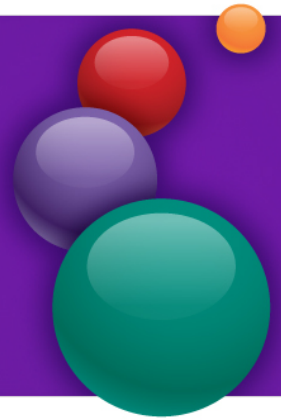
Webinar Topics

- **MIPI Alliance: A Brief Introduction**
  - Peter Lefkin, Managing Director, MIPI Alliance
- **MIPI SoundWire℠ Overview**
  - Pierre Bossart (Intel Corporation), MIPI LML WG Chair

MIPI Alliance:
A Brief Introduction



Peter B. Lefkin
Managing Director

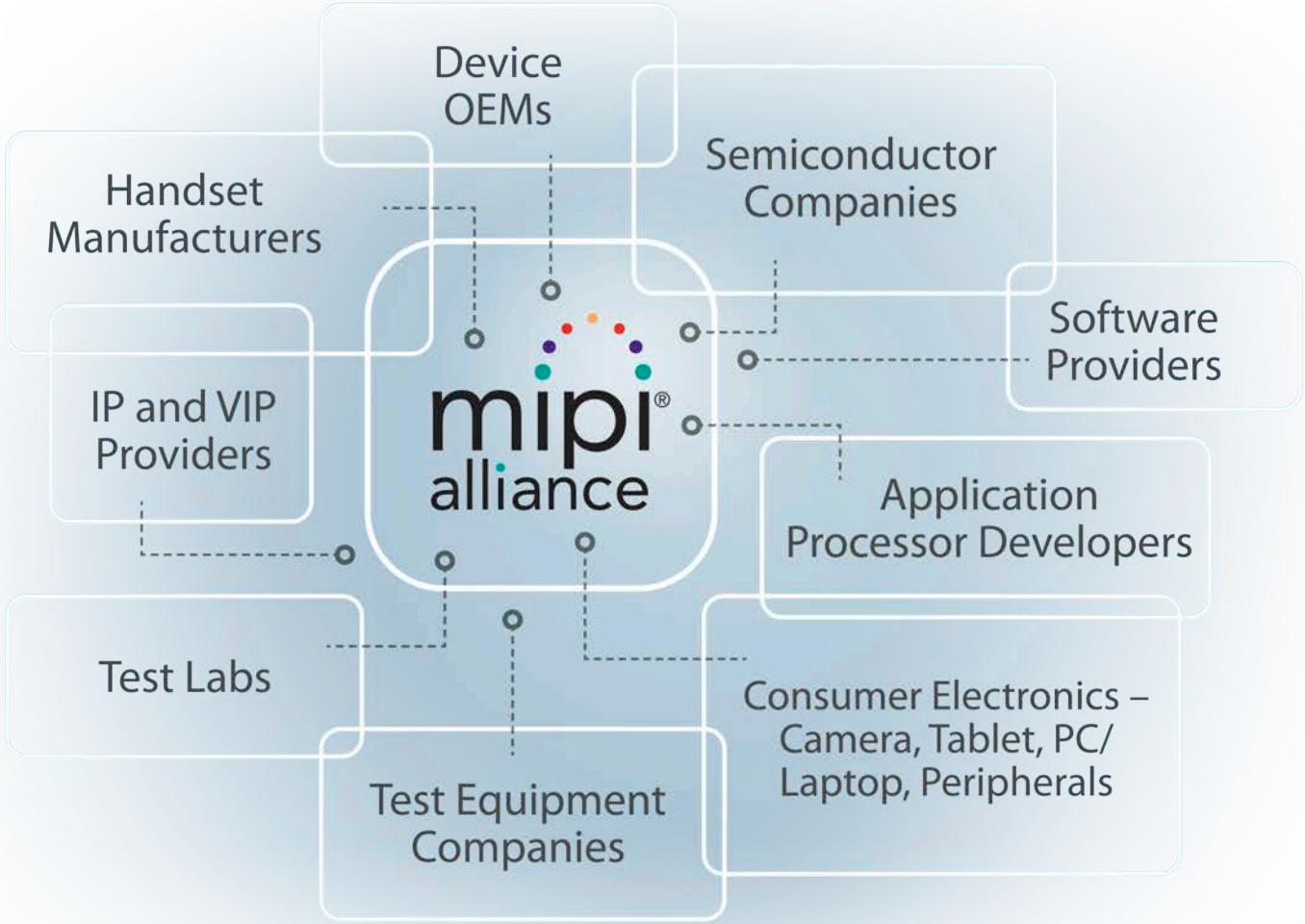
MIPI Alliance – Beginning => Today

Since its inception MIPI Alliance has continued to remain focused and meet the needs of the mobile industry and beyond in the midst of significant change over the 11 year lifespan of the organization. It continues to do so with the development of new technologies and evolution of current specifications.

About MIPI Alliance

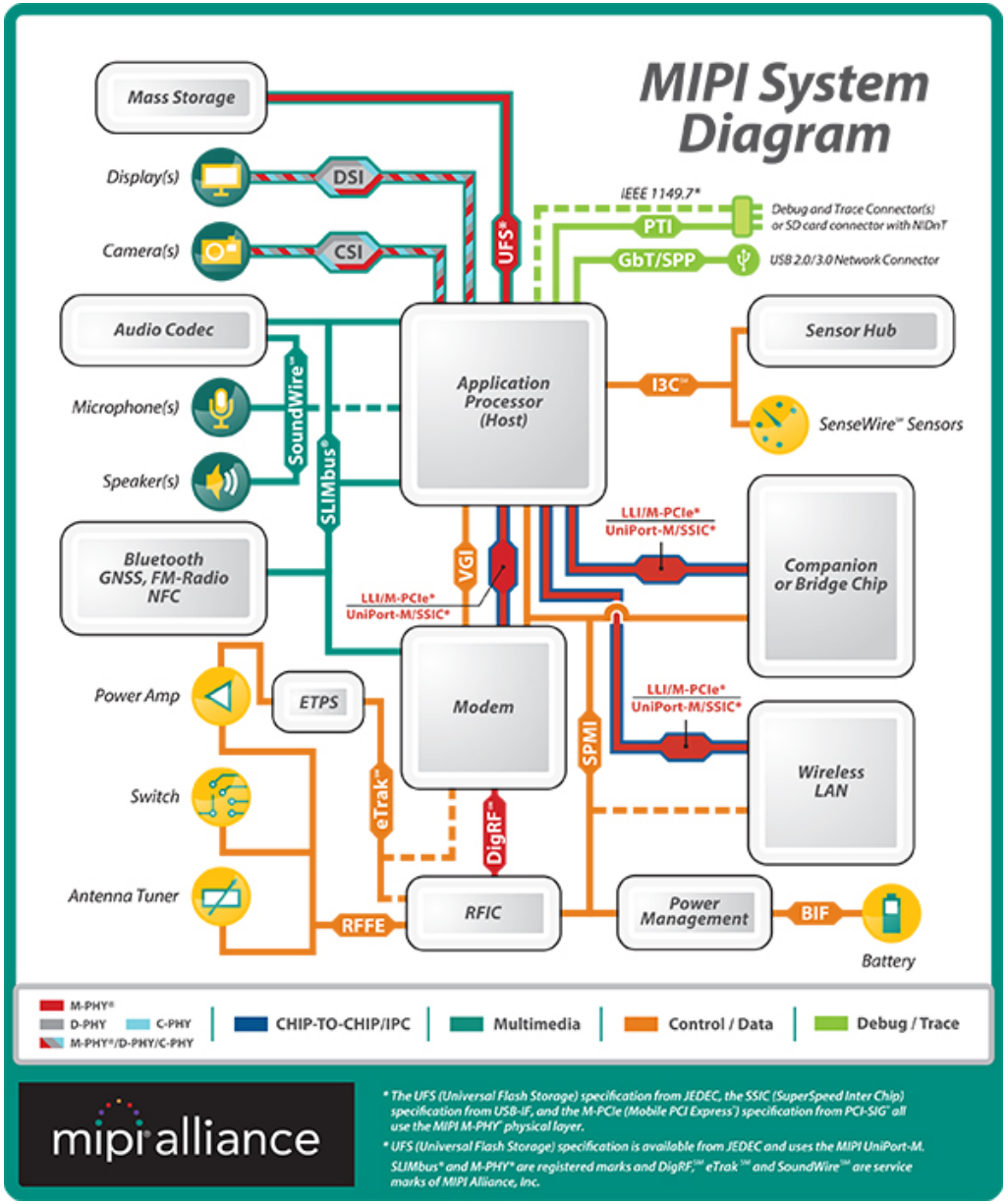
- 263 Members (as of 20 Jan 2015)
- 45+ specifications and supporting docs
- We drive **mobile** and **mobile-influenced interface** technology through the development of hardware and software **specifications**
- We work **globally** and **collaboratively** with other standards bodies to **benefit the mobile ecosystem**

MIPI Alliance Member Ecosystem



Active MIPI Alliance Working Groups

- Analog Control Interface
- Battery Interface
- Camera
- Debug
- Display High Speed Synchronous Interface
- Low Latency Interface
- Low Speed Multipoint Link (New - SoundWire℠)
- Marketing
- PHY (C / D / M)
- Reduced Input Output (RIO) (New)
- RF Front-End
- Sensor / I3C℠ (New)
- Software (New)
- Technical Steering Group
- Test
- UniPro℠



Recent Announcements

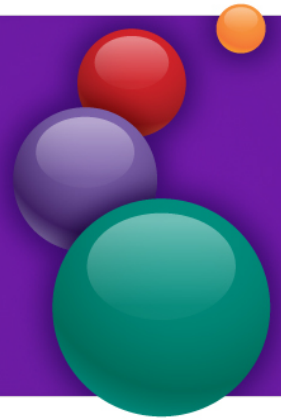
- 05 Nov 2014 -
  MIPI Alliance Introduces Sensor Interface Specification for Mobile, Mobile-Influenced and Embedded-Systems Applications

- 09 Oct 2014 -
  MIPI Alliance Introduces MIPI SoundWire℠, a Comprehensive Audio Interface for Mobile and Mobile-Influenced Devices

The Future of MIPI – Beyond Mobile

- Mobile influences **everything**
- Everything gets faster, smaller and lower power
  - MIPI will continue to evolve specs to take advantage of the evolution of technology in mobile devices

SoundWire[SM] Overview Webinar



Pierre Bossart
Intel Corporation
on behalf of MIPI Alliance LML WG

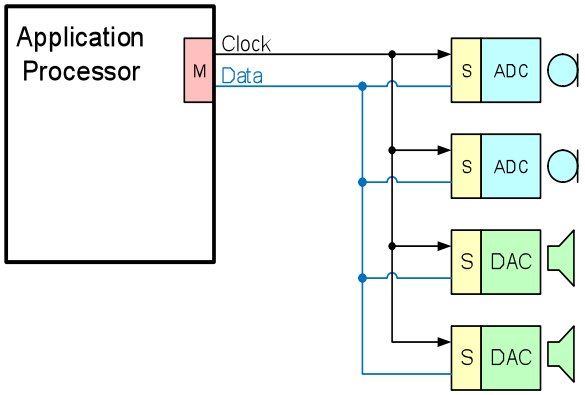
January 21, 2015

SoundWire introduction

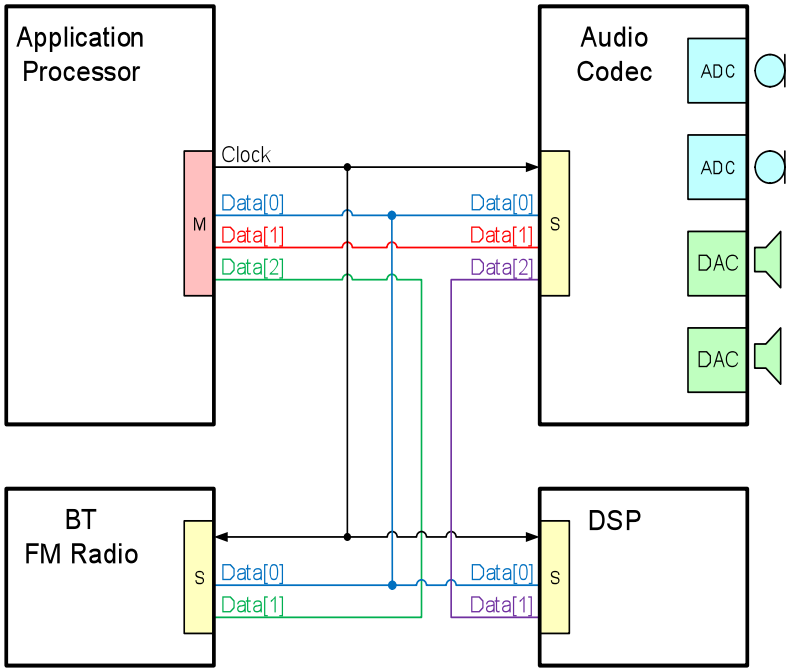
- Timelines
  - Initial discussions in June 2012
  - Contributions from ~16 MIPI contributor companies
  - V09r04 entered 30-day final Voting Draft review
  - V1.0 ratification by MIPI planned for mid-February 2015
- Key features
  - Two-pin dual-data rate multi-drop bus for audio applications (1.2 or 1.8V)
  - Robustness and Scalability (clock and multiple lanes)
  - Low power, low latency, well-bounded (PHY and transport)
  - Support for multiple streams, formats (PCM/PDM/DATA), modes (isoc/async/block)
  - Embedded control/commands
  - In-band interrupts/wakes, support for low-power jack detection
- Benefits:
  - New use cases not possible with existing interfaces (I^2S, SLIMbus®, HDAudio)
  - New system topologies across mobile and mobile-influenced industries
  - Lower gate count allows for integration in cost-sensitive devices

