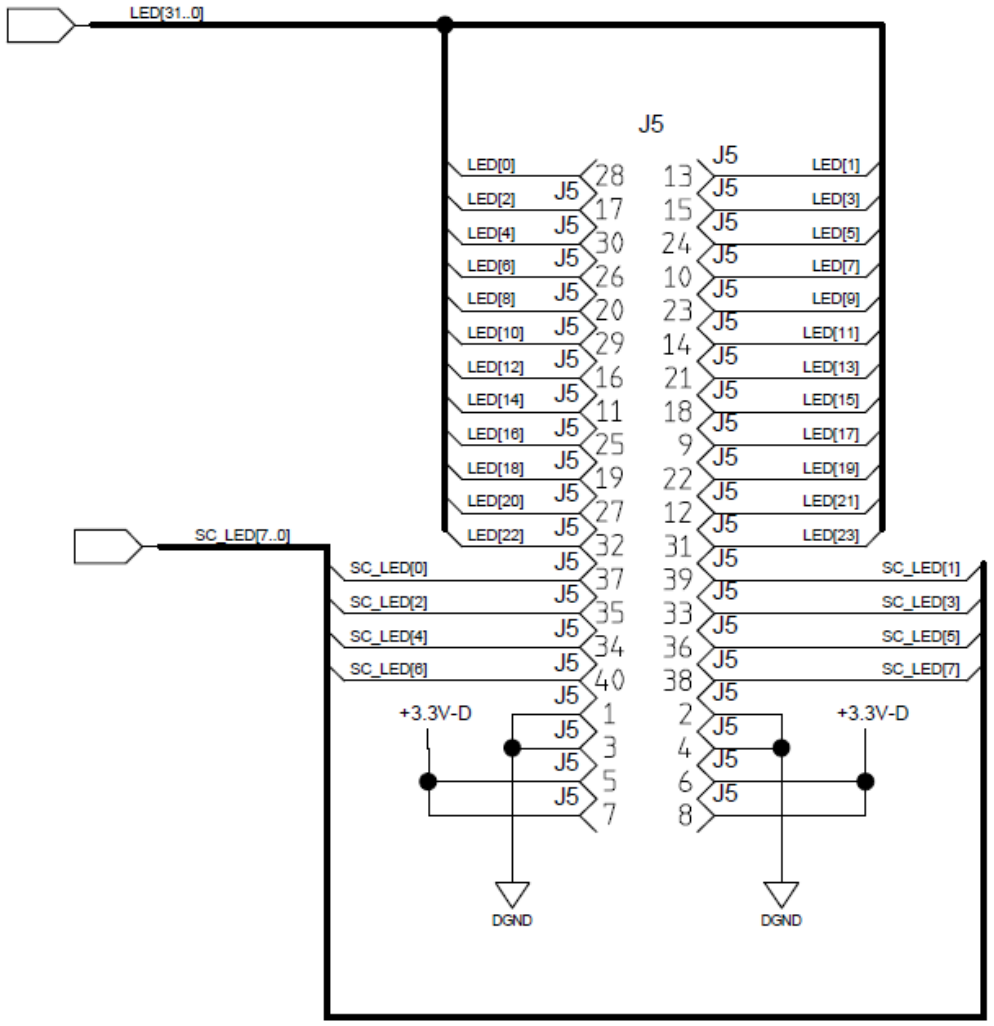
FEM-LED問題に対する調査

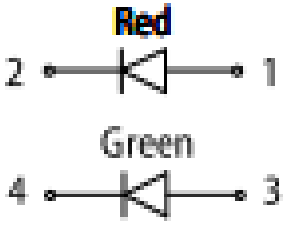
- ROCの動作確認中、FEM-LEDが光らないROCが見つかった
  - この場合はLED= 6
  - 問題はなにか？
- 調べたいこと
  - LEDはどのような時に光るのか？
  - 光っているLEDとROCポートとの関係？

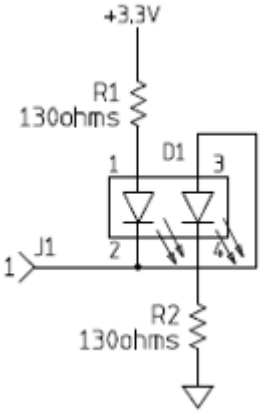
FEM上のLED基板の光り方/光らせ方



- FEM中のLED制御信号
- SlowControl-FPGAでの制御
  - SC_LED[7:0]
- DATA-FPGAでの制御
  - LED[23:0]

