# mipi<sup>®</sup> DEVCON

#### MIPI C-PHY<sup>sм</sup>: Introduction

From Basic Theory to Practical Implementation

Mohamed Hafed Introspect Technology





- 23-



#### **Basic Concept of Three Phase Encoding**

EST and







E ST Frank



## Always-Toggle Design Allows for Simple Clock Recovery (100% Transition Density)

12 Carl





## **Key Takeaways**

12 - A.

Three-level single-ended signaling

Non-deterministic transitions based on self-clocked mapping and encoding algorithm





### **Evolution from D-PHY (1 Lane, 4 Wires)**

E LA Frank





#### **Evolution from D-PHY (1 Lane, 4 Wires)** HS A (Up/Down) DQ HS B (Up/Down) LP A LP DP LP B LP DN HS C (Up/Down) $\Lambda\Lambda\Lambda$ N/C LP C LP CP LP CN

2.23 - 200



#### **Evolution from D-PHY (1 Lane, 4 Wires)** HS A (Up/Down) HS B (Up/Down) LP A LP A LP B LP B HS C (Up/Down) ~~~ N/C N/C LP C LP C

E LA Frank



#### **Evolution from D-PHY (1 Lane, 4 Wires)** HS A (Up/Down) HS B CDR (Up/Down) Symbol Decoder LP A LP A and De-Mapper LP B LP B HS C (Up/Down) \_^^\_\_ N/C N/C Less wires LP C Less power LP C More bandwidth

E ST From



#### **Architecturally Flexible** 18 pin Forwarded Sync Clock SoC 18 pin Embedded Clock and Data SoC Limited Fixed Configurations All Configurations Supported @2.5 Gbps @2.5 Gsps X1 DAT CD 5.7 Gbps X2 DAT 11.4 Gbps X3 X1 X4 CD CLK 17.1 Gbps 5.7 Gbps 10 Gbps X4 DAT 22.8 Gbps X1 CD 5.7 Gbps DAT X2 APP APP . . . . . . 11.4 Gbps . . . X1 CD X1 DAT 5.7 Gbps X2 2.5 Gbps CLK 11.4 Gbps X1 X1 X1 CD 5.7 Gbps 5.7 Gbps 5.7 Gbps DAT X1 X1 X1 X1 X1 CD 2.5 Gbps CLK 5.7 Gbps 5.7 Gbps 5.7 Gbps 5.7 Gbps D-PHY v1.2 Total Aggregate BW: 15 Gbps C-PHY v1.0 Total Aggregate BW: 34.3 Gbps Source: MIPI Alliance

FEST and

# mipi<sup>®</sup> DEVCON

## Mapping and Encoding





EST AN



#### **Wire States**

- A wire state is the collection of A, B, and C
- 6 possible wire states

	ANALOG			DIGITAL (3 bits)		
Α	В	С	A>B	B>C	C>A	Wire state name
HIGH	LOW	MID	1	0	0	+x
LOW	HIGH	MID	0	1	1	-x
MID	HIGH	LOW	0	1	0	+γ
MID	LOW	HIGH	1	0	1	-у
LOW	MID	HIGH	0	0	1	+z
HIGH	MID	LOW	1	1	0	-Z



#### Symbols: Now We're Transmitting!

• A symbol represents a transition between two wire states

ESS From

• 5 possible symbols

		Symbol (3	bits)
	Flip	Rotate	Polarity
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	DC	DC



	Flip		Rotate		Polarity
0	-	0	Decr. letter	0	-
1	Same letter, toggle sign.	1	Incr. letter	1	Toggle sign



### 

• C-PHY defines a mapping between 7-symbol words and 16-bit integers



{4444444} Post (End-of-Packet marker)