Example topologies

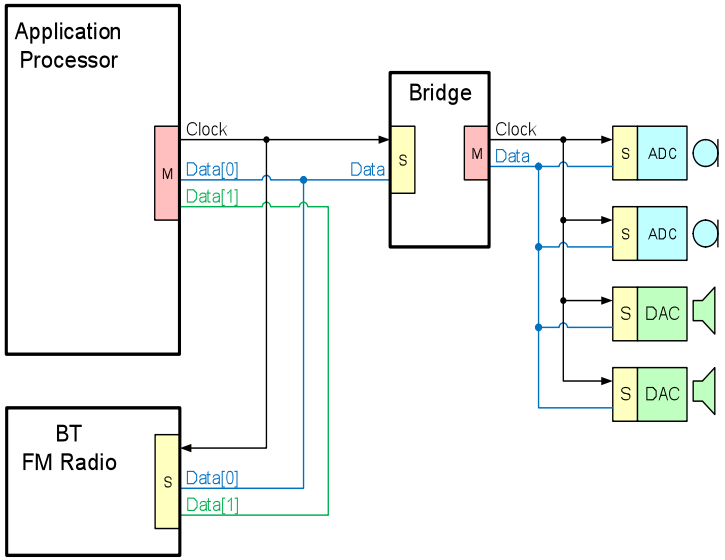
AP direct-attach



Inter-chip link, multi-lane support



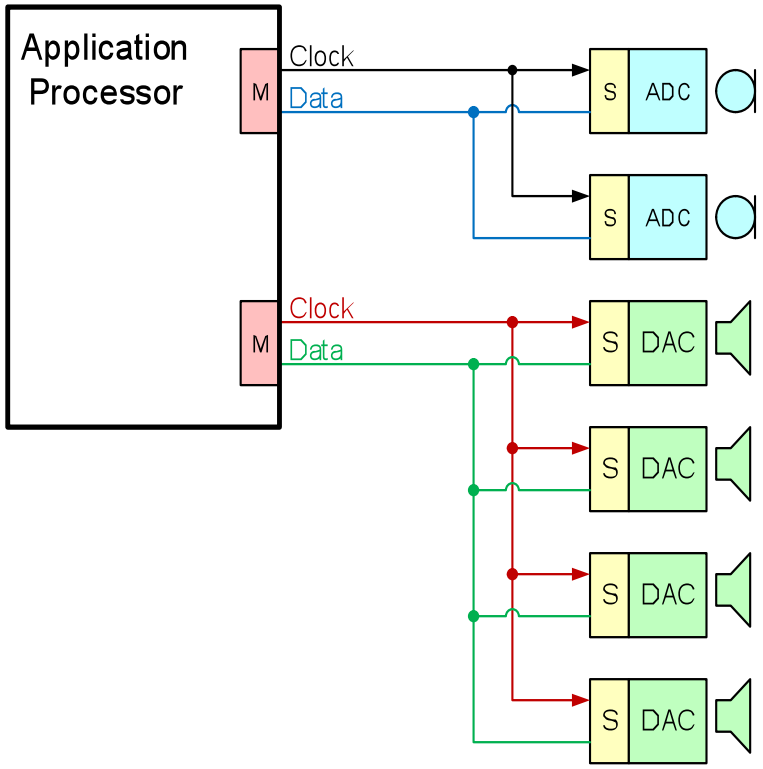
Bridges, inter-chip link



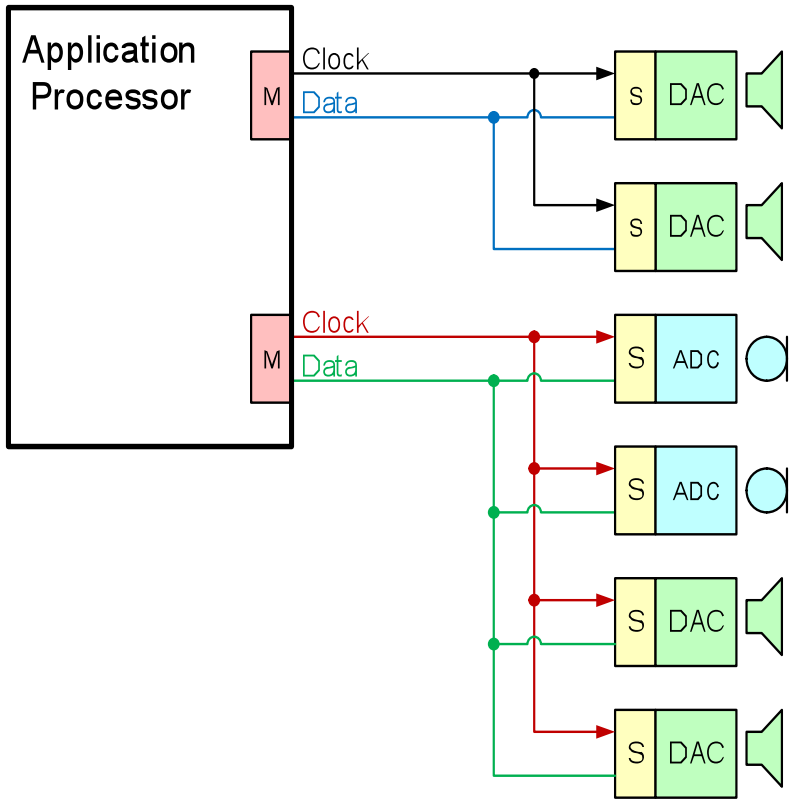
mipi alliance

Example topologies (2)

Functional partitioning



Routing/use case partitioning



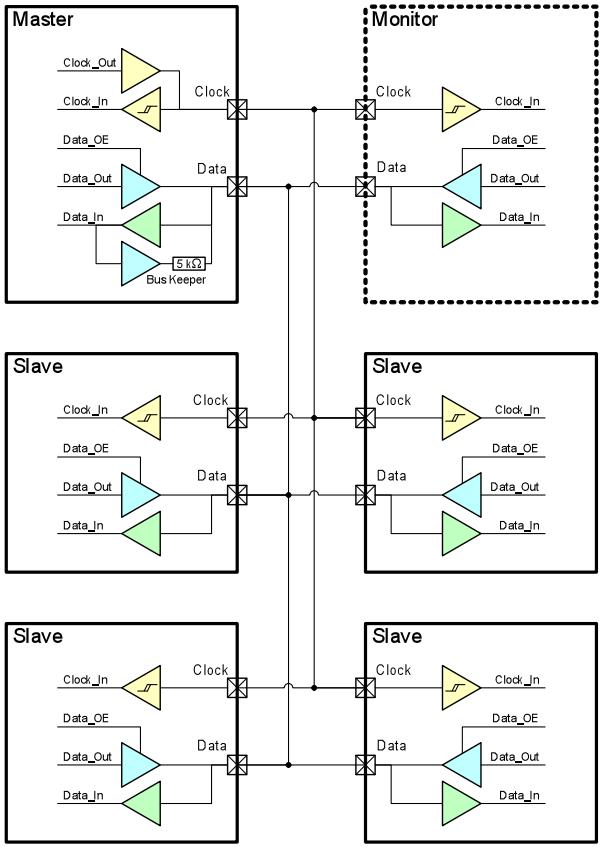
SoundWire Protocol

- Variable-size Frames
  - Reduces overhead
- Two-step synchronization
  - Static 8-bit sync word
  - Dynamic 4-bit word to remove ghost sync words
- Embedded commands
  - Removes need for I^2C/SPI
  - PING: Status, test support
  - READ/WRITE: 8-bit payload
- Bulk protocol to extend command bandwidth
  - Fast device configuration using up to 20 Mbits/s
- Up to 11 Slaves
- Unified Data Transport
  - Up to 14 ports, 1..8 channels
  - PDM, PCM
  - Bit-level interleaving to ensure low-latency
  - Isoc and async modes
  - Dynamic changes in bandwidth allocation without audio glitches
- Reconfigurations through synchronized bank switch
- Interrupt capabilities
  - 32-frame max latency

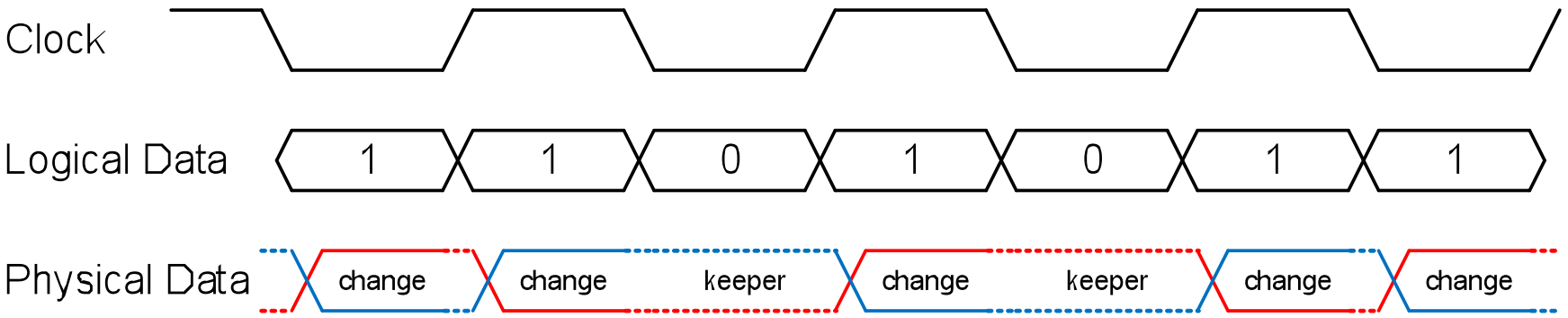
PHY details

Device types

- Master
  - Provides clock and sync pattern on data line
  - Handles all bus management, bit allocation
- Slave
  - Typically audio peripheral (microphone, codec, amplifier)
  - 1..11 Slaves connected to Master over multi-drop bus
  - Ability to signal Interrupt condition
  - Ability to wake-up system
- Monitor
  - Test equipment, in snooping/analyzer mode most of the time
  - Can temporarily take-over bus and issue read/write commands



