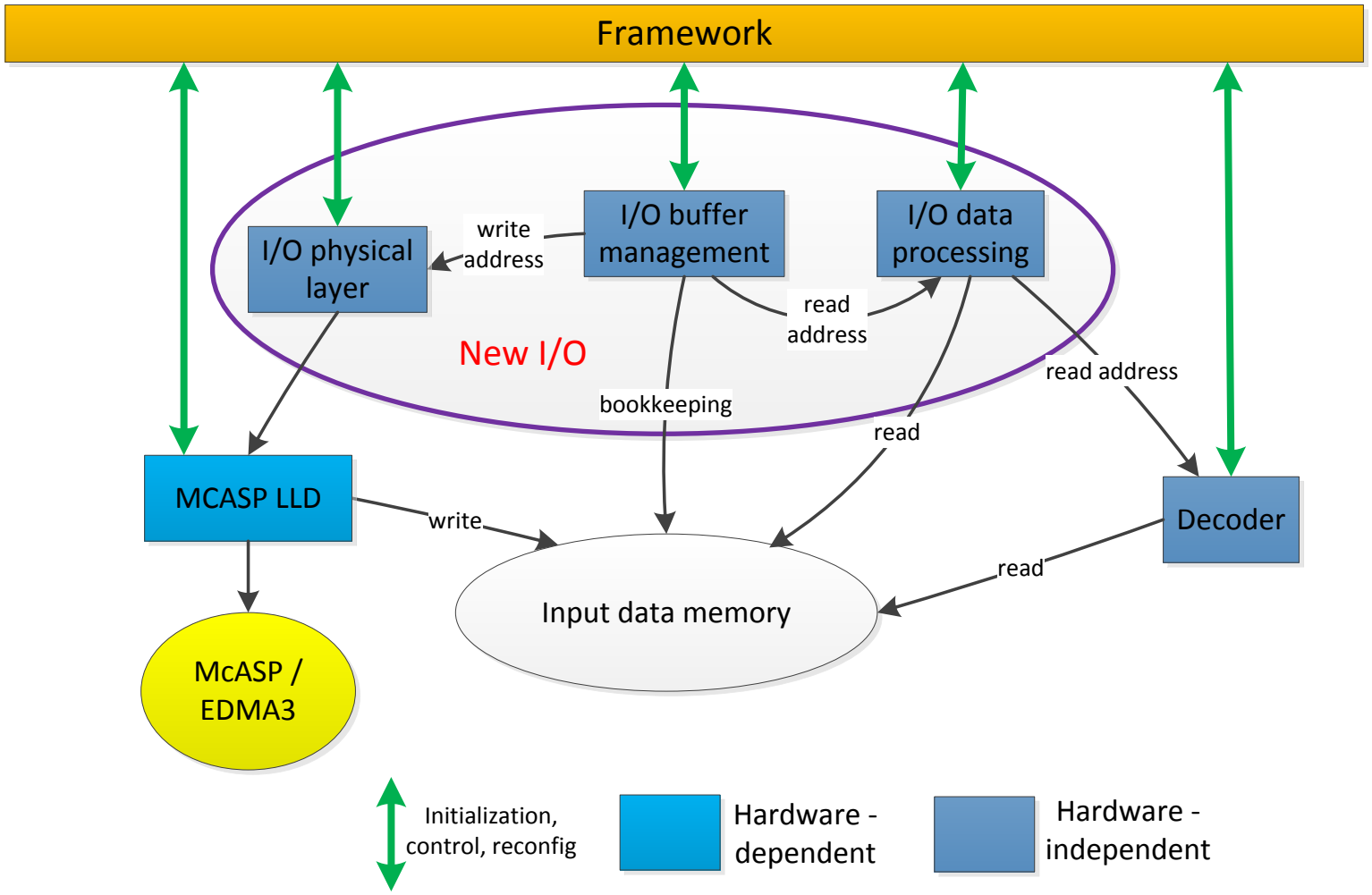
- Three components
  - IO Buff: memory pool/buffer management
  - IO Phy: physical layer interacting with McASP LLD
  - IO Data: data processing (auto detection)

- Every component instantiated by framework, multiple instances execute concurrently
  - ASIT & ASOT use separate IO Buff & IO Phy instances.
    ASIT uses IO Data instance for auto-detection, no ASOT IO Data instance.
  - Each IO Buff instance associated w/ IO buffer
  - Each IO Phy instance associated w/ McASP LLD handle & I/O Buff handle
  - Each IO Data instance associated w/ I/O Buff Handle

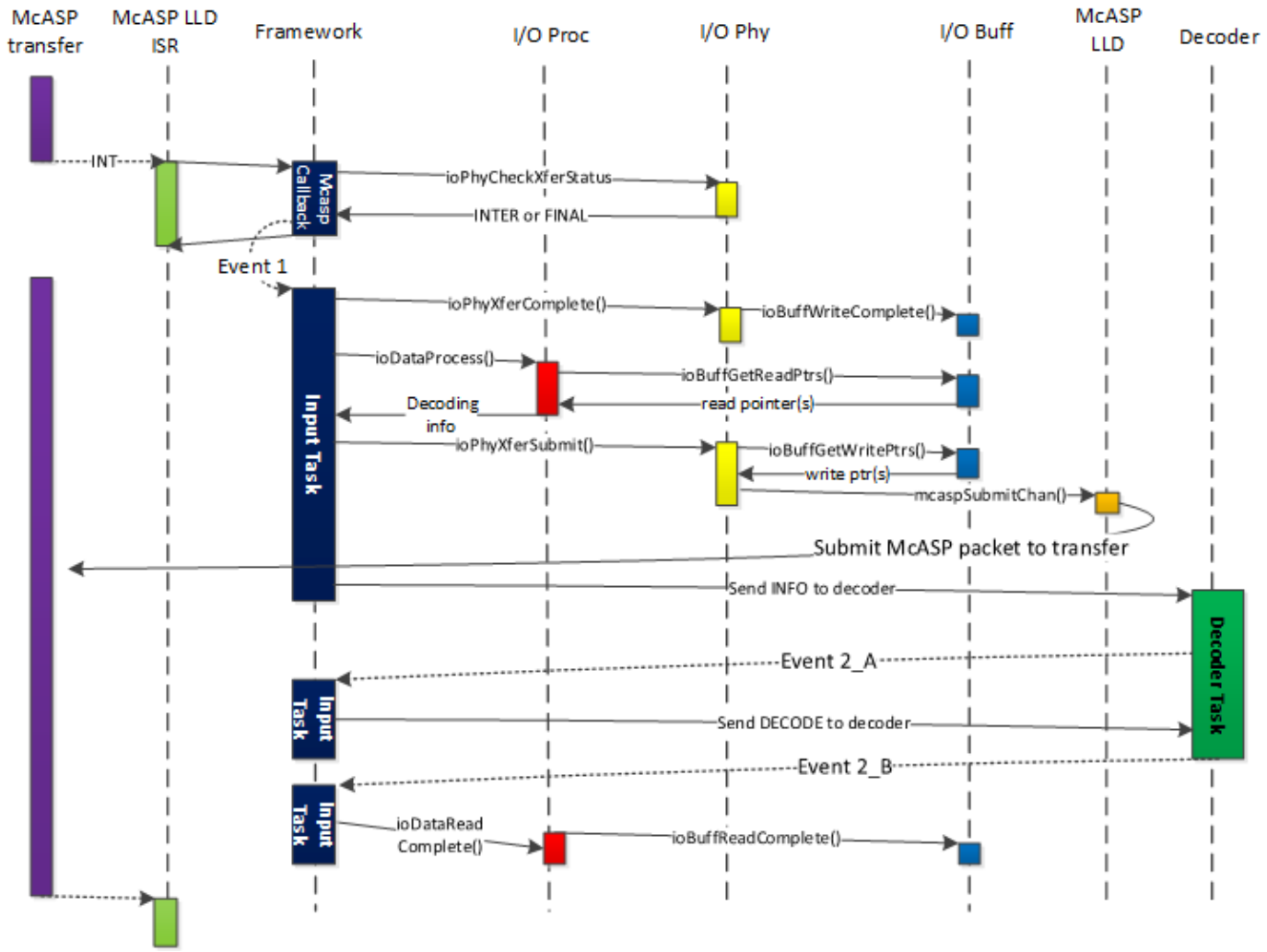
- IO Buff, Audio Input
  - Provides memory pointers to IO Phy for data write
  - Provides memory pointers to IO Data for data read
- IO Buff, Audio Output
  - Provides memory pointers to PCE for data write
  - Provides memory pointers to IO Phy for data read

- IO Data, Audio Input
  - Reads input data from I/O buffers
  - Auto-detects encoding format
  - Prepares frame of data for decoders to decode