#### **Well Defined Algorithms from MIPI Alliance**

EST AN

Intege	r rang	e Flip bits			-	-	Ro	tate and	Polarity bits
0322000	OxFFFF	F: Fip=[1,0,1,0,0,0]	0,0] Integer=	L 1,	1,	1,	1,	L, 1,ro:	5, po5, ro3, po3, ro2, po2, ro1, po1, ro0, po0]
032-800-	OxFBFF	F: Fip=[0,1,1,0,0,0]	0,0] Integer=	L 1,	1,	1,	1,	L, U,ro	6, po6, ro3, po3, ro2, po2, ro1, po1, ro0, po0]
0xF400-	OXE /FF	$F_{1} = [1,0,0,1,0,0]$	01 Integer	L 1,	1,	1,	1,	0, 1,10	5, po5, ro4, po4, ro2, po2, ro1, po1, ro0, po0]
0xEC00-0	VESEE	$F_{1} = [0, 1, 0, 1, 0, 0]$	01 Integer	L 1,	1	1	ā'	$1 1 m^{-1}$	(5, po5, ro4, po4, ro2, po2, ro1, po1, ro0, po0)
0xE800-0	XEBFF :	Flip=[1,0,0,0,1,0]	01 Integer=	r 1.	1.	1.	ő.	1, 0, ro	5, po5, ro4, po4, ro3, po3, ro1, po1, ro0, po0]
0xE400-0	ETEF:	Flip=[0,1,0,0,1,0]	.01 Integer=	r 1.	1.	1.	o,	0, 1, ro	1000, 001, 100, 101, 101, 200, 201, 400, 901, 00
0xE000-0	E3FF:	Flip=[0.0.1.0.1.0]	01 Integer=	1 <u>1</u> ,	1.	1.	o.	0, 0, ro	[000, 001, 100, 101, 200, 201, 201, 201,
0xDC00-0x	DFFF:	Flip=[0,0,0,1,1,0	01 Integer=	1,	1,	o,	1,	1, 1,rc	[000, 007, 100, 107, 400, 407, 200, 207, 306, 306
0xD800-0x	DBFF:	Flip=[1,0,0,0,0,1,	0] Integer=[	1,	1,	ο,	1,	1, 0,rc	[00g, 00g, 20g, 20g, 80g, 80g, 40g, 40g, 20g, 20g
0xD400-0xL	7FF :	Flip=[0,1,0,0,0,1,	0] Integer=[	1,	1,	ο,	1,	0, 1,rc	[000, 001, 200, 201, 500, 103, ro2, po2, ro0, po0]
0xD000-0xD	3FF:	Flip=[0,0,1,0,0,1,	0] Integer=[	1,	1,	ο,	1,	0, 0,rd	[00q, 00r, 20q, 20r, 60q, 60r, 50q, 707, 90q
0xCC00-0xC	FFF :	Flip=[0,0,0,1,0,1,	0] Integer=[	1,	1,	ο,	ο,	1, 1,r	06, po6, ro5, po5, ro4, po4, ro2, po2, ro0, po0]
0xC800-0xCE	BFF :	Flip=[0,0,0,0,1,1,0	] Integer=[	1,	1,	ο,	ο,	1, 0,r	[00g, 00r, E0g, E0r, 40g, 40r, 20g, 20r, 30g, 30
0xC400-0xC7	FF: 3	Flip=[1,0,0,0,0,0,1	] Integer=[	1,	1,	o,	ο,	0, 1,r	[10g, 107, 20g, 207, 80g, 807, 904, po4, 707, 20g, 20
0xC000-0xC3	FF: I	lip=[0,1,0,0,0,1]	] Integer=[	1.	1.	o,	ο,	0, 0,r	100, po6, ro4, po4, ro3, po3, ro2, po2, ro1, po1
0xBC00-0xBF1	F: F	lip=[0.0.1.0.0.1]	1 Integer=[	1	0.	1.	1.	1. 1.r	rof, po6, ro5, ro3, ro3, ro3, ro2, ro1, ro1)
0xB800-0xBBE		lip=[0, 0, 0, 1, 0, 0, 1]	1 Integer=[	1	o ′	1	1	1 0 m	$r_{0}$ $r_{0$
0×P400-0×P7F		lip=[0,0,0,0,1,0,1]	] Integer-[	1	ő,	1	1	0 1 2	$r_{00}$ pool
0,2000-0,2072			J Integer-[	±,	°,	1	±,	0, 1,1	
0xB000-0xB3FI	:	.ip=[0,0,0,0,0,1,1]	Integer=[	Τ,	Ο,	±,	т,	0, 0,1	rob, pob, rob, pob, ro4, po4, ro3, po3, ro2, po2
0xA000-0xAFFF	: F1	ip=[1,0,0,0,0,0,0]	Integer=[	ı,	Ο,	ı,	0,1	5,po5,r	00, 00, 100, 101, 200, 201, 201, 201, 20
0x9000-0x9FFF	: Fl:	ip=[0,1,0,0,0,0,0]	Integer=[	1,	ο,	ο,	1,r	96,po6,1	00q, 00r, 10q, 10r, 20q, 20r, 80q, 80r, 100 po
0x8000-0x8FFF	: Fli	p=[0,0,1,0,0,0,0]	Integer=[	1,	ο,	ο,	0,1	6,po6,1	00, 00r, 10g, 10r, 20g, 20r, 60g, 703 , 700 , 705 , 705 , 705
0x7000-0x7FFF:	Fli	p = [0, 0, 0, 1, 0, 0] = q	Integer=[	ο.	1.	1.	1.1	6.006.1	og, 0or, 1og, 1or, 2og, 2or, 4og, 4or, 2og, 2or
0×6000-0×6FFF.	Fli	r = [0, 0, 0, 0, 1, 0, 0]	Integer=[	0	1	1	0, 1	6 006	$r_{0}5$ $r_{0}5$ $r_{0}4$ $r_{0}4$ $r_{0}3$ $r_{0}3$ $r_{0}1$ $r_{0}1$ $r_{0}0$ $r_{0}$
CACCOC CACFFF.	- 11) 		Tutoger-[	~	- /	÷ ′	1,1	, poo,	
XSUUU-UXSEFF:	FII	<u>[0,0,0,0,0,1,0]</u>	integer=[	υ,	т,	υ,	1,1	, poe, so	ros, pos, ro4, po4, ro3, po3, ro2, po2, ro0, po
x4000-0x4FFF:	Flip	=[0,0,0,0,0,0,1]	Integer=[	ο,	1,	ο,	0,1	o6,po6,	ro5, po5, ro4, po4, ro3, po3, ro2, po2, ro1, p
x0000-0x3FFF:	Flip	=[0,0,0,0,0,0,0]	Integer=[	ο,	0,r	06, p	06,1	, 20q, 2c	ro4, po4, ro3, po3, ro2, po2, ro1, po1, ro0,