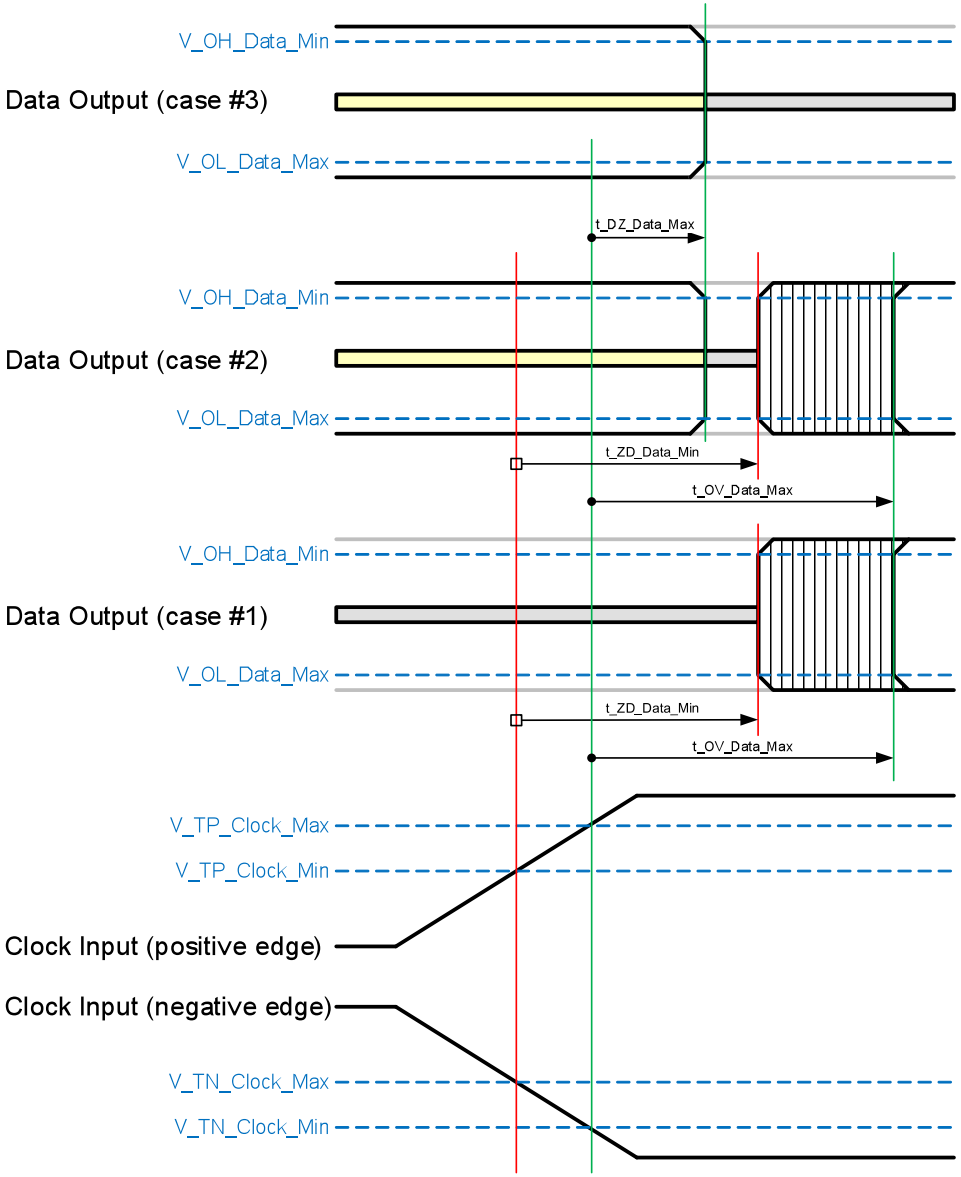
Modified NRZI encoding



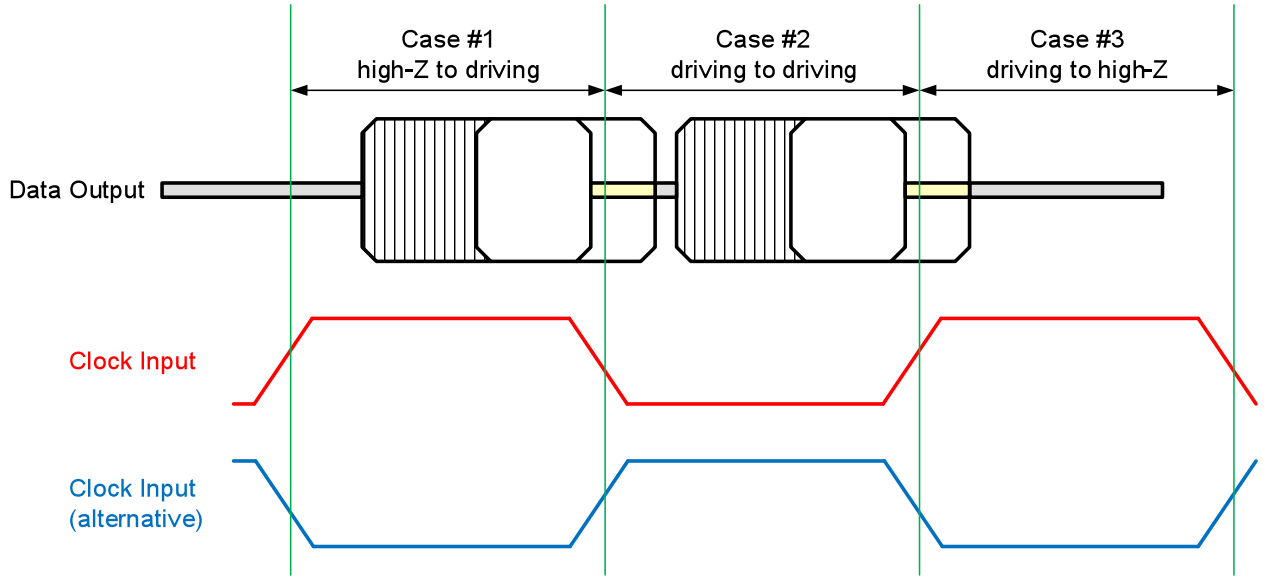
- One bit can be owned by multiple devices.
- Modified-NZRI encoding removes drive conflicts
  - Logic 1 represented as an active change
  - Logic 0 represented as a passive unchanged level, maintained by a bus-keeper.

Data handover (1)

- DDR transmission
- No secondary edge to control handover between adjacent bitSlots
- tDZ_Data_Max: worst case (latest) time to become high-impedance
- tZD_Data_Min: worst-case (earliest) time to drive.
- tDZ_Data_Max < tZD_Data_Min
- Margin included to allow devices detecting the clock edge at different times.



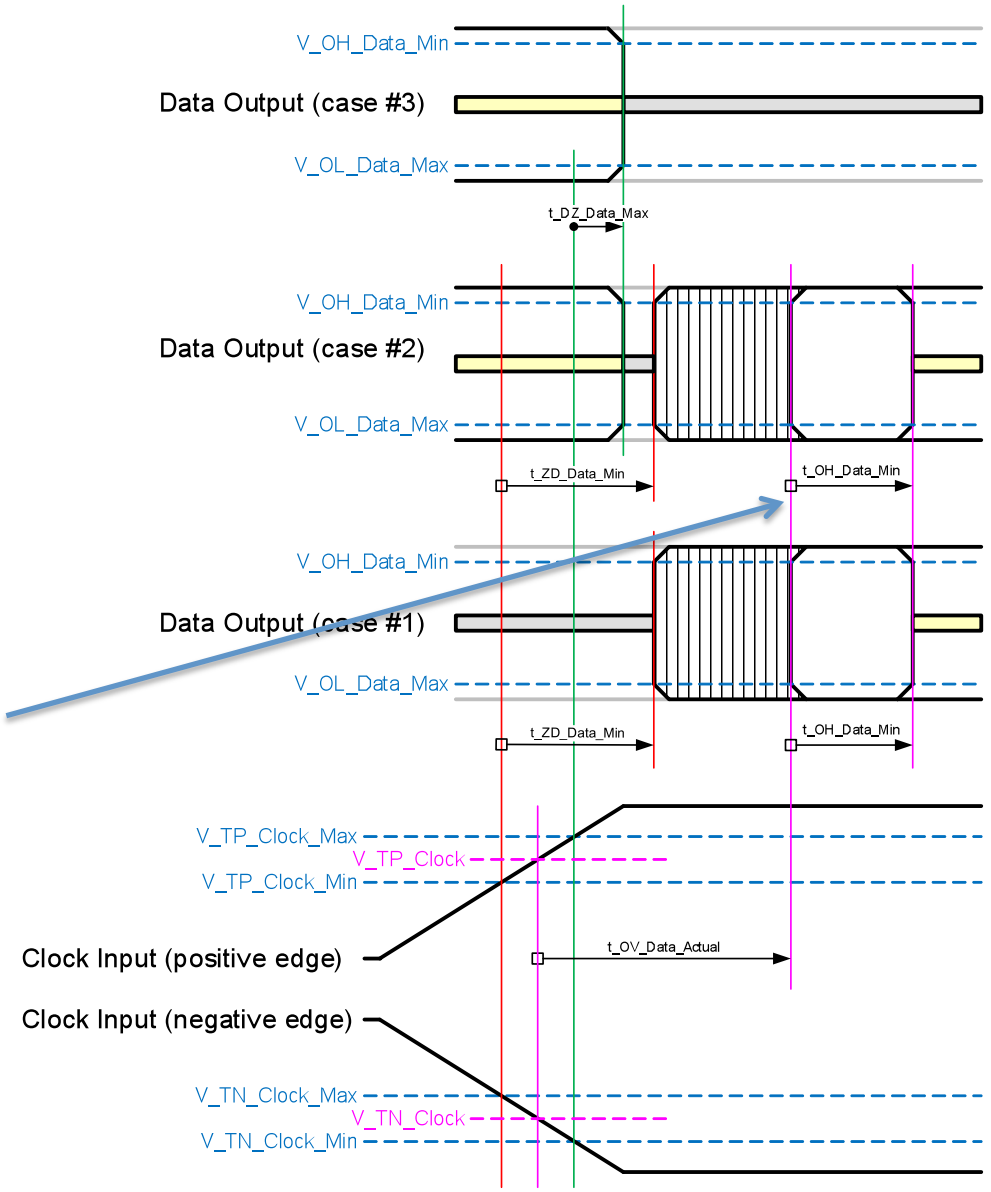
Data handover (2)



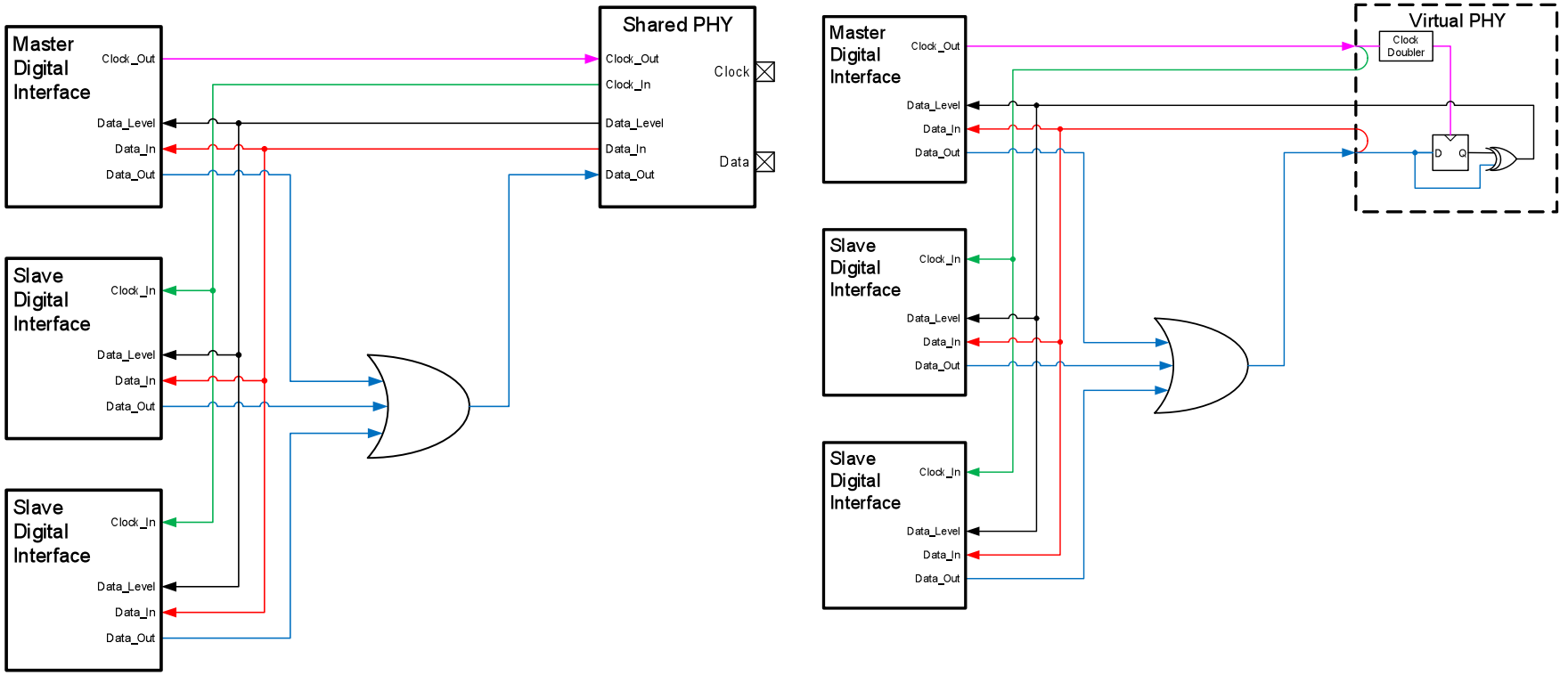
- More than one device might drive a particular BitSlot
- It is required to momentarily tri-state even in a device that is actively driving two adjacent BitSlots.