# Audio I/O – New I/O in ASIT



# Audio Input Sequence Diagram






Texas Instruments

# DSP HWIs

| HWI | Config | Int Num | Trigger | Timing | Description |
|---|---|---|---|---|---|
| HwiIpc | Dynamic | 5 | Rx decode ACK IPC message from ARM | Asynchronous | IPC ISR |
| HwiEvtCmb[0] | Static & Dynamic | 7 | Audio Input EDMA complete<br><br>Audio Output EDMA complete | Audio Input IB wrap:<br>• No wrap, 1 transfer for frame<br>• Wrap, 2 transfers for frame<br><br>Audio Input, auto-detection mode (no wrap):<br>• 1xI2S: 5.33 msec.<br>• 4xI2S: 5.33 msec.<br><br>Audio Input, decode mode (no wrap):<br>• HDMI, PCM 44.1 kHz: 21.54 msec.<br>• HDMI, PCM 48 kHz: 21.33 msec.<br>• HDMI, DTS: depends on Stream Format, Frame Period & Frame Rate (kHz), e.g.<br>DTS Type 2, 1024, 48: 21.33 msec.<br>DTS Type 4, 2048, 192: 10.667 msec.<br>• HDMI, Dolby: depends on Stream Type, e.g.<br>DDP: 20 msec.<br>MAT/THD: 10 msec.<br><br>Audio Output: depends on Fs & Output frame length, e.g.<br>(256 samples / 48 kHz) = 5.33 msec. | Event Combiner ISR, EVT0 |
| HwiEvtCmb[1] | Static & Dynamic | 8 | UART Tx/Rx chars complete | Asynchronous | Event Combiner ISR, EVT1 |
| HwiSysTimer | Static | 14 | DSP HW timer | 1 msec. | System Timer ISR |

# DSP SWIs & Clocks

| SWI | Config | Priority | Trigger | Timing | Description |
|---|---|---|---|---|---|
| SwiSysClk | Static | 15 | DSP HW timer | HwiSysTimer | SYSBIOS System Clock |
| SwiAsop | Dynamic | 5 | Audio Output EDMA complete (SWI post in McASP LLD callback) | HwiEvtCmb, Audio Output | Manage Audio Output<br>• Initiate Output transfer<br>• Trigger TaskAsop |
| SwiUart | Dynamic | 2 | UART Rx complete (SWI post in UART LLD callback) | HwiEvtCmb, UART | UART Rx S-records in alpha command (intr-callback mode) |

| Clock | Config | One-Shot/ Periodic | Period | Timeout | Description |
|---|---|---|---|---|---|
| ClkAsotWake | Static | Periodic | 1 system tick | - | Wake timer for polling start of decoding in ASOT |
| ClkUartTimeout | Dynamic | One-shot | - | 11 system tick | Check timeout for UART Rx chars in S-record (intr-callback mode) |

# DSP Audio Stream Tasks

| Task | Config | Priority | Trigger | Timing | Description |
|---|---|---|---|---|---|
| TaskAsip | Static | 5 | Audio Input EDMA complete<br>(Event post in McASP LLD callback)<br><br>Rx decode ACK IPC message from ARM<br>(Event) | Audio Input, auto-detection mode:<br>• 1xI2S: 5.33 msec.<br>• 4xI2S: 5.33 msec.<br><br>Audio Input, decode mode:<br>• HDMI, PCM 44.1 kHz: 21.54 msec.<br>• HDMI, PCM 48 kHz: 21.33 msec.<br>• HDMI, DTS: depends on Stream Format, Frame Period & Frame Rate (kHz), e.g.<br>DTS T2, 1024, 48: 21.33 msec.<br>DTS T4, 2048, 192: 10.667 msec.<br>• HDMI, Dolby: depends on Stream Type, e.g.<br>DDP: 20 msec.<br>MAT/THD: 10 msec.<br><br>Rx ACK message (Info, Decode, etc.): Asynchronous<br>• Average same as input ,decode mode<br>• Min/Max depend on ARM processing time | Audio Stream Input Processing<br>• Manage Audio Input<br>• Auto detection<br>• Tx decode IPC messages to ARM (Activate, Reset, Info, Decode, Deactive) |
| TaskAsop | Static | 4 | ClkAsotWake<br>(Event post in clock function)<br><br>SwiAsop<br>(Event post in SWI function) | Wake: ClkAsotWake timing<br><br>Audio Output: SwiAsop timing | Audio Stream Output Processing<br>• Read decoded output from CB<br>• Audio Stream Processing (ASP) chain |

# DSP Control Tasks & IDLE Functions

| Task | Config | Priority | Trigger | Timing | Description |
|---|---|---|---|---|---|
| TaskSysInit | Static | 6 | SYSBIOS Scheduler | Run-once | System Initialization<br>• IPC attach<br>• ASP Master Message initialize<br>• CB control initialize<br>• Heap Manager initialize |
| TaskAip | Static | 3 | DSP System Clock<br>(Task sleep) | 100 msec. | Alpha Interval Processing.<br>Embedded Control<br>• at-boot alpha commands<br>• at-interval alpha commands |
| TaskAfp | Static | 2 | Rx alpha command from serial port<br>(SwiUart sem post) | Asynchronous | Alpha Function Processing.<br>Control IO: process alpha commands. |
| TaskSysStream | Static | 1 | DSP System Clock<br>(Task sleep) | 1 sec. | System Stream.<br>System configuration & status:<br>• DEC CCR<br>• BM OC<br>• SPDIF pre-emphasis & DEM mode<br>• CPU load stats & PFP config/stats |

| IDLE Function | Config | Order | Description |
|---|---|---|---|
| IdleNotifyInfoChange | Static | 0 | Notify Host of change in system info (e.g. Decode type) using GPIO. |

# ARM HWIs & Tasks

| HWI | Config | Priority | Trigger | Timing | Description |
|---|---|---|---|---|---|
| HwiSysTimer | Static | 30 | DSP HW timer | 1 msec. | System Timer ISR |
| HwiIpc | Dynamic | 33 | Rx decode IPC message from DSP | Asynchronous | IPC ISR |

| Task | Config | Priority | Trigger | Timing | Description |
|---|---|---|---|---|---|
| TaskSysInit | Static | 4 | SYSBIOS Scheduler | Run-once | System Initialization:<br>• IPC attach<br>• ASP Slave Message initialize<br>• CB control initialize<br>• Heap Manager initialize |
| TaskAsdp | Static | 3 | Rx decode IPC message from DSP (Info, Decode, etc.) | Asynchronous<br><br>Info, Decode:<br>• Average same as Audio Input, decode mode<br>• Min/Max depends on ARM processing time | Audio Stream Decode Processing<br>• Rx decode IPC messages from DSP (Activate, Reset, Info, Decode, Deactive)<br>• Perform decoding |

# Memory Map

| Physical Memory | Description | Base Address | Size (Bytes) | Note |
|---|---|---|---|---|
| DSP L2 | DSP Level 2 SRAM (DSP internal) | 0080_0000 | 1048576 | |
| MSMC | Multicore Shared Memory Controller SRAM (K2Gx internal) | 0C00_0000 | 1048576 | |
| DDR | Double Data Rate SDRAM (K2Gx external) | 8000_0000 | 2147483648 | DDR on K2Gx EVM |

| Memory Region | Description | Base Address | Size (Bytes) | Physical Memory |
|---|---|---|---|---|
| L2SRAM | DSP L2 SRAM region | 0080_0000 | 786432 | DSP L2 |
| SR_MSMC | DSP/ARM MSMC Shared Region (IPC managed) | 0C00_0000 | 524288 | MSMC |
| HOST_MSMC | ARM MSMC dedicated region | 0C08_0000 | 196608 | MSMC |
| MSMC_RSVD | DSP MSMC reserved region (boot load) | 0C0B_0000 | 327680 | MSMC |
| SR_0 | DSP/ARM DDR Shared Region (IPC managed) | 8000_0000 | 2097152 | DDR |
| COMMON_DDR3 | DSP/ARM DDR shared region (application managed) | 8020_0000 | 6291456 | DDR |
| SR_DDR3 | DSP/ARM DDR Shared Region (IPC managed) | 8200_0000 | 16777216 | DDR |
| COMMON2_DDR3 | DSP/ARM DDR Non-Cache Shared Region (IPC managed) | 8100_0000 | 16777216 | DDR |
| HOST_DDR3 | ARM DDR dedicated region | 8300_0000 | 50331648 | DDR |
| CORE0_DDR3 | DSP DDR dedicated region | 8600_0000 | 67108864 | DDR |
| DDR3 | DDR shared region (application managed) | 8A00_0000 | 1979711488 | DDR |