# "Don't Even Worry About It"

Ell' and











EST From

# mipi® DEVCON

#### Practical Experiences



### Tx: Both Mapping and Encoding Before Serializer

123 - A.





## Rx: Avoiding False Sync Detection (Problem Statement)







123 - A.





The second





The second



#### CSI-2 Long Packets in C-PHY Filler Payload Footer Header Header Payload CRC (msbyte) Wordcount (msbyte) Payload CRC (Isbyte) Wordcount (msbyte) Wordcount (Isbyte) Wordcount (Isbyte) Each pair of Data Identifier Payload byte 3 Payload byte 5 Payload byte 2 Payload byte 0 Payload byte 4 Data Identifier Payload byte 1 CRC (msbyte) CRC (msbyte) CRC (Isbyte) CRC (Isbyte) bytes makes an Filler byte Reserved Reserved Filler byte integer 0x 05 0x 00 0x **0x** 02 **03** 0x 04 0x 00 0x 80 0x 07 0х 9В 0x 07 0x 9B Ох 0x 0x B5 0х С1 Ох 0x 24 0х 80 0x 08 0x 00 0x Оx Bytes ... 24 00 01 08 00 Integers – 1 lane 0x**24**00 0x**07**80 0x**9B**08 0x**07**80 0x**9B**08 0x**03**02 0x**C1***B5* 0x**24**00 0x**01**00 0x**05**04 ... Integer LSB = $1^{st}$ byte Integer MSB = $2^{nd}$ byte And so on... Integers - 4 lanes ... distributed