Data handover (3)

- Turn-off time (tDZ_Data_Max) might be problematic with large-geometry silicon process.
- Explicitly permit a "self-timed" turn-off that occurs before the clock edge that ends a given BitSlot.
- The tOH_Data_Min parameter ensures that the valid signal is presented for long enough to guarantee that a bus-keeper has snapped to the new value, even when it is at the opposite end of the PCB trace to the device driving the data signal.



Shared and virtual PHY



- Shared PHY: module-level integration, pins shared
- Virtual PHY: pins not visible externally, SoundWire used within module.

Clock

- Max bus clock frequency of 13MHz for typical geometries
- Audio transport most efficient with 'natural' clock frequencies, e.g. 9.6, 12, 12.288 MHz.
- Can be faster in specific settings
  - e.g. single Slave close to Master
- Clock quality constrained to meet PHY parameters
- Jitter needs to be limited for best audio quality
- Jitter performance considered a differentiating feature and system design concern
- No requirements on jitter (ppm, ps) in SoundWire spec

High-PHY

- Mechanism to go beyond mandatory timings
- Requires system-level knowledge on integrated components. All components on High-PHY link need to support same requirements.
- Hand-over sequence defined between ‘normal’ and ‘high-PHY’ mode.
- Mode identification using one bit of the static sync word.

PHY Test modes

- Master pins can be configured in several test modes
- External master or test equipment can drive data and clock instead of master and replace the master bus keeper
- ('x' means functional in table)

Mode name	Data bus-keeper	Clock Output	Data Output
Normal	x	x	x
M_DataOff	x	x	Off/high-Z
M_ClockDataOff	x	Off/ high-Z	Off/high-Z
M_AllOff	Off/high-Z	Off/ high-Z	Off/high-Z
M_KeeperOff	Off/high-Z	x	x
M_LowLow	x	Static Low	Static Low
M_LowHigh	x	Static Low	Static High

Frame structure

Frame shapes

- Columns: 2-4-6-8-10-12-14-16
- Rows: 48 to 256. Only selected row values in this range are valid
- Control word: first 48 rows of Col0
- Payload: PCM, PDM, bulk or raw data
- Frame shape determined by command/control bandwidth
- Slaves are required to handle pairwise combinations of Rows, Cols
- Master will typically only use a few combinations with rows and columns scaled by 2^N factors

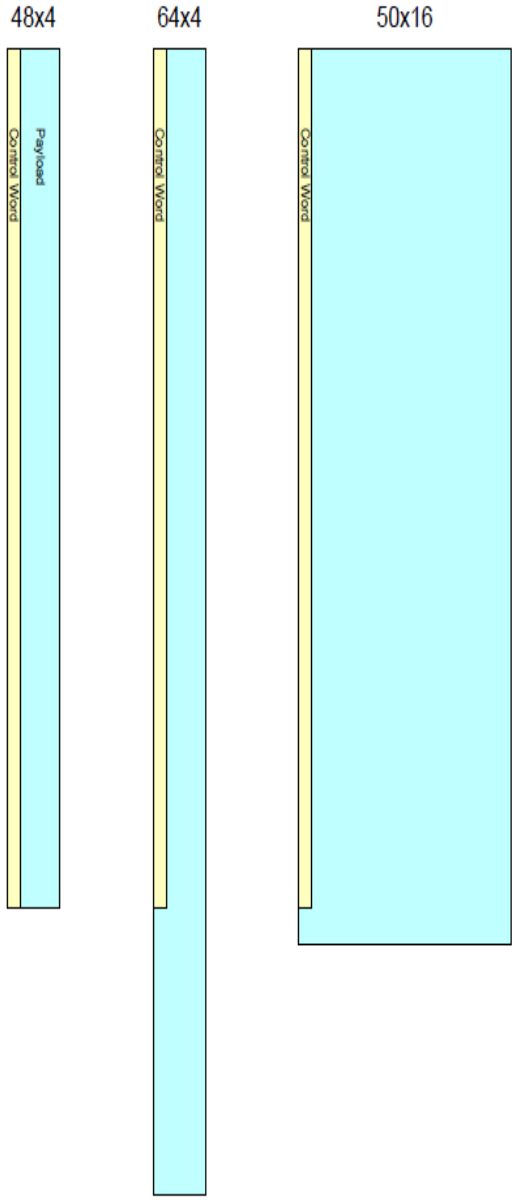
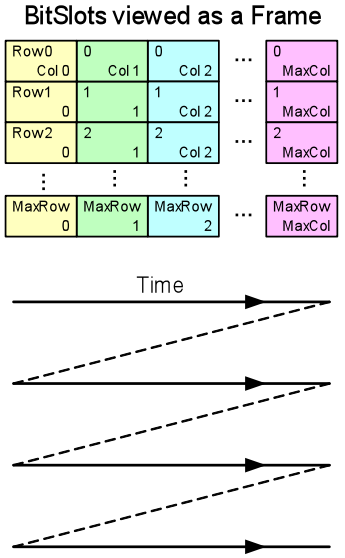
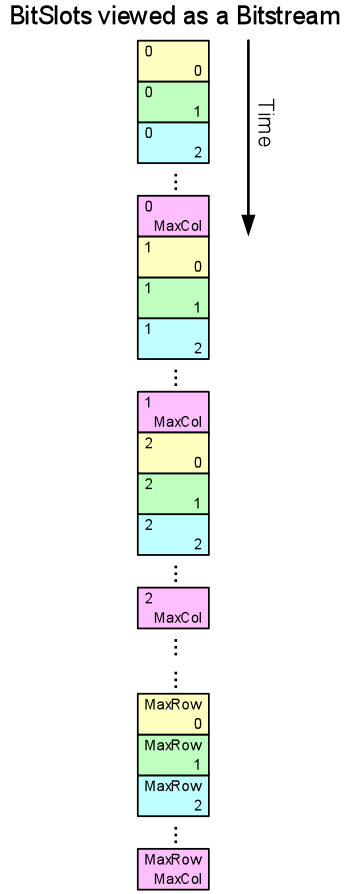


Figure 15 Control Word within Example Frame Shapes

Two-dimensional frame structure

- Serial transmission but frame defined by 2D pattern (MaxRow, MaxCol)
- Low latency streams (1-bit PDM):
  - Samples evenly distributed in time
  - 'vertical stripes' in bit allocation
  - No conflicts with command/ control or PCM
- PCM streams
  - Can use multiple columns and/or multiple rows





48-bit Control Word

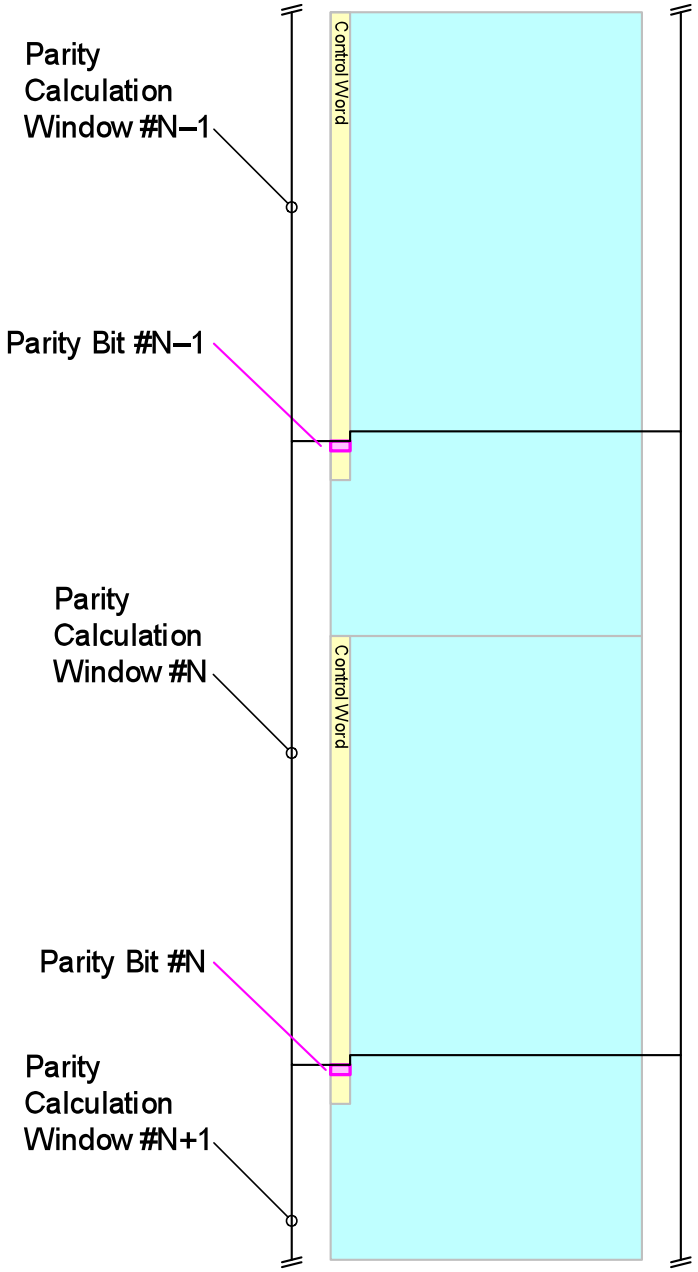
- Interrupt/Wake signal (PREQ)
- Synchronization
  - Static sync word: 8+1 bits to lock on frame boundaries
  - Dynamic sync: 4-bit CRC pattern with 15 frame period
- PING command
  - Slave interrupt and status
    - Not Attached (not present or operational)
    - Attached (synchronized with Master and able to handle commands)
    - Alert (synchronized, at least one Interrupt condition raised)
  - Multi-stream synchronization
  - Bus arbitration with Monitor (BREQ, BREL)
- Read/Write commands
  - 16 bit address, 8 bit payload
  - Address: single Slave, group of Slaves, broadcast.
- Command status
- Parity check

Error handling

- Command Failed condition on
  - Parity Error
  - Bus Clash
- Command_Ignored condition on
  - Non-existent device (programming error)
  - Device not attached (lost sync or power)
  - Register reserved or not implemented
- PHY definition leads to infrequent single-bit errors
  - Command retransmitted or bus reset if errors
  - Payload not suppressed or retransmitted
- No checks for programming errors
- No check if value written to a register makes sense

Parity checks

- Parity bit set by Master or Monitor
- Slaves need to set NAK bit and interrupt condition on parity error
- Parity based on physical level
  - Use the value read on bus, not the value sent to the bus
  - Parity will detect some bus conflicts
- Parity calculation window:
  - BitSlot0[44,1] in previous frame to BitSlot[44,0] in current frame
  - Error may be reported with a 1-frame delay
- Slaves do not compute parity until they have successfully synchronized to master



Slave registers and banks

- Normative registers: 0x0 - 0xFFF
  - ~50% used
- Device-class reserved: 0x1000 - 0x17FF
- Implementation-defined:
  - 0x2000-0xFFFF
  - 0x10000-0x3FFFFFFF additional space accessible with paging registers
- Register writes take effect at end of frame if command successful
- No action if command failed

- Some registers are banked
  - Software prepares next configuration in 'shadow' bank
  - Software write to SCP_FrameCtrl0-1 to switches bank
- All devices switch banks in synchronized manner
- Impact: frame shape changes, channel activation/deactivation, bitSlot allocation changes

Interrupt registers

- Interrupt status stored in SCP_Intstat registers
- Hierarchical representation with cascade
- Optimized for simple devices with up to 4 ports

0x0040	SCP_IntStat_1	$ SCP2 cascade	$ Port 3 cascade	$ Port 2 cascade	$ Port 1 cascade	$ Port 0 cascade	*$ IntStat ImpDef1*	$ IntStat Bus Clash	$ IntStat Parity
0x0040	SCP_IntClear_1	—	—	—	—	—	** IntClear ImpDef1*	* IntClear Bus Clash	* IntClear Parity
0x0041	SCP_IntMask_1	—	—	—	—	—	*IntMask ImpDef1*	IntMask Bus Clash	IntMask Parity
0x0042	SCP_IntStat_2	$ SCP3 cascade	$ Port 10 cascade	$ Port 9 cascade	$ Port 8 cascade	$ Port 7 cascade	$ Port 6 cascade	$ Port 5 cascade	$ Port 4 cascade
0x0043	SCP_IntStat_3	—	—	—	—	$ Port 14 cascade	$ Port 13 cascade	$ Port 12 cascade	$ Port 11 cascade