# Heaps

| Heap | Memory Region | Size (Bytes) | Local / Shared | Defined | Note |
|---|---|---|---|---|---|
| DSP heapMemL2Sram | L2SRAM | 563200 | Local | pasrc/test_dsp/app.cfg | |
| srHeapMsmc | SR_MSMC | 524288 | Shared | pasrc/shared/ipc.cfg.xs | |
| ARM heapMemMsmcSram | HOST_MSMC | 196608 | Local | pasrc/test_arm/app.cfg | |
| DSP heapMemMsmcSram | MSMC_RSVD | 131072 | Local | pasrc/test_dsp/app.cfg | Memory region reserved for boot, but heap can be used by application. |
| srHeap | SR_0 | 2097152 | Shared | pasrc/shared/ipc.cfg.xs | |
| srHeapDdr3 | SR_DDR3 | 16777216 | Shared | pasrc/shared/ipc.cfg.xs | |
| srHeapNonCacheDdr3 | COMMON2_DDR3 | 16777216 | Shared | pasrc/shared/ipc.cfg.xs | |
| ARM heapMemDdr3 | HOST_DDR3 | 5350528 | Local | pasrc/test_arm/app.cfg | |
| DSP heapMemDdr3 | CORE0_DDR3 | 7496256 | Local | pasrc/test_dsp/app.cfg | |

# Stacks

| DSP Stack | Memory Region | Size (Bytes) | Defined | Note |
|---|---|---|---|---|
| SystemStack | DSP_L2 | 8192 | pasrc/test_dsp/app.cfg | Can be moved out of L2 |
| TaskAfpStack | CORE0_DDR3 | 4096 | pasrc/test_dsp/app.cfg | |
| TaskAipStack | CORE0_DDR3 | 4096 | pasrc/test_dsp/app.cfg | |
| TaskAsipStack | CORE0_DDR3 | 16384 | pasrc/test_dsp/app.cfg | |
| TaskAsopStack | CORE0_DDR3 | 16384 | pasrc/test_dsp/app.cfg | |
| TaskSysInitStack | CORE0_DDR3 | 4096 | pasrc/test_dsp/app.cfg | |

| ARM Stack | Memory Region | Size (Bytes) | Defined | Note |
|---|---|---|---|---|
| SystemStack | HOST_MSMC | 8192 | pasrc/test_arm/app.cfg | Links to HOST_DDR3 despite app.cfg setting, no conflicting setting linker command file |
| TaskAsdpStack | HOST_DDR3 | 36864 | pasrc/test_arm/app.cfg | |
| TaskSysInitStack | HOST_DDR3 | 4096 | pasrc/test_arm/app.cfg | |

- System (Program) Stack used for:
  - HWI context save/restore
  - SWI context save/restore
- Task Stack used for:
  - Local variables
  - Local function calls (stack frame, passing arguments, etc.)
  - Two full HWI contexts + scheduler variables. Additional HWI contexts saved to System Stack
- Run-time stack usage not analyzed

# Static Memory Usage

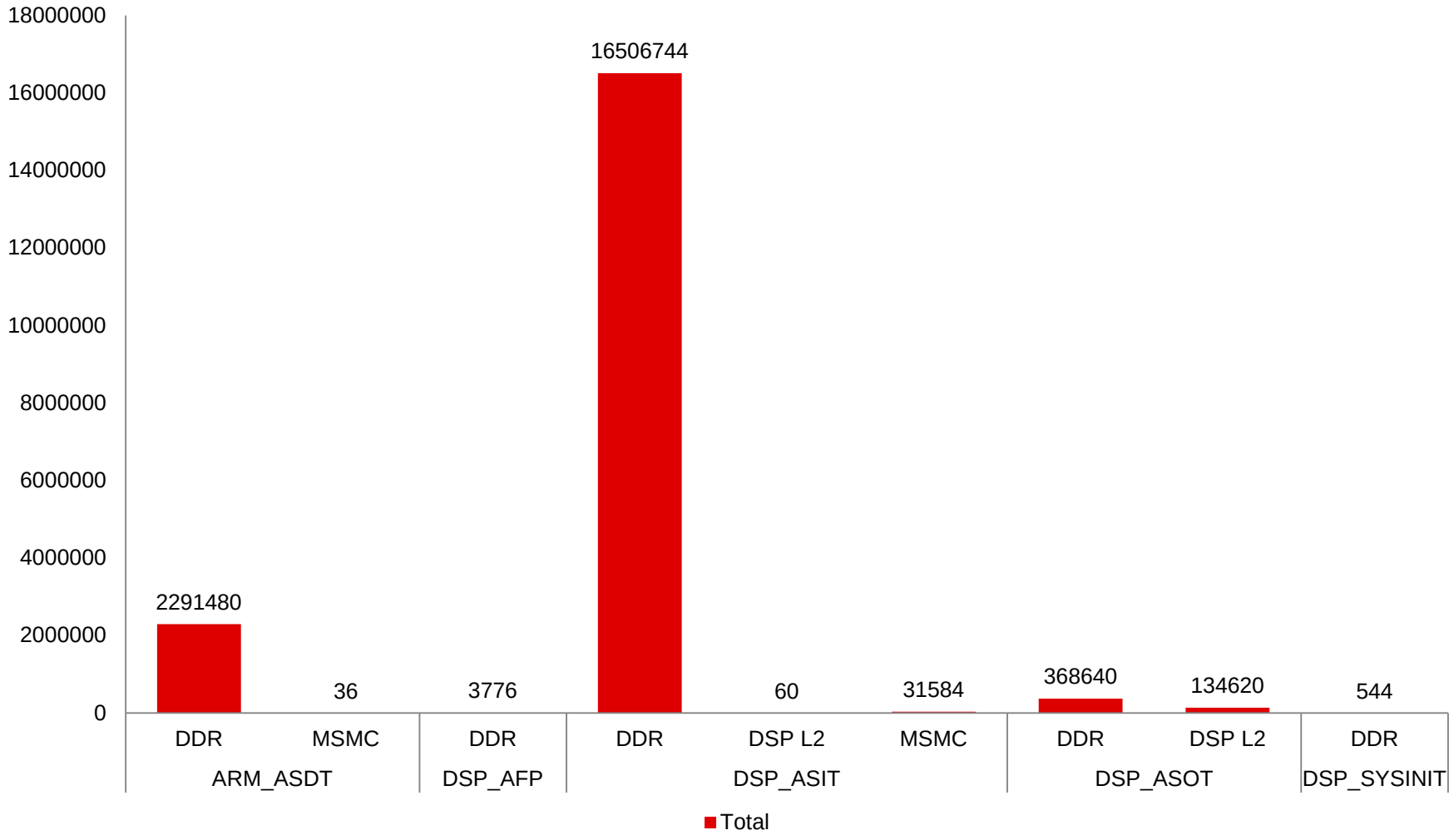
| Memory Type | Used (KB) | % Total | % L2 RAM |
|---|---|---|---|
| Code | 0.00 | 0.00% | 0.00% |
| Heap | 550.00 | 98.56% | 53.71% |
| Initialized_Data | 0.00 | 0.00% | 0.00% |
| System_Stack | 8.00 | 1.43% | 0.78% |
| Task_Stack | 0.00 | 0.00% | 0.00% |
| Unintialized_Data | 0.02 | 0.00% | 0.00% |
| **Total** | **558.02** | **100.00%** | **54.49%** |

| Memory Type | Used (KB) | % Total | % DDR RAM |
|---|---|---|---|
| Code | 2164.22 | 4.01% | 0.10% |
| Heap | 50577.69 | 93.61% | 2.41% |
| Initialized_Data | 389.46 | 0.72% | 0.02% |
| System_Stack | 8.00 | 0.01% | 0.00% |
| Task_Stack | 90.00 | 0.17% | 0.00% |
| Unintialized_Data | 802.06 | 1.48% | 0.04% |
| **Total** | **54031.42** | **100.00%** | **2.58%** |