FEST and













Carl and





Sin and





Si - and







#### **DSI-2 Sample Protocol Analyzer Trace** 🖳 CPHY DataCapture: Run\_2016-08-05\_1129\_2lane / dsiDataCapture1 HS\_immediate Iane1 ○ Iane2 ○ Iane3 ○ Iane4 Go To: Packet#.. $\sim$ HS Bursts DSI Packets Burst VC DT DT name Header CRC WC Short Payload CRC Distributed 0 0x39 DcsLongWrite 0x0192 integers/symbols as 0 0x39 DcsLongWrite 0x0192 3841 0x6414 0 0 0x39 DcsLongWrite 0x0192 0x63D5 0 3841 seen on the lanes 0 0 0x39 DcsLongWrite 0x0192 0xA823 3841 0 0x39 DcsLongWrite 0x0192 3841 0xD8F4 0 Packet 0 Detail **|**<< << >>| >> < 0 2 3 4 5 6 7 8 1 8139 '1230410' 8192 '2012410' DSYNC '3444443' 80F0 '0033400' DSYNC '3444443' 012C '0320100' 1207 '3100201' 000B '3200000' 0002 '2000000' lane1 data 80F0 '0033400' DSYNC '3444443' 8139 '1230410' 8192 '2012410' DSYNC '3444443' 000F '3300000' 0104 '0100100' 0000 '0000030' 0007 '3100000' lane2 data: 3981F080 9281 3981 F0809281 2C010F00 07120401 0B00000C 02000700 bytes: **Transmitted bytes**