+0x00	DPN_IntStat	*$ IntStat ImpDef3*	*$ IntStat ImpDef2*	*$ IntStat ImpDef1*	—	—	—	$ IntStat Port Ready	$ IntStat Test Fail
+0x00	DPN_IntClear	** IntClear ImpDef3*	** IntClear ImpDef2*	** IntClear ImpDef1*	—	—	—	* IntClear Port Ready	* IntClear Test Fail
+0x01	DPN_IntMask	*IntMask ImpDef3*	*IntMask ImpDef2*	*IntMask ImpDef1*	—	—	—	IntMask Port Ready	IntMask Test Fail

Bus set-up and programming sequence

Programming model

- SoundWire is a transport protocol mostly
- Audio functionality will be configured through implementation-defined registers
- First devices likely to use same register map as I^2C/SPI devices
- Many similarities with existing audio driver models

Slave enumeration value

- Hard-coded 48-bit unique value
  - SoundWire spec version (to handle future revisions, if any)
  - UniqueID (if there are identical parts on bus)
    - Value set by implementation-defined mechanism such as pin-strapping
    - Set by system integrator, not manufacturer
  - MIPI ManufacturerID
  - PartID
  - Class (not defined yet)
- 48 bits stored as 6 SCP_Device0-5 registers
- Registers read by Master to identify Slave
- Enumeration stops when Slave has non-zero Device Number

Slave startup sequence

- Slave determines the frame format without supervision
- Slave verifies static and dynamic sync pattern for 16 frames
- Slave drives PREQ and/or drives "Attached" status bits for Device Number 0
- Master reads 48-bit enumeration value from Slave
- Master assigns non-zero Device Number (1-11) to Slave
- Catch: multiple Slaves can report as attached at same time, solved by hardware arbitration

- Enumeration procedure:
  - Master reads from Device0, SCP_Device_ID0 register
  - On bus conflict, Slave with highest enumeration value will win arbitration, all other devices back-off
  - Master needs to redo enumeration until there are no Device0 reporting as Attached

Reset levels

- Hard-Reset
  - Power-on or implementation-defined reset
  - Bus Reset: Master drives 4096 Logic1 transitions
  - Device Reset: Master writes Reset bit in SCP_Ctrl
- Soft Reset
  - Slave detects two sync errors (not necessarily successive)
- Hard/Soft Reset difference
  - Slave maintains Interrupt Status register to allow for debug (look at sync loss cause)
- After reset
  - Interrupt masks disabled
  - Device Number lost, re-enumeration required

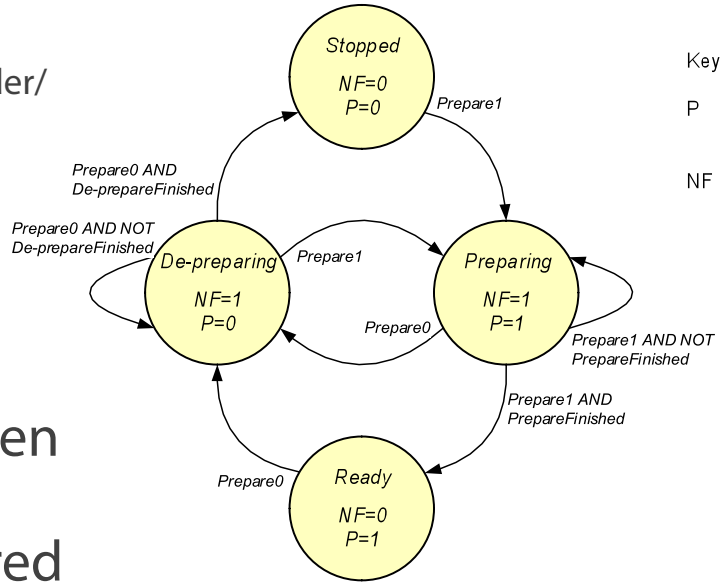
Transport

Data protocols

- Ports definition:
  - Source (generate data on bus),
  - Sink (retrieve data from bus)
- Isochronous 'Normal' mode for regular audio playback
- Asynchronous modes handled with two-bit preamble per sample (RX-Ready, TX-Ready)
- Data only transmitted when both RX-Ready and TX-Ready are set
  - TX-controlled: RX-Ready=1, Source defines when TX-Ready is set
  - RX-Controlled: TX-Ready=1, Sink defines when RX-Ready is set
  - Full-Async: both Source and Sink control rate
- Usages of async modes
  - Slave can generate better audio clock than SoundWire clock
  - 44.1 kHz playback over 48 kHz link
  - Bursty behavior for always listening and voice calls

Data transmission

- Two concepts
  - Prepare: make sure Slave is ready to render/ capture
  - Activate: transport data on bus
- Prepare state machine:
  - Port can be ready immediately
  - Port can require time to be ready
- Software can unmask an interrupt to be notified when ready
- Activate might be configured at any time, but audio might not be valid
- Activation typically done with bank switch to avoid bus conflicts between streams

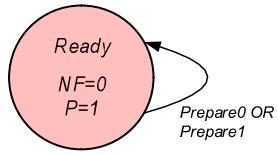


CP_SM

Key to Register bits:

P Value read from Prepare[c] in DPx_PrepareCtrl

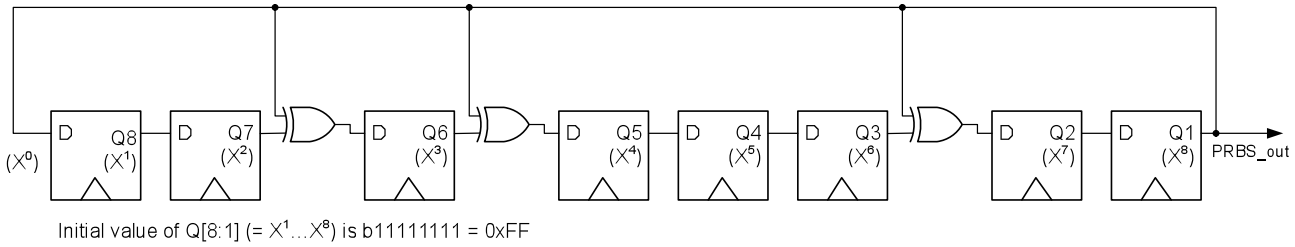
NF Value read from NotFinished[c] in DPx_PrepareStatus



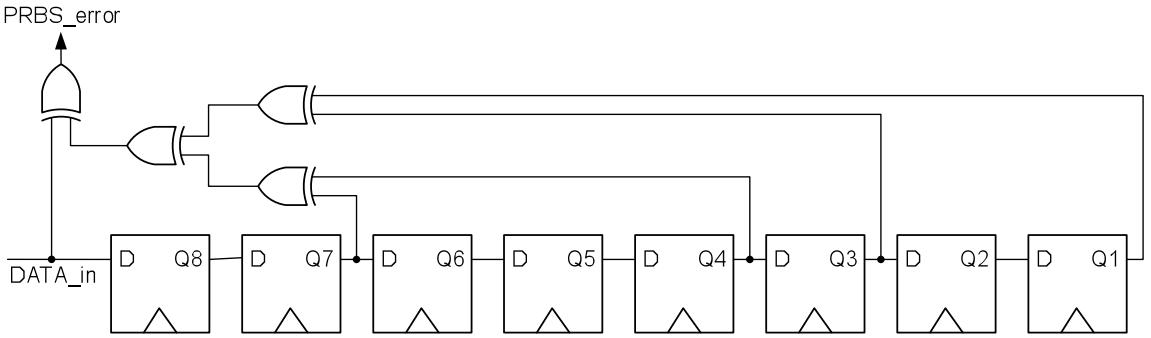
Simplified_CP_SM

Transport test modes

- Mandatory support in each Port
- Static0: helps detect Bus Clash Errors (another port drives in same bitSlots)
- Static1: helps generate Bus Clash Errors
- PRBS: help detect data integrity
  - 8-bit LFSR for PRBS
  - Generates 255-bit maximal length sequence
- Different structure and init value for TX and RX
  - Receiver synchronizes in up to 8 bits
- Interrupt can be generated on error



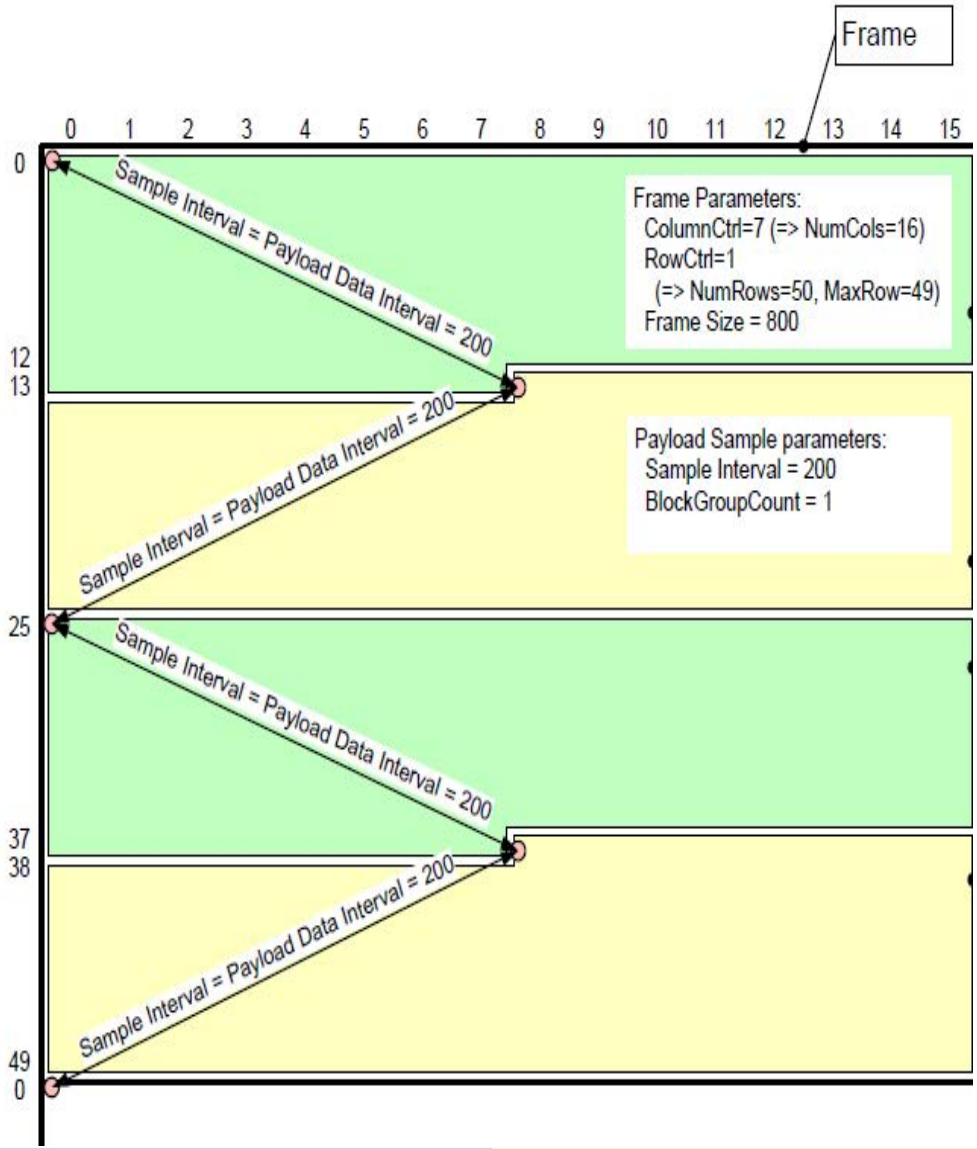
Initial value of Q[8:1] (= X^1…X^8) is b11111111 = 0xFF



Initial value of Q[8:1] is b11010010 = 0xD2

Transport (1): Sample Interval definition

- Sample Events: instant when data is captured/rendered
  - word clock / frame sync periodic event
- Sample Interval: Number of bits between successive Samples
- Sample Interval value is updated when Frame Shape or frequency changes
  - Not necessarily multiple of row size
  - Not dependent on # of channels
- Audio data does not need to be placed in specific location within Sample Interval
- Bit allocation can change dynamically without impact on capture/rendering



Transport (2): Sub-Frame definition

- Vertical' partition of Frame
- Can also be viewed as temporal aperture
- Registers

- HStart
- HStop
- Notional:

Hwidth = HStop-Hstart

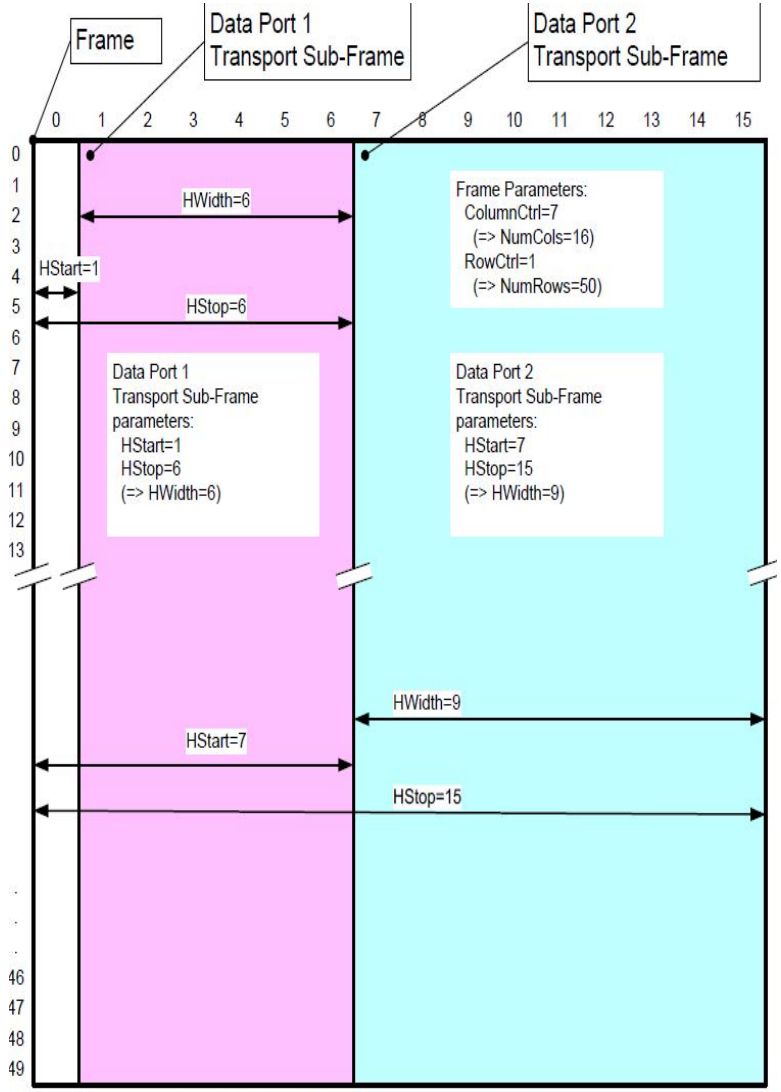
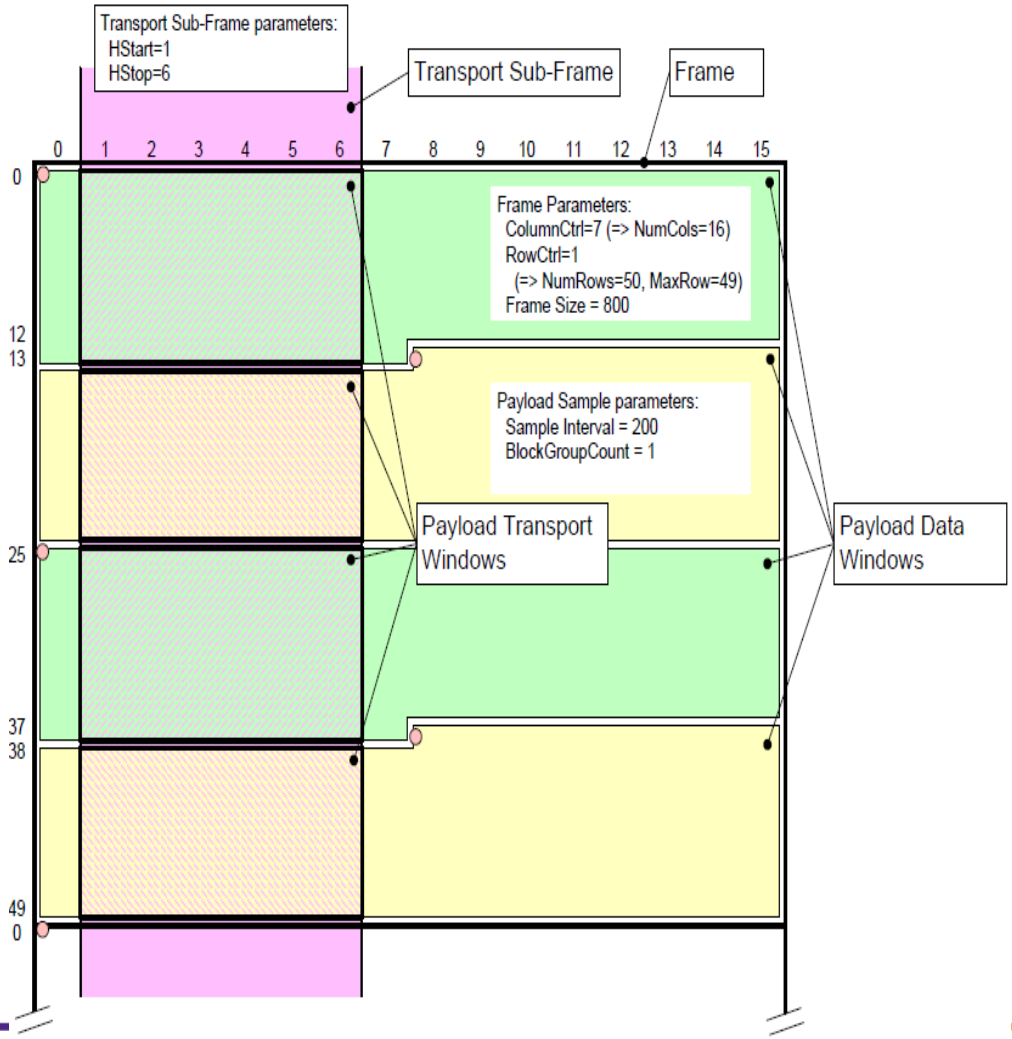


Figure 83 Transport Sub-Frame Examples

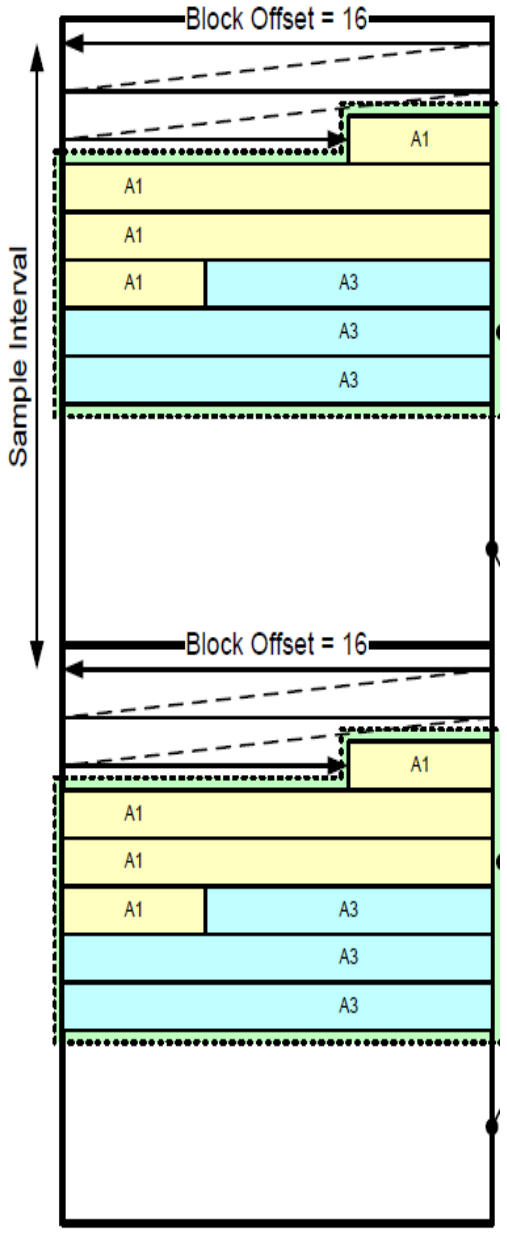
Transport (3): Payload Transport Window

- Intersection between Transport SubFrame and Payload Data Windows



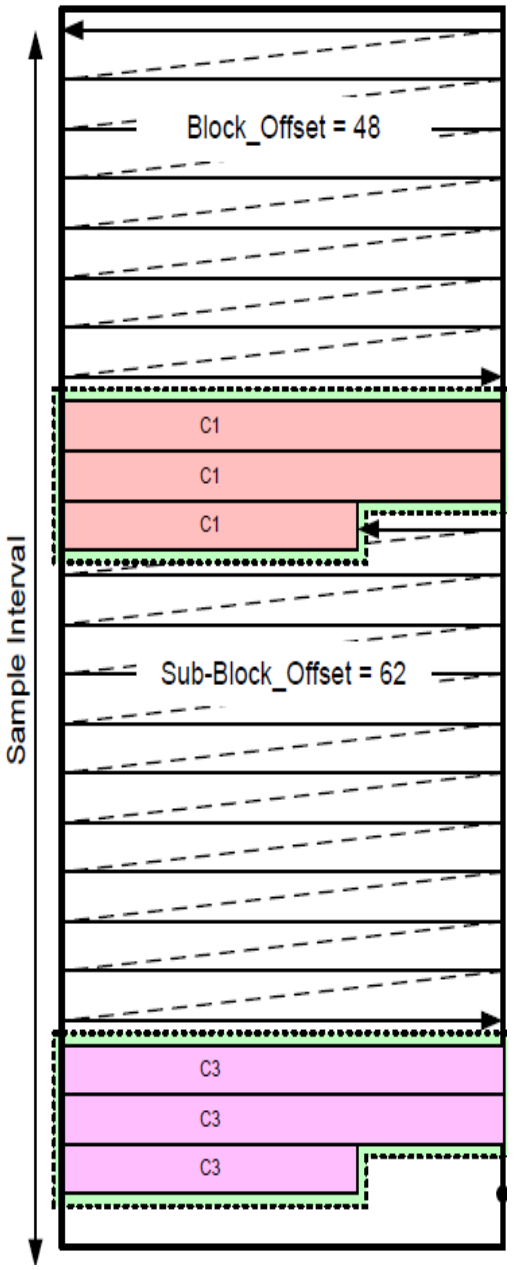
Transport (4): Block-Per-Port

- Payload Data Windows defined by Sample Interval and Transport SubFrame can be shared between similar streams
- Notion of BlockOffset from start of Payload Data Window
- Block-Per-Port mode: all channels packed as single data chunk, from lowest to highest-numbered channel



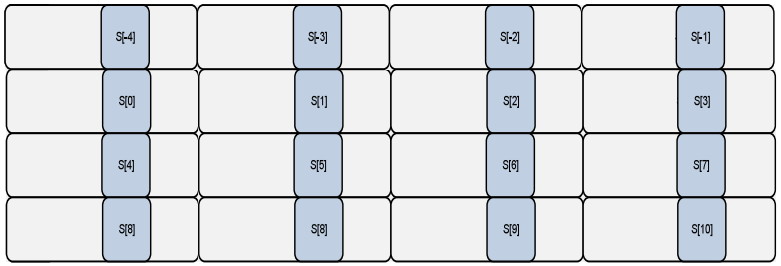
Transport (5): Block-Per-Channel

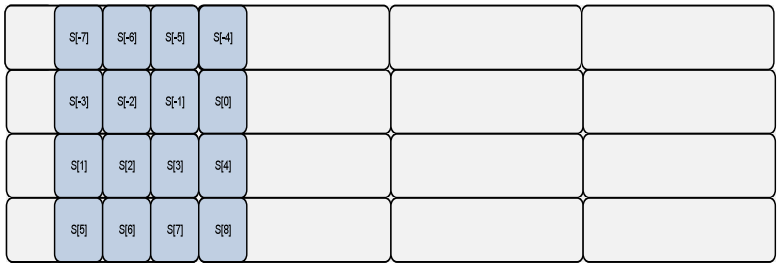
- Channels transmitted in individual chunks within same Sample interval
- Initial Block_Offset
- Inter-sample Sub-Block_Offset
- Benefits of this mode:
- Create 'holes' in bit allocation to be reclaimed by other streams
  - Example: 48kHz stereo signal will use same pattern as 96kHz mono signal
- No impact on buffering or capture/ rendering, only a transport-level capability.