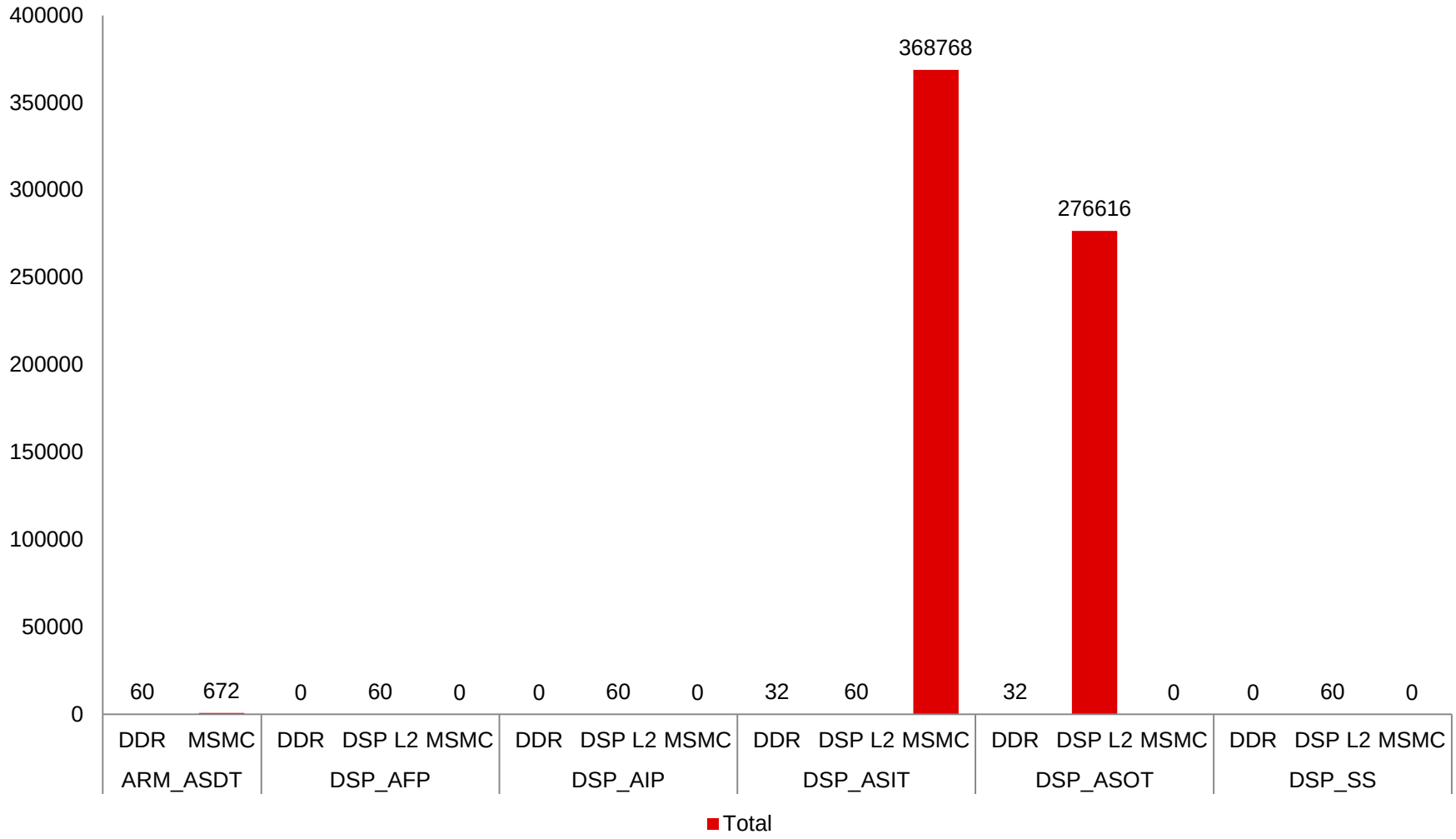
| Memory Type | Used (KB) | % Total | % MSMC RAM |
|---|---|---|---|
| Code | 0.00 | 0.00% | 0.00% |
| Heap | 704.00 | 100.00% | 68.75% |
| Initialized_Data | 0.00 | 0.00% | 0.00% |
| System_Stack | 0.00 | 0.00% | 0.00% |
| Task_Stack | 0.00 | 0.00% | 0.00% |
| Unintialized_Data | 0.00 | 0.00% | 0.00% |
| **Total** | **704.00** | **100.00%** | **68.75%** |

| Memory Type | Used (KB) | %Total |
|---|---|---|
| Code | 2164.22 | 3.91% |
| Heap | 51831.69 | 93.74% |
| Initialized_Data | 389.46 | 0.70% |
| System_Stack | 16.00 | 0.03% |
| Task_Stack | 90.00 | 0.16% |
| Unintialized_Data | 802.08 | 1.45% |
| **Total** | **55293.44** | **100.00%** |

# Heap Usage – Framework



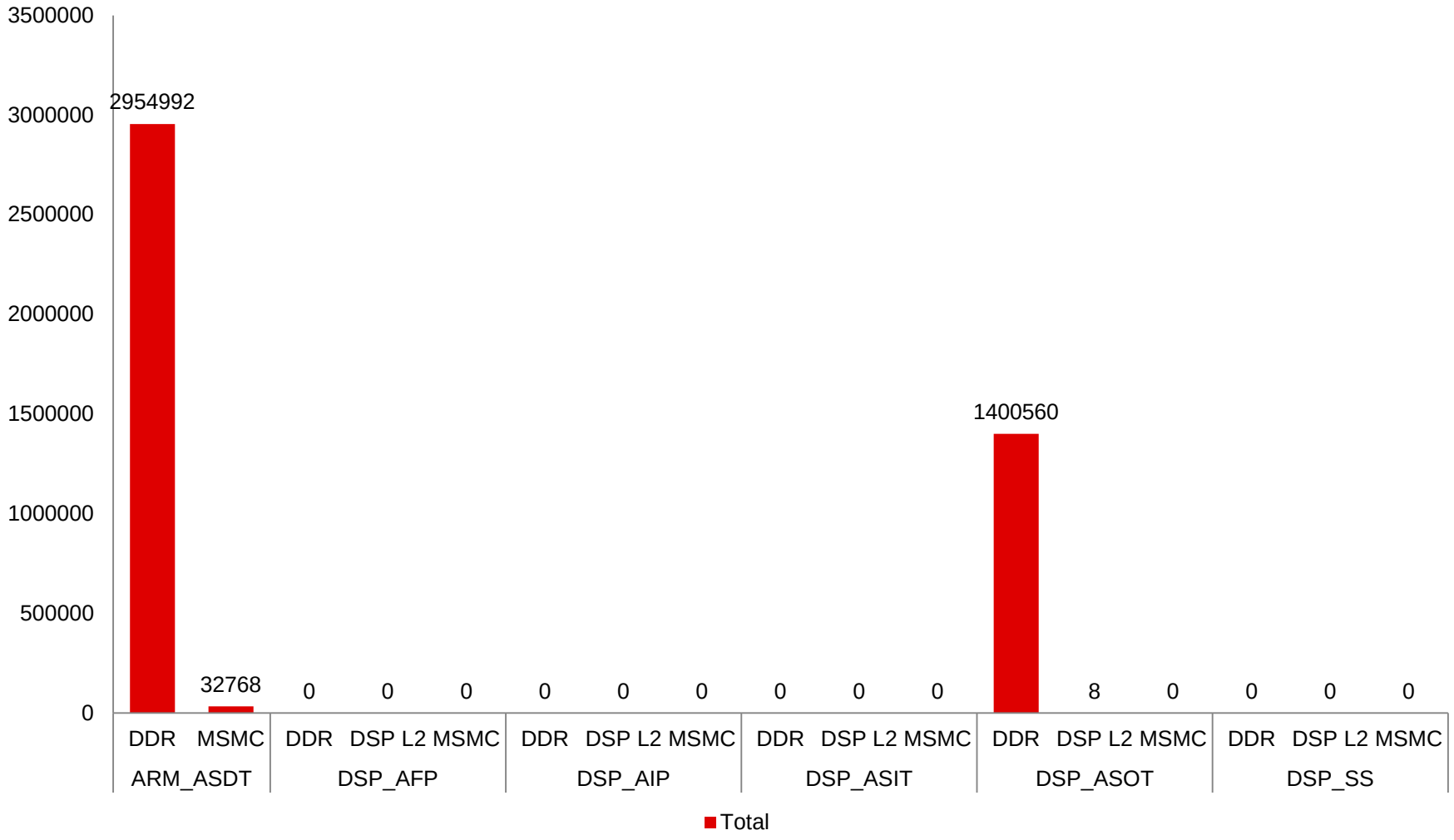
# Heap Usage – Framework Components






Texas Instruments

# Heap Usage – DTS Components






Texas Instruments

# PRSDK Inter-Processor Communications (IPC) Module

- IPC contains packages for communication between processors in multi-processor environment
  - Includes message passing, streams & linked lists
  - Works transparently in uni- & multi-processor configurations
  - Designed for SYS/BIOS applications
  - Can be used to communicate with:
    - threads on same processor
    - threads on other processors running SYS/BIOS

- Several IPC packages employed by framework
  - Shared Region
    - Shared memory manager and address translator
    - Creates and stores a local shared memory region table. Table contains processor's view for shared regions in system.
  - MessageQ
    - Structured sending & receiving of variable length messages
    - Messages sent & received by placement in and removal from a message queue
  - GateMP
    - Multi-processor gate that provides local and remote context protection
    - Entering a GateMP can prevent preemption by another thread running on same processor & simultaneously prevent a remote processor from entering same gate
    - GateMP's typically used to protect reads/writes to a shared resource, such as shared memory