The second



#### **DSI-2 Sample Protocol Analyzer Trace**

TERS FROM

IS Burete Do	CI Dealesta				0.0	o. Duist														
Burst ID	NumData	PreBegin	ProgSeg	PreEnd	Post	NumBits	SyncOffset	PostOffset	DSI Packet	3										-
)	20306	97	14	7	63	144320	223	142372	21	-										
	122808	97	14	7	60	861824	213	859876	<u>121</u>											
2	122808	97	14	7	59	861760	198	859861	<u>127</u>											
ļ.	122808	97	14	7	64	861824	249	859912	<u>127</u>											
ł	122808	97	14	7	61	861824	235	859898	<u>118</u>											
	122808	97	14	7	57	861824	224	859887	<u>108</u>											
Burst O Detail se 223 wir wir wir wireSta symb	TEAB: 10 TEBC: 10 TECA: 01 Ates: 61 Dols: 34	K< 0101000 0101010 1010111 1616131 444443	<pre> <!-- 110111 001010 011001 453465 230410 </pre--></pre>	11101 10010 01001 65425 20124	1001 00010 0110 54134	> 010110 0101001 0101011 1343453 1444430	>> 01110100 11001001 10010011 36452413 00334003	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	SYNC 0000110 1011100 0101011 2123651 3201003	POST 10101010101010101010101010101010101010	PKT 1101111 1011010 0110101 5536565 0000020	0110101 1101110 1011000 3653624 0000010	.000010 011100 101001 123241 0001300	1000110 0011101 0101011 4123653 2001000	01100011 1010101 00111000 86531246 00001001	010000 100101 001110 241312 100310	00010010 L0100101 L1001000 31241242 L1000002	10001: 001110 01010: 41236: 000100	1011010 0110111 101100 5365362 0000001	11 01 00 46 01
Surst O Detail see 223 wir wir wire symb data (d data (h	TEAB: 10 TEBC: 10 TECA: 01 TECA: 01 TEC	k< 0101000 0101010 1010111 1616131 1444431	<pre>&lt;&lt; 110111 001010 011001 453465 230410 33081 8139 8139 ************************************</pre>	111101 010010 01001 665425 20124 . 331 9 81	1001 00010 0110 54134 1034 1034 1034 1034 1034 1034 1034	> 010110 0101001 0101011 0343453 0444430 DSYNC	>> 01110100 1001001 36452413 00334003 33008 80F0 2410' D	>>  01010110 0101001 0101011 34343453 34444430 DSYNC	SYNC 0000110 1011100 0101011 2123651 3201003 300 012C	POST 10101010 01111111 1010001 53636263 100201320 4615 1207 '0033400'	PKT 1101111 1011010 0110101 5536565 0000020 11 000B	0110101 1101100 3653624 0000010 2 0002	000010 011100 101001 123241 0001300 3329 0D01	1000110 0011101 0101011 4123653 2001000 264 0108	01100011 1010101 00111000 36531246 00001001 256 0100	010000 100101 241312 100310 1797 0705	00010010 10100101 11001000 31241242 1100002 5 0005	10001: 001110 01010: 41236: 000100 25: 0102	0011010 0110111 101100 0365362 0000001 8 102 2 040	11 01 00 46 01 4 0
Burst 0 Detail tee 223 wir wir wireSta symb data (d data (h	reAB: 10 reBC: 10 reCA: 01 ttes: 61 ools: 36 iect: iect: iect:	I           0101000           0101111           1616131           1444431           Iane1 data:           Iane2 data:	110111001010011001453465230410330818139 '18139 '18070 '0	<pre></pre>	(1001 00010 0110 04134 1034 .70 .92	> 0101010 0101001 0101001 1343453 4444430 DSYNC 192 '201	>> 01110100 11001001 36452413 00334003 33008 80F0 2410' D 44443' 8	>>  01010110 0101001 0101011 0101011 0101011 0101011 0101011 0101011 0101011 0101011 01010110 01010110 010101010 010101001 0101010010	SYNC 0000110 1011100 0101011 2123651 3201003 300 012C 443' 80F 10' 819	POST 10101010 01111111 11010001 536362630 100201320 4615 1207 0 '0038400' 2 '2012410'	PKT 1101111 1011010 110101 5536565 0000020 11 000B DSYNC DSYNC	0110101 1101110 1011000 3653624 0000010 2 0002 '3444443' '3444443'	0000010 0011100 1010101 1123241 0001300 3329 0D01 012C *	1000110 0011101 0101011 4123653 2001000 264 0108 0320100' 3300000'	01100011 1010101 00111000 36531246 00001001 256 0100 1207 '31 0104 '01	0100000 100101 241312 100310 1797 0705	00010010 10100101 1001000 1241242 1000002 5 0005 0008 '320 0008 '320	10001: 001110 010100 412363 000100 258 0102	011010 0110111 101100 3365362 0000001 3 102 2 040 0002 '20 0007 '31	11 01 00 46 01 4 0 2000 2000