Transport (6): Simplifications

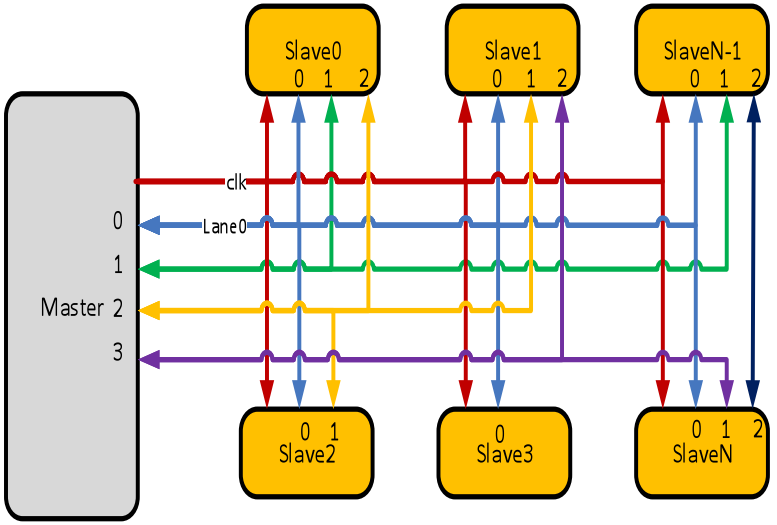
- Full Data port needs to implement
  - Hstart, Hstop, SampleInterval, BlockOffset, SubBlockOffset
- Simple Data ports only needs
  - SampleInterval, BlockOffset
- Grouping:
  - avoid 'vertical stripes' for PDM, ability to group up to 4 successive samples
  - Required for PDM
  - Optional for PCM





Transport (7): multi-lane

- Multilane support is completely optional
- Lane 0 is shared between all devices
  - Common control interface
  - Col0, Rows0..47 are reserved for Command Word
- Lanes 1..7 may be shared or private to group of devices
  - For device-to-device lanes, a bus-keeper must be enabled on one of the devices
- No restrictions on Lane 1..7, all bits can be used including Col0
- Dynamic switch between lanes possible for each Port



Transport synchronization capabilities

- SSP bit in PING frame can be driven at regular intervals
- Useful to maintain alignment
  - Between multiple links with different frame rates
  - Between ports using different sampling rates on the same link
- Typically large enough to be multiple of all sample intervals
- Driven at least once every 100ms
- Linked with Sample Events
  - Used by Slaves to reset SampleInterval counters and resync transport if needed
- Defines 'safe' time position where bus can be reconfigured (frame shape changes, channel/port enablement, etc)
- SSP event can be used to maintain phase coherence between devices.

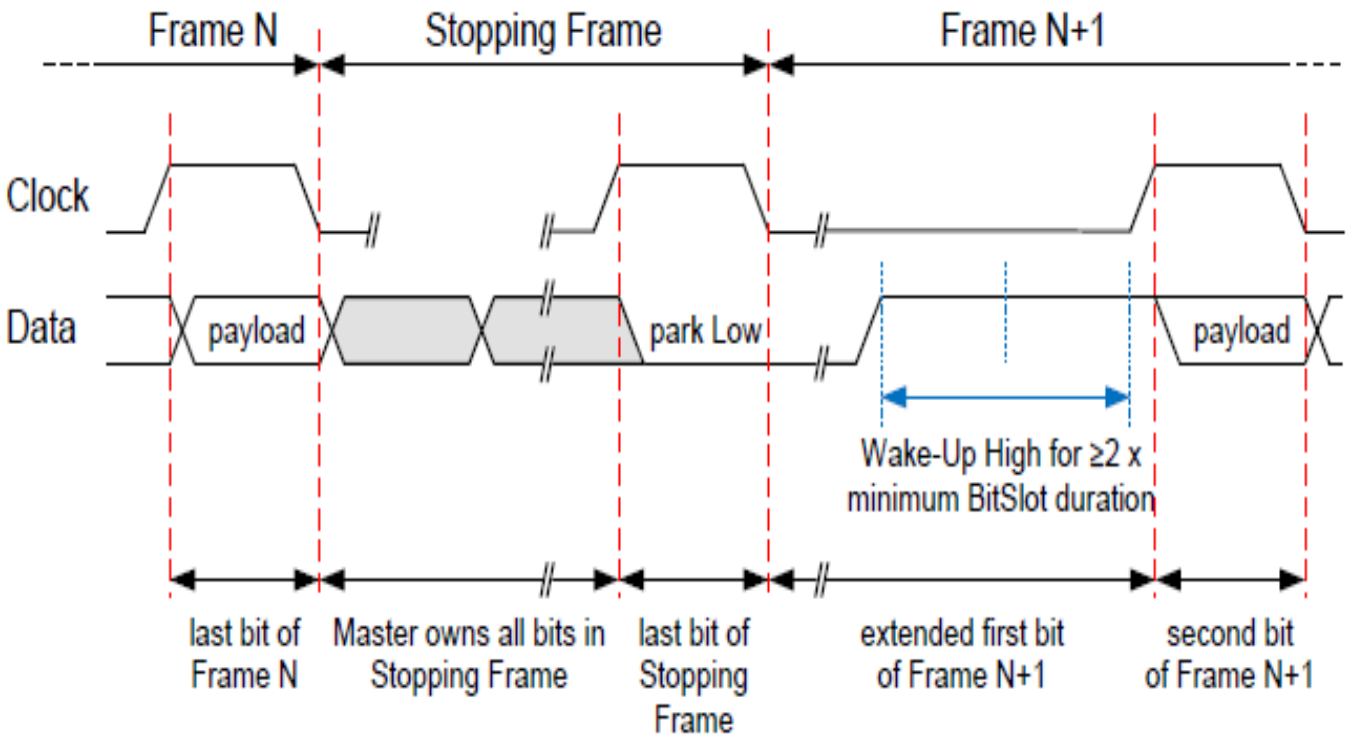
Bulk Transport Protocol (BTP/BRA)

- BTP: use transport protocol to access contiguous registers
  - Dedicated DataPort0 (DP0)
  - Extends command bandwidth limited by frame rate and 8-bit command bandwidth
  - Reconfigurations, Firmware download
- Header defines command, followed by raw data
- CRC protection
- Notion of Initiator/Target(s)
  - Master typically Initiator, Slave is Target
  - Some specialized Slaves could be Initiator (eg. debug tool)
- Protocol is bi-directional by nature
  - Difference between DP0 and other DP1-14 ports (which are single direction)

Clock Stop

ClockStop (1)

- 'ClockStopNow' Command followed by 'Stopping Frame' to let Master drive clock and data to Low



ClockStop (2)

- Wake-ups can be master or Slave initiated
- Master can program which Slaves can wake-up bus

- Master can program two modes in Slave:
  - ClockStopMode0 (mandatory): Slave will keep the context and restart immediately
  - ClockStopMode1 (optional): Slave may lose context, go to very low-power mode and will need to be re-enumerated on startup
- ClockStopMode1 removes need for extra GPIO for e.g. jack detection

Conclusion/Summary

- Questions? Please direct to admin@mipi.org
- More information:
  - MIPI LML WG Public Page
  - MIPI SoundWire Specification Brief
- Join the MIPI LML WG
  - More than 25 companies from across the audio technology ecosystem—including audio peripheral, electronic design automation and silicon vendors as well as OEMs—took part in developing MIPI SoundWire.
  - Requires MIPI Contributor membership
  - Telcos every Tuesday at 07:00 GMT
    - 90-120 minutes depending on the agenda
  - Three face-to-face sessions with all MIPI WGs at member meeting
    - Generally meet for 3-4 days

Backup slides

Comparison with other interfaces

Other bus	Pro SoundWire	Con SoundWire
I²S/TDM	Lower pin count, clock scaling, dynamic slot mapping, burst mode, command embedded with data (No need for I²C/SPI), in-band interrupt capability (no need for GPIO), support for PDM	Slight command overhead, no ability to switch Master/Slave roles for clock
PDM	Clock scaling, embedded command and control, interrupt capability	Overhead is 70% for dual-mic, less than 5% for single mic-mode. In multilane mode power consumption is lower than PDM.
HDAudio	Clock scaling, lower pin count, support for PDM, scales to simple devices	Lower bandwidth device class functionality not yet standardized
SLIMbus	Lower gate count allows for integration in cost-sensitive devices, simpler protocol, low latency PDM support, lower power with adjustable Frame size and double-data rate	No clock and manager hand-over capabilities. Only Master and Monitor can send messages.

Stream aggregation

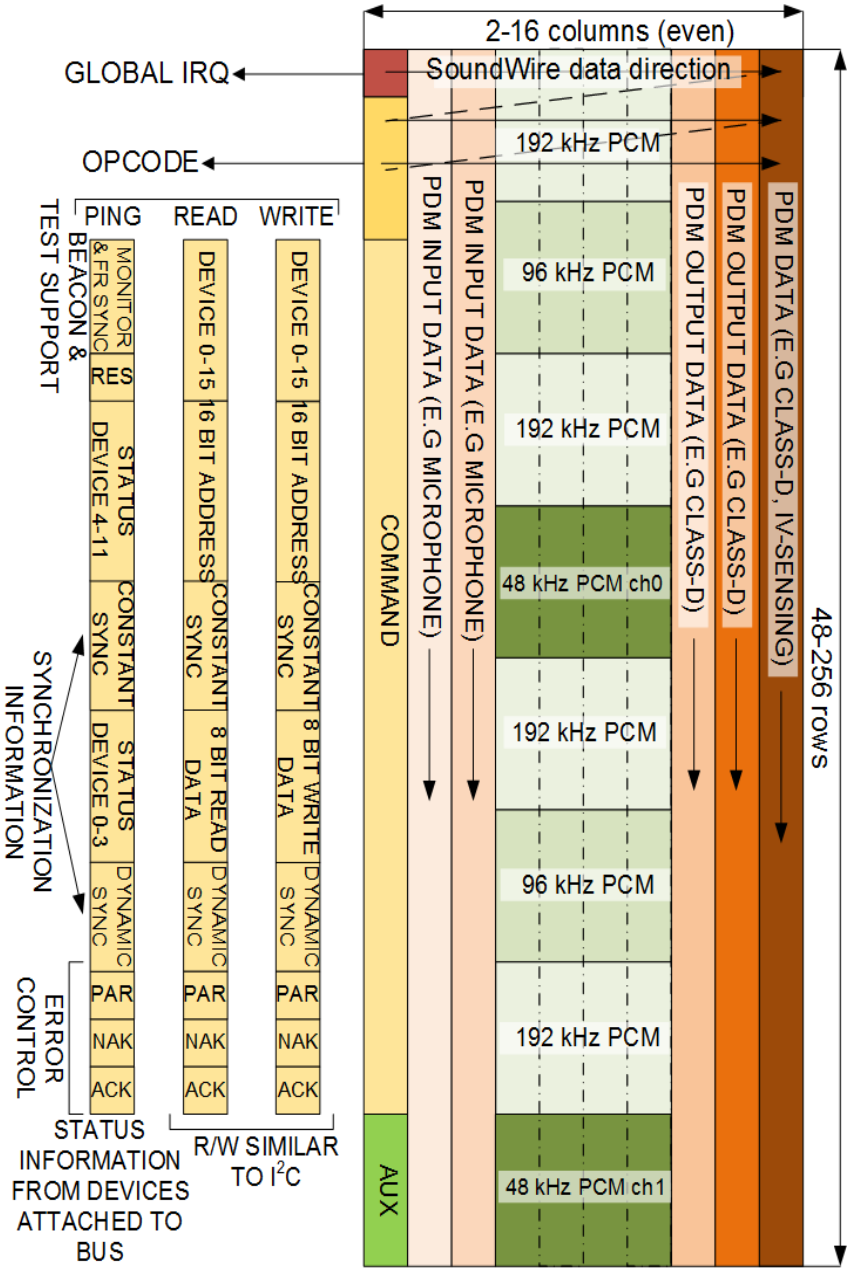
- No notion of 'link' between Source and Sink
- No requirement that Source and Sink port are programmed with same parameters
- -> Ability to perform stream aggregation
- Examples:
- 4 microphones push data on bus, Master retrieves single 4-ch input
- Master writes 2-ch data on 4 ports, Slave reads 8-ch data
- Only requirements:
- Channels enabled on all Devices with common bank switch
- Source ports and Sink ports have same SampleInterval
- 'Smart' bit allocation with no spacing between ports
- Limitation: Aggregation is not possible with Async modes (RX/TX-ready are per port)