# IPC Shared Regions

| Index | Name | Owner | DSP cache enabled? | ARM cache enabled? | Heap? | Base Address | Length (Bytes) | Framework Contents |
|---|---|---|---|---|---|---|---|---|
| 0 | SR_0 | DSP | Y | Y | Y | 8000_0000 | 2097152 | AspMM MessageQ message pool<br>AspMM message info<br>CB GateMP<br>Dec Out Init-Sync GateMP |
| 1 | SR_MSMC | DSP | Y | Y | Y | 0C00_0000 | 524288 | Control structures<br>• Audio Input IO<br>• Input<br>• Decode<br>• CB<br>CB audio frames<br>Audio IO Buff, Phy & Data instances |
| 2 | SR_DDR3 | DSP | Y | Y | Y | 8200_0000 | 16777216 | CB PCM<br>CB metadata |
| 3 | COMMON2_ DDR3 | DSP | N | N | Y | 8100_0000 | 16777216 | Dec Out Init-Sync control structure |

- COMMON2_DDR3: non-cached Shared Region
  - Contains shared decoder component (DEC, PCM, DTSUHDA) control/status structures
  - Created to avoid cache coherency issues. Benchmarking shows negligible performance impact vs. using cache.
  - GateMP used for protected access (simultaneous access from DSP & ARM)

# Application Shared Regions

| Name | Owner | DSP cache enabled? | ARM cache enabled? | Heap? | Base Address | Length (Bytes) | Framework Contents |
|---|---|---|---|---|---|---|---|
| COMMON_DDR3 | DSP | Y | Y | N | 8100_0000 | 6291456 | AST Configuration<br>STD Beta Table (alpha commands)<br>CUS Sigma Table (shortcuts)<br>CUS Beta Table (empty)<br>CUS Phi Table (empty) |

- Created since IPC Shared Region disallows static data allocation
- Access unprotected

# COMMON_DDR3: AST Global Configuration

- AST Config
  - Pointers to Config/Status for Inp, Dec, Str, Enc & Out stages
  - Config/Status for each stage can be located in different memory
  - ARM/DSP shared Config/Status should be located in Non-Cached Shared Region
  - Str, Enc & Out only used by ASOT, can be moved to ASOT Configuration in unshared memory

| AST Config Member | Shared? | Location | Used By | Protection | Protection Comment |
|---|---|---|---|---|---|
| xInp | Yes | SR_MSMC | ASIT (RW)<br>ASDT (R)<br>ASOT (R) | None | Not required |
| xDec | Yes | COMMON2_ DDR3 | ASIT (RW)<br>ASDT (RW) | Decode Status protected<br><br>Otherwise unprotected | Add protection for unprotected members |
| xStr | No | DSP L2 RAM | ASOT (RW) | - | - |
| xEnc | No | DSP L2 RAM | ASOT (RW) | - | - |
| xOut | No | DSP L2 RAM | ASOT (RW) | - | - |

# COMMON_DDR3: STD Beta & CUS Sigma Tables

- STD Beta Table
  - Pointers to Config/Status for Standard components
  - Tasks (ASIT,ASOT,AFP,AIP,SS,ASDT) access through dedicated ACP handles
  - Used for alpha command processing
  - Access unprotected
    - Table entries only written during system initialization
    - ASIT/ASDT system initialization sync'd
      - Ensures these threads don't perform simultaneously write to Table
      - DSP & ARM cache coherency maintained
    - ASDT/ASOT system initialization not sync'd, simultaneous writes to Table possible
      - Beta Units written by ASDT/ASOT are different
      - ARM doesn't need access to Beta Units written by ASOT
    - System initialization for other DSP Tasks sync'd by Task priority
      - ARM doesn't need access to Beta Units written by other Tasks
    - Only read access after system initialization

- CUS Sigma Table
  - Pointers to Custom shortcuts, e.g. at-boot, IO
  - Shortcut: alpha code sequence stored in processor memory
  - Access unprotected
    - Only DSP write access in SYSINIT Task during system initialization
    - Only DSP read access in ASIT/ASOT after system initialization

# COMMON_DDR3: Other Framework Contents

- CUS Beta Table: (empty)
  - Pointers to Config/Status for all Custom components
  - Tasks (ASIT,ASOT,AFP,AIP,SS,ASDT) access through dedicated ACP handles
  - Used for custom alpha command processing

- CUS Phi Table: (empty)
  - Pointers to Custom target-side functions

# STD Beta & CUS Sigma Tables

| Beta Unit | IALG_Status * | Memory Region |
|---|---|---|
| STD_BETA_BETATABLE | 0x80200024 | COMMON_DDR3 |
| STD_BETA_PHITABLE | 0x868E73D8 | CORE0_DDR3 |
| STD_BETA_SIGMATABLE | 0x868E5110 | CORE0_DDR3 |
| STD_BETA_IDENTITY | 0x868C4770 | CORE0_DDR3 |
| STD_BETA_UART | 0x00861D1C | L2SRAM |
| STD_BETA_SYSIDL | 0x868E4368 | CORE0_DDR3 |
| STD_BETA_SYSINT | 0x86913DE0 | CORE0_DDR3 |
| **STD_BETA_IB** | **0x0C0000B0** | **SR_MSMC** |
| STD_BETA_OB | 0x008007F4 | L2SRAM |
| **STD_BETA_DECODE** | **0x81000090** | **COMMON2_DDR3** |
| STD_BETA_ENCODE | 0x00800558 | L2SRAM |
| STD_BETA_VOLUME | 0x00800618 | L2SRAM |
| STD_BETA_PCE | 0x00821AF8 | L2SRAM |
| **STD_BETA_PCM** | **0x81000C00** | **COMMON2_DDR3** |
| **STD_BETA_PCM2** | **0x81000B80** | **COMMON2_DDR3** |
| **STD_BETA_DECOPCB** | **0x0C05A8F0** | **SR_MSMC** |
| **STD_BETA_DTSUHDA** | **0x81000C80** | **COMMON2_DDR3** |
| STD_BETA_DTSUHDB | 0x8605A7F8 | CORE0_DDR3 |

| Sigma | Shortcut | ACP_Unit * | Memory Region |
|---|---|---|---|
| 2 | cus_atboot_s_patch | 0x868C21B8 | CORE0_DDR3 |
| 32 | execPAIInNone | 0x868C2948 | CORE0_DDR3 |
| 33 | execPAIInHDMIStereo | 0x868C1D50 | CORE0_DDR3 |
| 34 | execPAIInHDMI | 0x868C1AB8 | CORE0_DDR3 |
| 35 | execPAIInDigital | 0x868C18C0 | CORE0_DDR3 |
| 36 | execPAIInAnalog | 0x868C1E58 | CORE0_DDR3 |
| 48 | execPAIOutNone | 0x868C3898 | CORE0_DDR3 |
| 49 | execPAIOutAnalog | 0x868C2498 | CORE0_DDR3 |
| 50 | execPAIOutAnalogSlave | 0x868C24C8 | CORE0_DDR3 |
| 51 | execPAIOutAnalog12Ch | 0x868C24F8 | CORE0_DDR3 |
| 52 | execPAIOutAnalog16Ch | 0x868C2528 | CORE0_DDR3 |

- ARM&DSP shared status in SRs
  - Access **protected** w/ GateMP
  - Access **unprotected**
- All shortcuts execute from DSP

# Shared Info – Audio Input

- Input Control and Ring Buffer in IPC SR
  - Region / Heap : SR_MSMC / srHeapMsmc
  - Alloc Location : ASIT initialization

- On Audio Input McASP EDMA complete, following steps are taken in ASIT during active decoding:
  1. For Input data update:
     - Cache invalidate data (necessary since DSP & EDMA not cache coherent)
     - Update data (e.g. byte swap operation)
     - Cache write back data
  2. Execute auto detection
  3. Write current Input Buffer Configuration to IB Queue
  4. Input Control update and cache write back
  5. Read least recent Input Buffer Configuration from IB Queue
  6. Decode Info message sent to ARM indicating least recent input data ready for decode