Burst O Detail Fiel 223 wir wir wireSta symb data (d data (h	reAB: 10 reBC: 10 reCA: 01 ttes: 61 reC3: 36 rec4: rec4: 10 rec4: 10 rec4: 10 rec4: 10 rec4: 10 rec4: 10 rec4: 10 rec4: 10 rec2: 10 rec3: 10 rec2: 10 rec2: 10 rec2: 10 rec2: 10 rec2: 10 rec2: 10 rec2: 10 rec2: 10 rec2: 10 rec3:	I<         0101000           0101000         0101010           101011         1616131           1616131         144443           Iane1 data:         Iane1 data:           Iane2 data:         bytes:	<pre> <!--  110111 001010 011001 453465 230410 33081 8139 8139 8139 8139 8139 8139 8139 81</th--><th>111101 010010 001001 665425 020124 . 331 9 81 </th><th>(1001 00010 0110 04134 1034 1034 1034 103</th><th>&gt; 010110 101000 101011 1343453 4444430 DSYNC 192 '201 192 '201 SYNC '34 281</th><th>&gt;&gt; 01110100 11001001 36452413 00334003 33008 80F0 2410' D 44443' 8</th><th>&gt;&gt;  01010110 0101001 0101011 04343453 0444430 DSYNC DSYNC 3444430 DSYNC 3444430 DSYNC 3444 139 '12304 3981</th><th>SYNC 0000110 1011100 0101011 2123651 3201003 300 012C 443' 80F 10' 819 F08</th><th>POST 101010100 01111111 11010001 536362636 100201326 4615 1207 0 '0033400' 2 '2012410' 09281</th><th>PKT 1101111 1011010 110101 5536565 0000020 11 000B DSYNC DSYNC</th><th>0110101 1101110 1011000 3653624 0000010 2 0002 '3444443' '3444443'</th><th>000010 011100 101001 123241 0001300 3329 0D01 012C ' 000F ' 2C010F</th><th>1000110 0011101 0101011 4123653 2001000 264 0108 0320100' 3300000'</th><th>01100011 1010101 00111000 36531246 00001001 256 0100 1207 '31 0104 '01 07120403</th><th>0100000 100101 2413122 100310 1797 0705</th><th>00010010 10100101 1001000 1241242 1000002 5 0005 0008 '320 0000 '000 00000000</th><th>10001 00111( 01010) 412365 00010( 256 0102</th><th>0011010 0110111 101100 3365362 0000001 3 102 2 040 0002 '20 0007 '31 0200700</th><th>11 01 00 46 01 4 0</th></pre>	111101 010010 001001 665425 020124 . 331 9 81 	(1001 00010 0110 04134 1034 1034 1034 103	> 010110 101000 101011 1343453 4444430 DSYNC 192 '201 192 '201 SYNC '34 281	>> 01110100 11001001 36452413 00334003 33008 80F0 2410' D 44443' 8	>>  01010110 0101001 0101011 04343453 0444430 DSYNC DSYNC 3444430 DSYNC 3444430 DSYNC 3444 139 '12304 3981	SYNC 0000110 1011100 0101011 2123651 3201003 300 012C 443' 80F 10' 819 F08	POST 101010100 01111111 11010001 536362636 100201326 4615 1207 0 '0033400' 2 '2012410' 09281	PKT 1101111 1011010 110101 5536565 0000020 11 000B DSYNC DSYNC	0110101 1101110 1011000 3653624 0000010 2 0002 '3444443' '3444443'	000010 011100 101001 123241 0001300 3329 0D01 012C ' 000F ' 2C010F	1000110 0011101 0101011 4123653 2001000 264 0108 0320100' 3300000'	01100011 1010101 00111000 36531246 00001001 256 0100 1207 '31 0104 '01 07120403	0100000 100101 2413122 100310 1797 0705	00010010 10100101 1001000 1241242 1000002 5 0005 0008 '320 0000 '000 00000000	10001 00111( 01010) 412365 00010( 256 0102	0011010 0110111 101100 3365362 0000001 3 102 2 040 0002 '20 0007 '31 0200700	11 01 00 46 01 4 0



### **Key Takeaways**

Tx mapping and encoding in parallel domain

Rx false sync avoidance required pre-begin monitoring

Packet header definition required careful design of SYNC manipulation (both Tx and Rx)

CSI-2 & DSI-2 treat SYNC insertion differently





## April 10, 2014: World's First C-PHY Interoperability!

Leading Image Sensor Manufacturer Test Chip (Tx)



introspect technology

First Eye Diagram

#### **First Packet Received**

#### MipiCphyDataCaptureViewer Captured: Wire Lane 1 💿 Lane 2 🔘 Lane 3 🔘 Lane 4 40 Bits: 3612:3701 << > >> wireState: 00000002645132645132645132645132645132645132645134343453635365214241563124363123512425632531231 3654 60328 412 data(dec): 160 17450 36890 data (hex) : 00A0 442A 901A 0E46 EBA8 A1 phase delay (ps)



#### **Thank You!**