Monitor arbitration

- BREQ = 0 -> Master owns Command Word
- BREQ =1 -> Monitor requests Command Word ownership
- BREL = 1 -> Master will yield Command Word ownership at end of frame

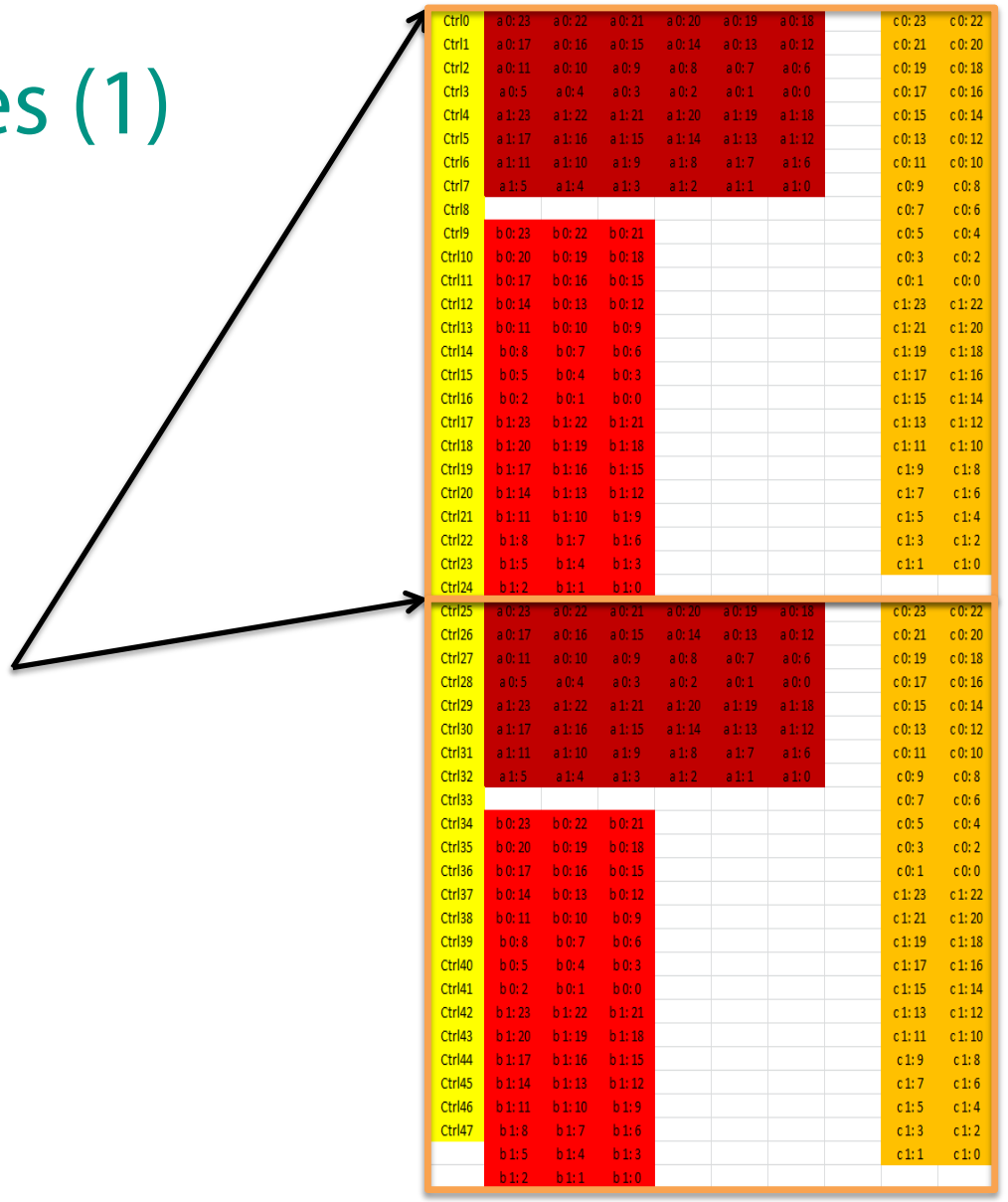
- Notes:
- Monitor will keep Command Word ownership as long as BREQ = 1 and BREL = 1
- Master can reclaim ownership by clearing BREL
- Master always drives static and dynamic bits
- Master does not drive parity bit when Monitor owns bus, but it shall set NAK on parity error
- BREQ=0, BREL=1 is an illegal sequence
- Master is permitted to never release bus ownership (e.g. in a shipping device)
- If Monitor loses sync, command will default to PING with BREQ cleared and Master will reclaim ownership

SoundWire frame overview



Transport examples (1)

- Frame size: 50 rows * 10 cols
- Frame rate: 48kHz
- -> 2 samples per frame
- Sample Events (Row0, Col0) and (Row 25, Col0)
- Position of 3 streams is equivalent in terms of capture/rendering



Transport examples (2)

- Frame size: 50 rows * 8 cols
- Frame rate: 48kHz

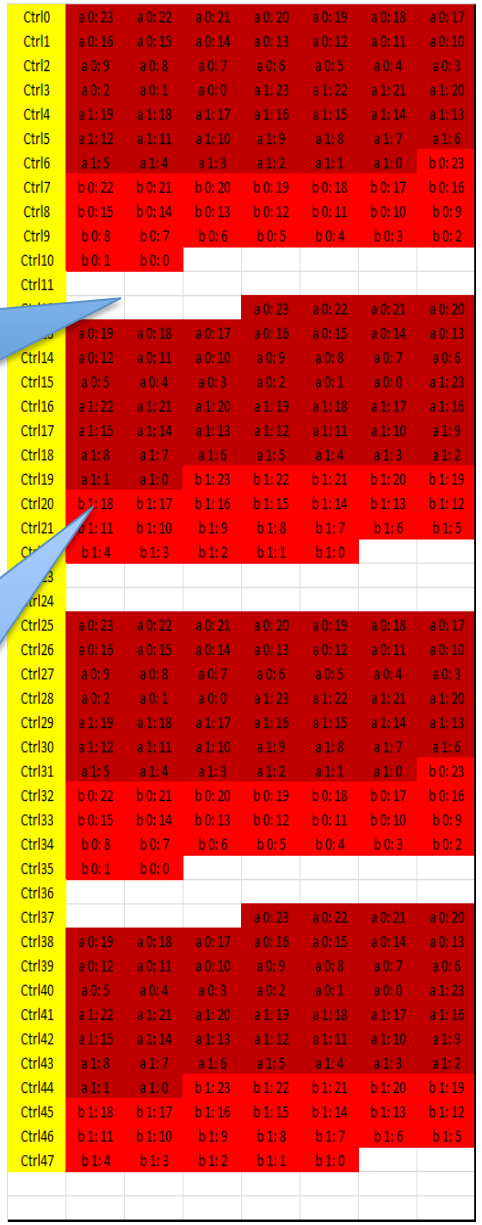
- 192 kHz stream
- -> 4 samples per frame

- 96 kHz stream
- -> 2 samples per frame

- Using BlockPerChannel mode required for 96kHz stream

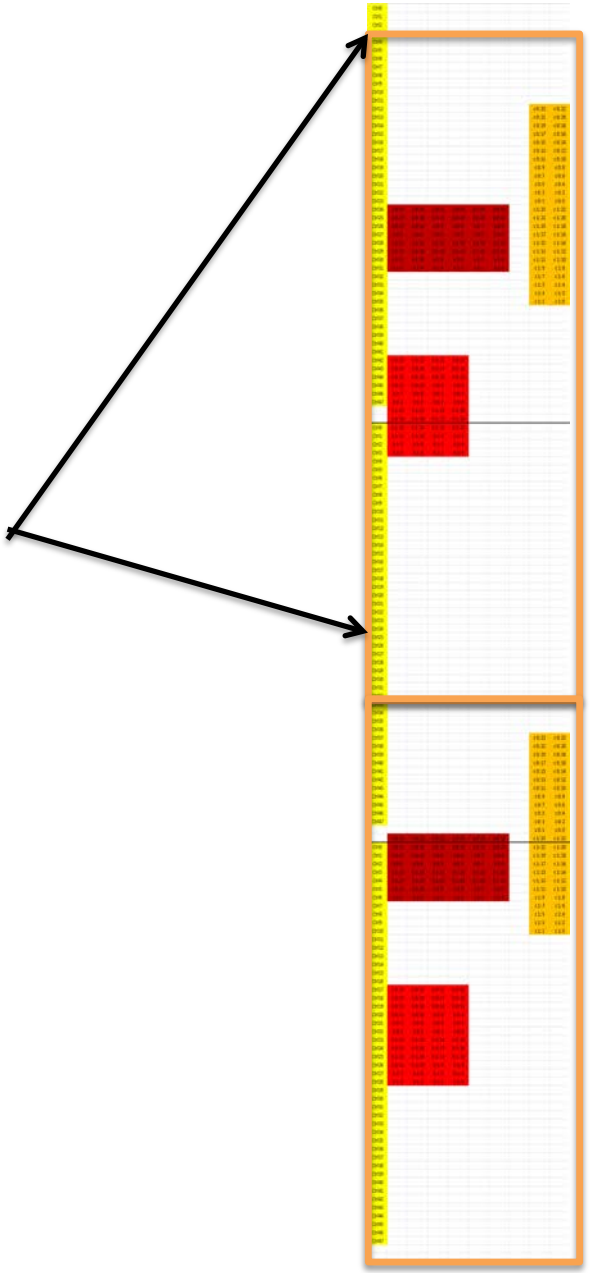
Not enough space for 2nd channel

2nd channel of 96kHz stream pushed after 2nd sample at 192 kHz



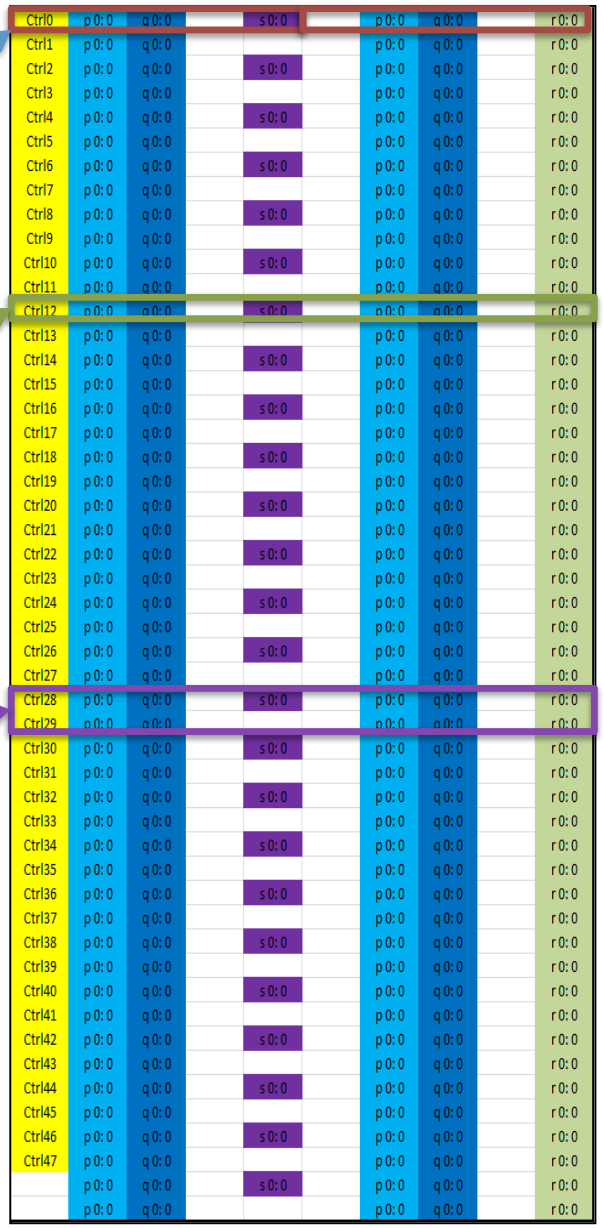
Transport examples (3)

- Frame Rate: 48kHz
- Sample Rate: 32 kHz
- 2 sample intervals for 3 frames
- Samples may be spread across two frames
- Not illegal and even required for some combinations of frame/ sample rate



Transport examples (3)

- Frame size: 50 rows * 10 cols
- PDM:
- 100x oversampling:
  - 2 sample intervals/row
- 50x oversampling:
  - 1 sample interval/row
- 25x oversampling
  - 1 sample for every other row



Clock Stop Prepare

- Support for stopping of the clock
- Same concept as Channel Prepare/Activate
- ClockStopPrepare: Slave may need time to enable an alternate clock source or be ready immediately