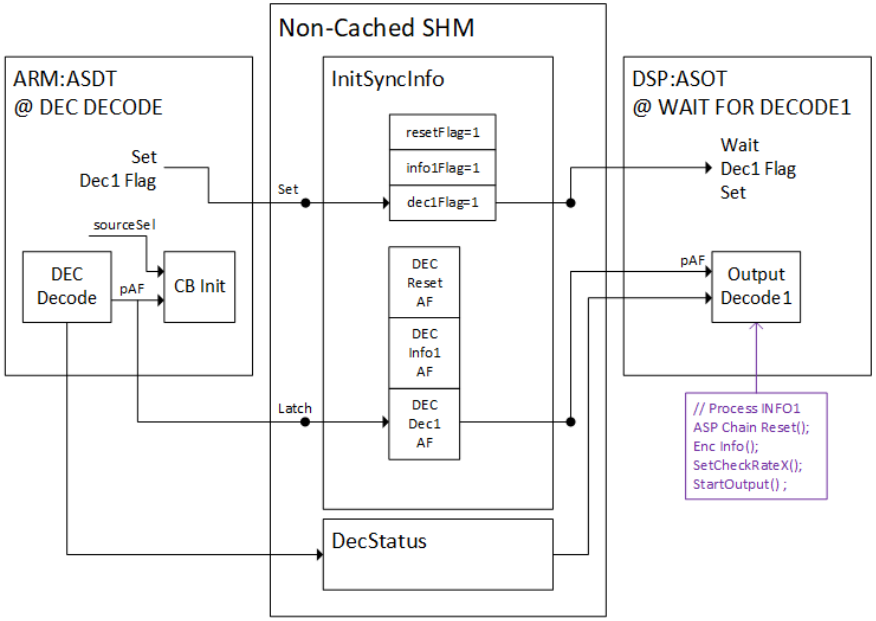
- Access unprotected
  - Input Control & Buffer Configuration: read occurs in ASDT immediately after Decode Info message received & no writes occur in ASDT. Buffer Configuration is from latched info in queue.
  - For Input data: unnecessary since DSP / ARM don't simultaneously access same Input data frame

# Shared Info – ASP Message Manager (AspMM)

- AspMM provides Decode & Decode ACK messages between ARM & DSP using IPC MessageQ
- DSP is message master
  - Creates & owns master MessageQ
  - Creates heap in IPC SR_0. Allocates message pool & message save info in heap.
  - DSP:ASIT TX Decode messages to ARM:ASDT
- ARM is message slave
  - Creates & owns slave MessageQ
  - ARM:ASDT RX Decode messages from DSP:ASIT, TX Decode ACK messages to DSP:ASIT
- Message contents
  - Sender Processor ID: DSP and ARM have distinct ID
  - Message ID: DSP generates ID for each TX message, ARM sets ACK bit in ID for ACK message
  - Command (/ACK): Decode command for Slave, Decode Acknowledge for Master
  - Payload (byte buffer)
- Command (/ACK) set
  - Initialize Slave: START, EXIT
  - Mount Decoder: DEC_SOURCE_SELECT
  - Decode Start / Stop: DEC_ACTIVATE, DEC_RESET, DEC_DEACTIVATE
  - Decode frame: DEC_INFO, DEC_DECODE
  - Decode Control: DEC_CONTROL
- Slave generates ACK for all RX Decode Command messages, Master checks integrity of Rx Decode ACK messages.
  - Processor ID is Slave ID
  - Message ID matches saved message ID for least recently sent Decode Command message
  - ACK bit is set in Message ID
  - Rx Command ACK matches ACK for least recently sent Decode Command message
- Master RX message API allows different types of synchronization
  - Event, immediate. RX message API called after sync Event received.
  - Event, pend forever. RX message API called before sync Event received, pend forever in location of call.
  - Semaphore, pend forever. RX message API called before sync event received, pend in location of call.

# Shared Info – Decode Output Init-Sync (DecOutIS)

- DecOutIS synchronizes ARM:ASDT (Decode) & DSP:ASOT (Output) at start of decode
- Shared IS info in IPC SR
  - Region / Heap: COMMON2_DDR3 / srHeapNonCacheDdr3
  - Alloc: ASIT initialization
- Three IS stages: Reset, Info1, Dec1
  - ASDT shares Decode AF output with ASOT, sets flag indicating AF ready
  - ASOT SM polls for AF ready flags, obtains Decode AF output for initializing Output processing
  - ASDT resets flags during Decode Deactivate
- Access protected w/ GateMP



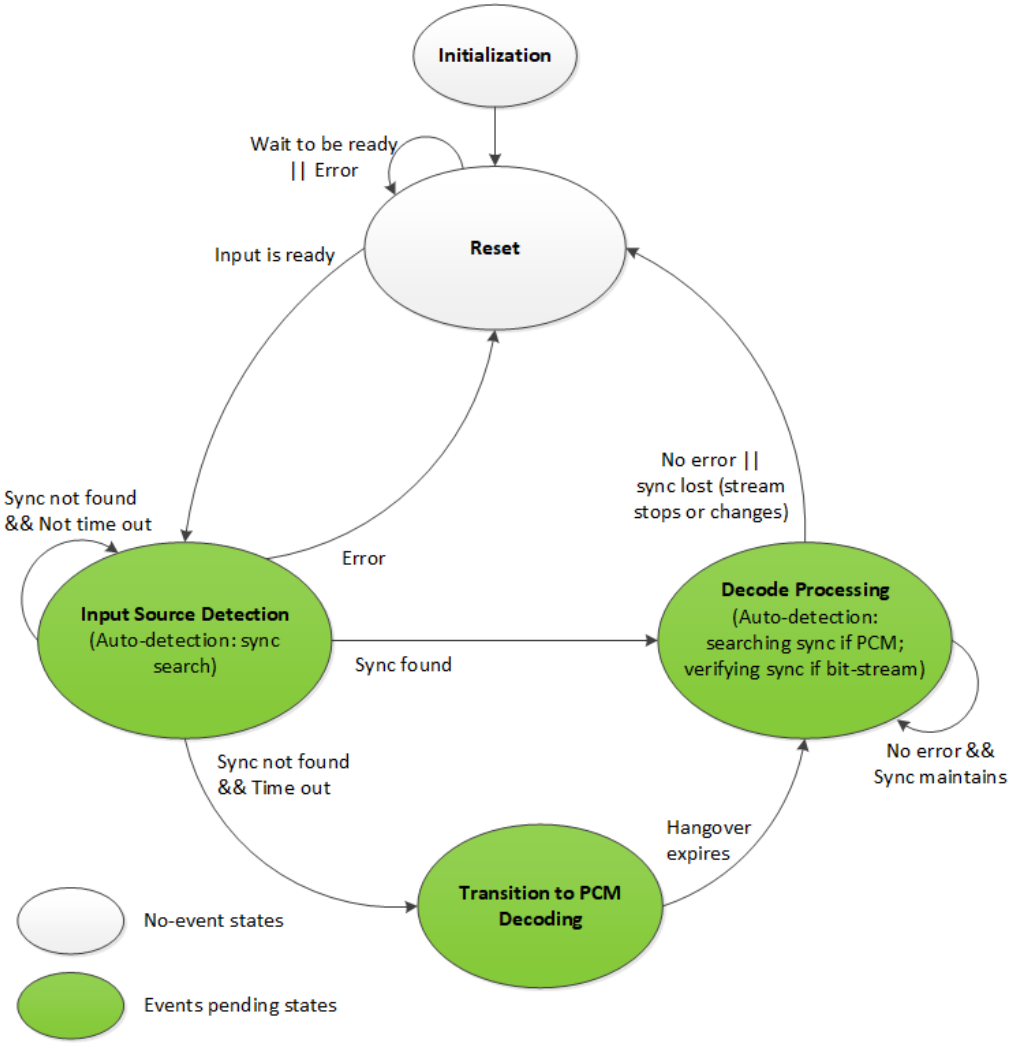



Texas Instruments

# Shared Info – Decode Output Circular Buffer (CB)

- Shares ARM decoder output (PCM & metadata) with DSP output processing
  - ARM:ASDT writes decoder output to CB at Decode (input frame) rate
  - DSP:ASOT reads from CB at Output rate
  - Decode and Output rates can be different, e.g. ASDT 20 msec. & ASOT 5.33 msec.
- CB separated into AF CB & PCM/metadata CB. For AF write to CB:
  - AF info (sampling rate, CC stream, etc.) written to entry in AF CB
  - AF PCM is written to PCM CB, AF metadata is written to metadata CB
  - For DTS:X: 16 channels audio transferred, metadata packed in LSB PCM samples even if no audio present in channel
- ASDT PCM writes are multiples of 256 samples, ASOT PCM reads are 256 samples. Once all PCM samples for AF read from PCM CB, CB moves to next available AF & associated PCM samples.
- CB size
  - PCM size for worst-case Dolby THD decoder delay
  - Metadata size for worst-case THD decoder delay & metadata payload size
- CB Nominal Delay (ND)
  - At start of decoding for stream, ND determined for stream type & sampling frequency
  - CB reads held until ND satisfied (CB filled with samples before ASOT read can occur)
  - ND avoids CB underflow / overflow for properly formatted streams
- CB enters drain state when stream ends and no more decode samples available
  - ASOT continues CB reads until CB empty
  - One CB empty, empty status reported to ASOT. CB enters inactive state & ASOT SM starts polling DecOutIS flags.
- CB collects various stats for use by application, e.g. AF UND/OVR, PCM UND/OVR
- Access protected w/ GateMP

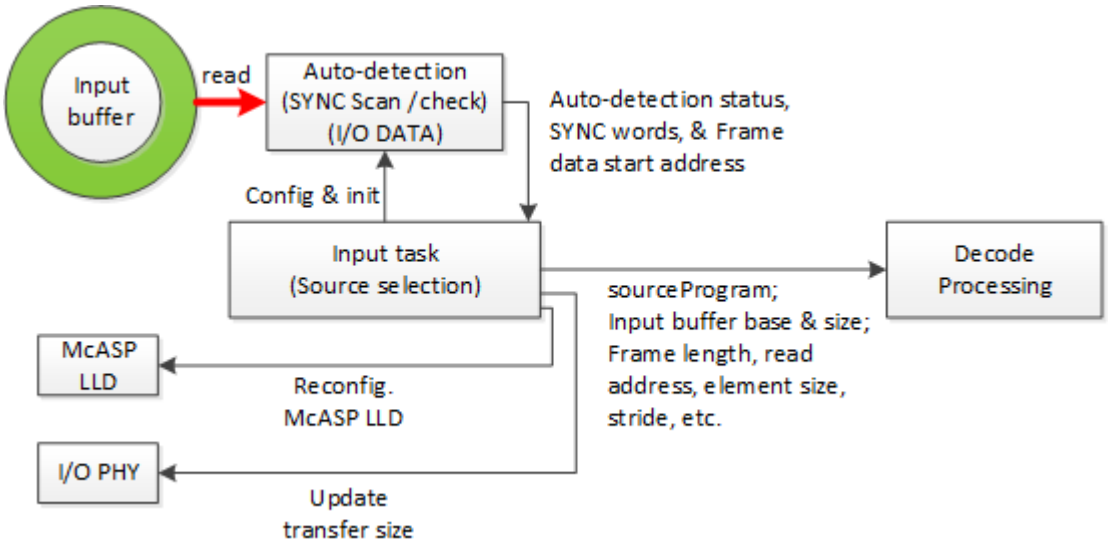
# ASIT State Machine






Texas Instruments

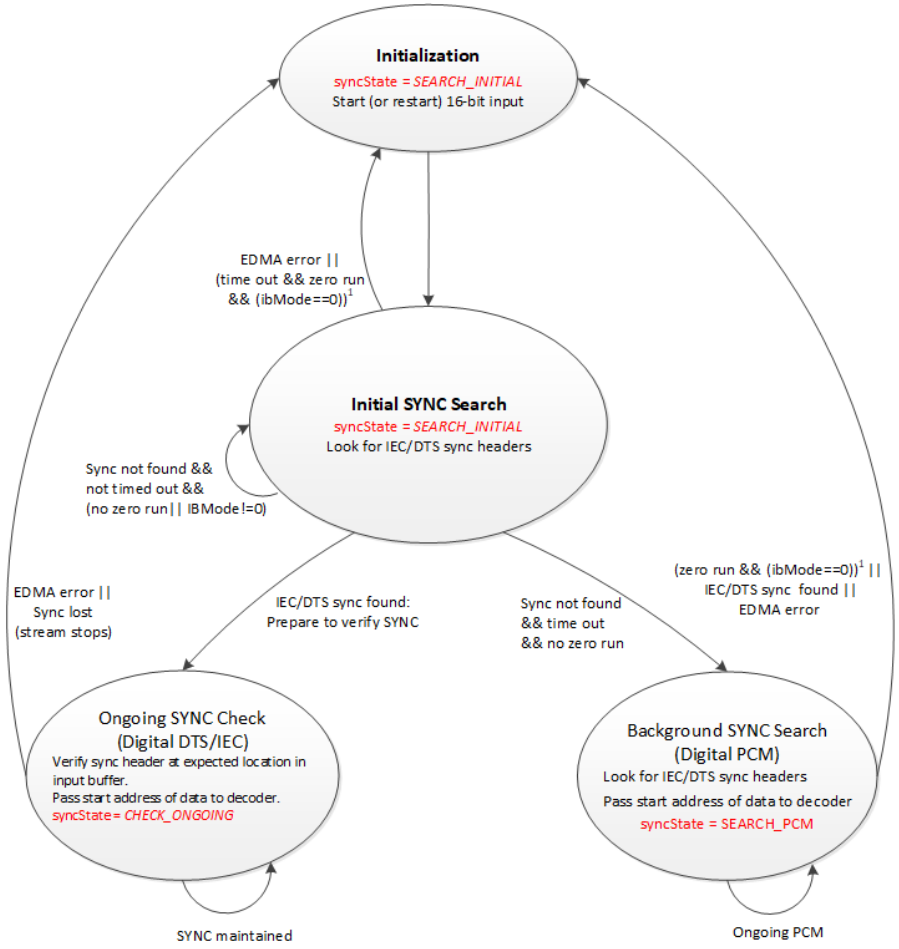
# ASIT Auto-Detection System Level Diagram






Texas Instruments

# ASIT Auto-Detection State Machine

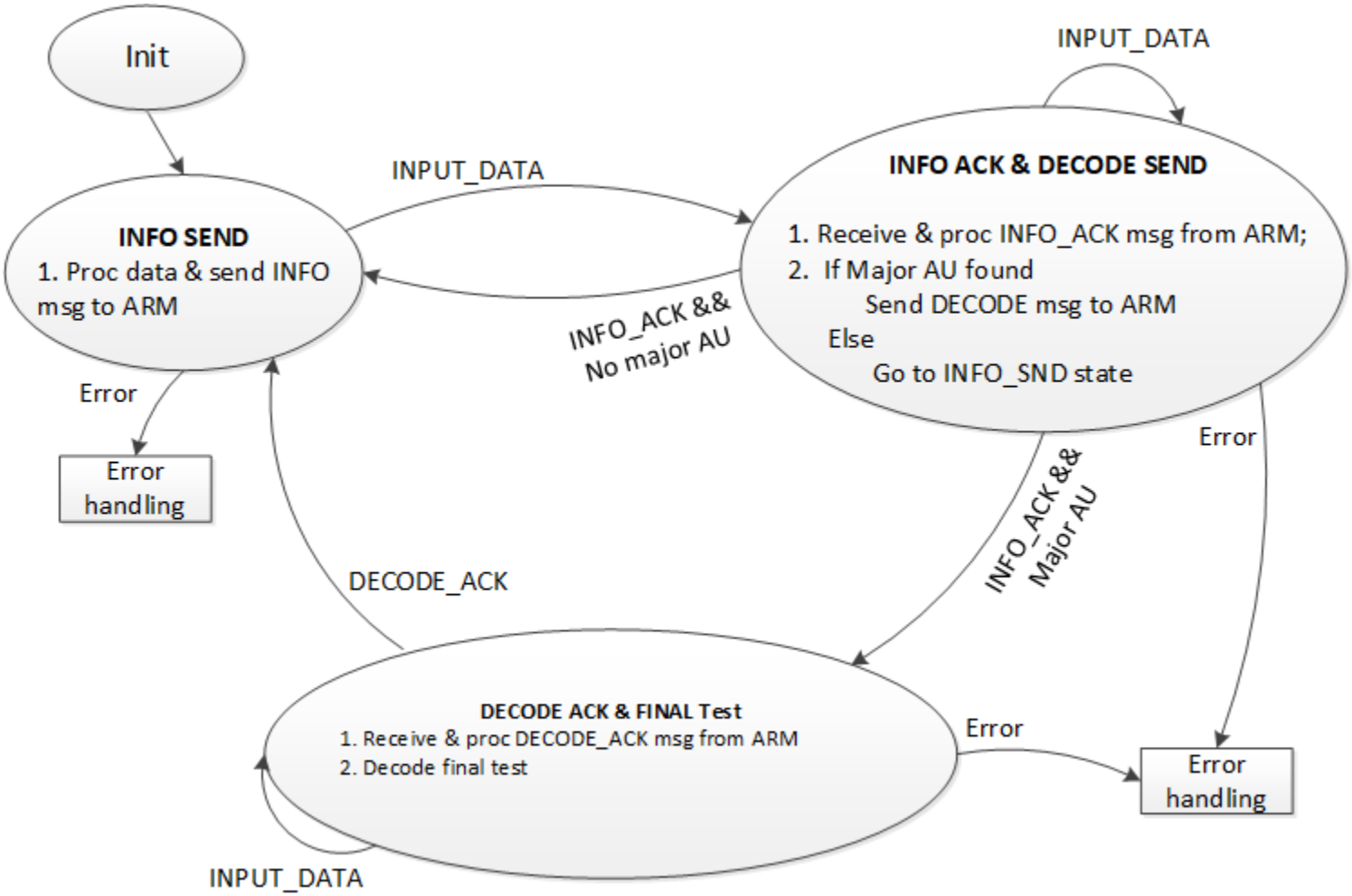



**Note 1: when IBMode != 0, zero run does not impact state transition**

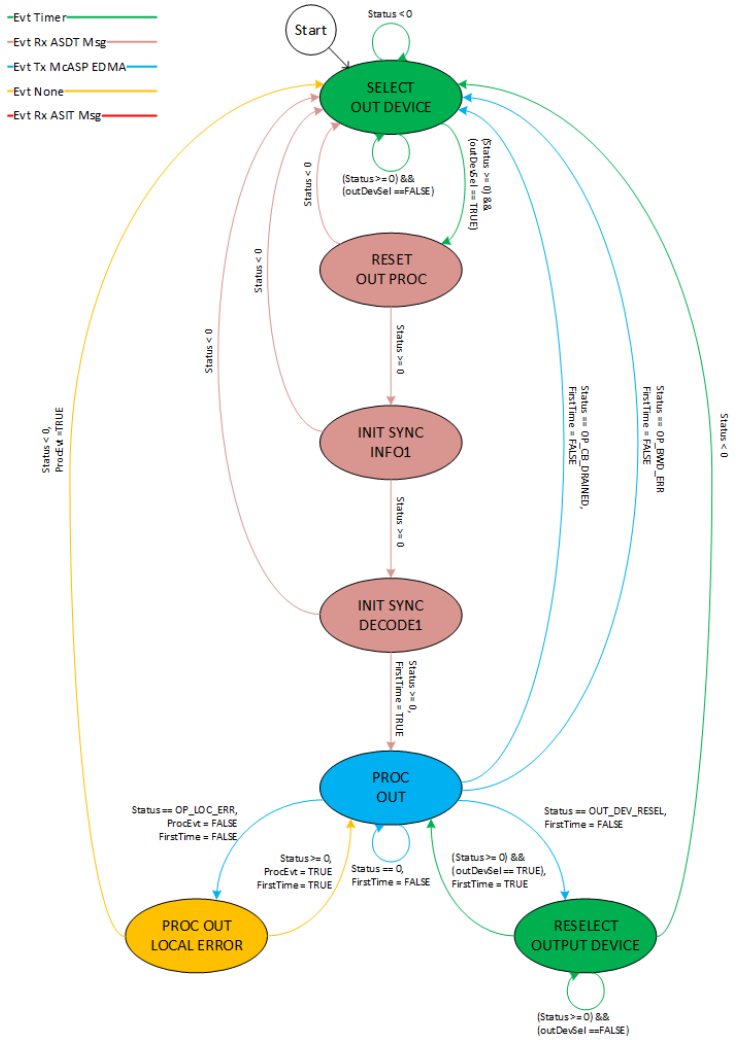


Texas Instruments

# ASIT Decode State Machine



# ASOT State Machine






Texas Instruments

# Backup

**Framework Overview**



Texas Instruments

# PRSDK Drivers / Components

| Driver / Component | Usage in System | Member Of | PRSDK 5.1 Location |
|---|---|---|---|
| Board | EVM initialization | PDK DRV | <prsdk_inst>/pdk_k2g_1_0_11/drv/board |
| McASP LLD | Audio IO | PDK DRV | <prsdk_inst>/pdk_k2g_1_0_11/drv/mcasp |
| UART LLD | Control IO | PDK DRV | <prsdk_inst>/pdk_k2g_1_0_11/drv/uart |
| EDMA3 LLD | Control IO | PRSDK | <prsdk_inst>/edma3_lld_2_12_05_30C |
| IPC | ARM/DSP communications | PRSDK | <prsdk_inst>/ipc_3_50_02_02 |
| AUD | Audio IO (external HW config) | PDK AddOn | <prsdk_inst>/pdk_k2g_1_0_11/addon/aud |
| CSL | Internal LLD | PDK | <prsdk_inst>/pdk_k2g_1_0_11/csl |
| OSAL | Internal LLD | PDK | <prsdk_inst>/pdk_k2g_1_0_11/osal |
| DSPLIB | ASP libraries | PRSDK | <prsdk_inst>/dsplib_c66x_3_4_0_0 |
| MATHLIB | ASP libraries | PRSDK | <prsdk_inst>/mathlib_c66x_3_1_1_0 |
| SPI LLD | Control IO | PDK DRV | <prsdk_inst>/pdk_k2g_1_0_11/drv/spi |
| I2C LLD | Control IO | PDK DRV | <prsdk_inst>/pdk_k2g_1_0_11/drv/i2c |

# PAF Libraries – DEC, ASP, ENC

| Library | Core | Type | Description | PAF Location |
|---|---|---|---|---|
| PCM1 | ARM | DEC | PCM Decoder | pa/dec/pcm1 |
| SNG1 | ARM | DEC | Signal Generator | FD package |
| BM2 | DSP | ASP | Bass Management | FD package |
| DEL3 | DSP | ASP | Speaker Location Delay | pa/asp/del3 |
| DEM | DSP | ASP | De-Emphasis | FD package |
| FIL | DSP | ASP | Filter Library | pa/asp/fil |
| GEQ3 | DSP | ASP | Graphic Equalizer | FD package |
| ML0 | DSP | ASP | MIPS Load | pa/asp/ml0 |
| SRC4 | DSP | ASP | Sample Rate Converter | pa/asp/src4 |
| PCE2 | DSP | ENC | PCM Encoder (VOL,DEL,OUT) | pa/enc/pce2 |

# PAF Libraries – Control, I/O, Utilities

| Library | Core | Type | Description | PAF Location |
|---|---|---|---|---|
| ACP | DSP, ARM | Control | Alpha Code Processor | pa/sio/acp1 |
| DCS7 | DSP | Control IO | Control I/O Switch | pa/sio/dcs7 |
| PAFSIO | DSP | Audio IO | IB, OB Interface Layer | pa/sio/paf |
| ASP_STD | DSP, ARM | Util | ASP Common Utilities (CC mask, sample rate, volume) | pa/asp/std |
| COM_ASP | DSP, ARM | Util | ASP Common Control | pa/asp/com |
| COM_DEC | DSP, ARM | Util | DEC Common Utilities (downmix, delay, FFT, FIFO) | pa/dec/com |
| C674X_INTR | ARM | Util | C674x C intrinsics | pa/util/c67x_cintrins |
| MISC | DSP, ARM | Util | Run-Time IP Protection | pa/util/da10x_misc |
| STATUS_OP_ COMMON | DSP, ARM | Util | ARM/DSP Shared Status Access Protection | pa/util/statusop_common |

# SoC Interrupt Routing – Chip-Level Interrupt Controller (CIC)

| System Event No. | Event Source | Event Source Description | Host Interrupt | Host Interrupt Name | Host Interrupt Description | Function | Function Defined | Configured |
|---|---|---|---|---|---|---|---|---|
| 132 | UART_0_UARTINT | UART_0 interrupt | 32 | CIC_0_OUT32 | CIC output 32 host interrupt | UART_v0_hwiIntFxn | UART LLD | DCS7:UART_open() |

# DSP Interrupt Routing – CorePac INTC Event Combiner

| Combined Event | Event No. | Event Source | Event Source Description | Function | Function Defined | Configured |
|---|---|---|---|---|---|---|
| EVT0 | 6 | EDMACC_0_REGION_0_INT | EDMACC_0 region 0 DMA completion interrupt | lisrEdma3ComplHandler0 | EDMA3 LLD | FWK:registerEdma3Interrupts() |
| EVT0 | 7 | EDMACC_1_REGION_0_INT | EDMACC_1 region 0 DMA completion interrupt | lisrEdma3ComplHandler0 | EDMA3 LLD | FWK:registerEdma3Interrupts() |
| EVT1 | 48 | CIC_0_OUT32 | CIC output 32 host interrupt | ti_sysbios_family_c66_tci66xx_CpIntc_dispatch_E | SYSBIOS | DCS7:UART_open() |

# DSP Interrupt Routing – CorePac INTC Interrupt Selector

| Interrupt No. | Event No. | Event Source Name | Event Source Description | Function | Function Defined | Configured |
|---|---|---|---|---|---|---|
| 7 | 0 | EVT0 | C66x CorePac internal event | ti_sysbios_family_c64p_EventCombiner_dispatch__E | SYSBIOS | DSP app.cfg |
| 8 | 1 | EVT1 | C66x CorePac internal event | ti_sysbios_family_c64p_EventCombiner_dispatch__E | SYSBIOS | DSP app.cfg |
| 14 | 32 | TIMER_0_INTL | TIMER_0 low interrupt | ti_sysbios_knl_Clock_doTick__I | SYSBIOS | SYSBIOS |
| 5 | | IPC_GR0 | BOOT_CFG inter-processor communication register 0 interrupt | ti_sdo_ipc_family_tci663x_Interrupt_isr__I | IPC | IPC |