- 各線は、信号レベルを1,0, Zに設定すると、緑、赤、OFFになる





LEDの光り方

J1の信号レベル	赤	緑
1	OFF	ON
0	ON	OFF
Z	OFF	OFF

SYNC状態を示すLED制御のFPGAコード

```
LED_OUT(23) <= LED_TEST(0) when LED_TEST(1) = '1' else
               ‘Z‘              when (LED_TEST(1) = '0' and LED_TEST(0) = '1') else
               ‘1‘               when SYNC_OK_3_3 = '1' else
               'Z';
```

- データファイバの状態LEDは１(緑)かZ(OFF)の状態しかない。
- LED_TEST[1:0] は、FEMIBのスイッチで決まっているFPGA_ADDR_VME[2:0]の下位2ビット
  - FPGA_ADDR_VME <= FPGA_ADDR_REF;　(FEMIB)
  - LED_TEST <= FPGA_ADDR_VME(1 downto 0) when FPGA_ADDR_VME(2) = '1' else (others => '0’);
    - FPGA_ADDR_VME[2]=1のとき、 LED_TEST =FPGA_ADDR_VME[1:0], でなければLED_TEST=00、
  - SYNC_OK_3_3の状態をLEDで表示するためには、FPGA_ADDR_VME<=4でなければならない。
- SYNC_OKの値によって、1or Zを出力

SYNC_OKの定義

- SYNC_OK=0
  - データがK30.7だったとき＝SYNCのリセットコード
- SYNC_OK=1
  - SYNC用データをすべて正しく受診したとき=SYNC_LOGIC=1

- FEM DATA-FPGA中のコード

```
SYNC_LOGIC <= COND_1_BUF and COND_2_BUF and COND_3_BUF; ^M
DELAY_ARR <= DELAY_ARR(11 downto 0) & SYNC_LOGIC;^M
if (DATA_BUF(17 downto 16) = "11" and ^M
      (DATA_BUF(15 downto 0) = K30_7&K30_7 or DATA_BUF(15 downto 0) = K31_7&K31_7 )) then
   SYNC_OK_int <= '0';^M
end if;^M
if (DELAY_ARR(12) = '1') then^M
   SYNC_OK_int <= '1';^M
end if;^M
```

データファイバをSYNCする方法　ROC側(送信側)

- GUIからのLATCH_FPGA関数でFIBERの同期を行う。
- 同期手順
  - LATCH_FPGA関数によりROCのDATA-FPGAにLATCHパルス(1クロック分)が送られる。
  - ROC内のsync_block_logicでLATCHを受け取り、SYNCロジックを実行する。
  - SYNCロジック
    - LATCH受け取り後、以下のデータを送る

クロック毎のステップ	1，K30.7（resetSYNC）	2，K28.5/D5.6 Alignment	3，K28.5/D5.6 Alignment	4	5	6	7	8
DATA	FE＆FE	C5&BC	C5&BC	E3&E3	5E&5E	76&76	0	0
MSB	1	0	0	0	0	0	0	0
LSB	1	1	1	0	0	0	0	0

Alignmentは、SerdesチップがByteの切れ目(Boundary)を判定するために必要
CommaDetectという。

NOTE:　FPHXからのデータが来ていない時(DATA=0), 常時alignment data (BC&C5, LSB=1)を出力している。
通常通信時(FPHXからデータあり、LATCH=0)、Dコード(LSB, MSB=0)としてデータを出力している

データファイバをSYNCする方法　FEM受信側

- 同期手順
  - SYNK＝OFFの条件：　以下２つが成り立つ場合、SYNC_OK=0にする
    1. デシリアライザ(tlk2711)がK codeを受信したとき＝RKLSB=RKMSB=1
    2. 16 bitデータは、 0xFE (k30.7) & 0xFE か　0xFF & 0xFF である
       1. K30.7=EndBadPacket

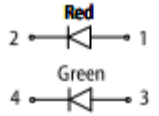
  - SYNC_ONの条件：データが以下のように並ぶと、12CLOCK後にSYNC_OK=1にする
    - 一旦SYNC_ONになると、RESETまでOFFならない。

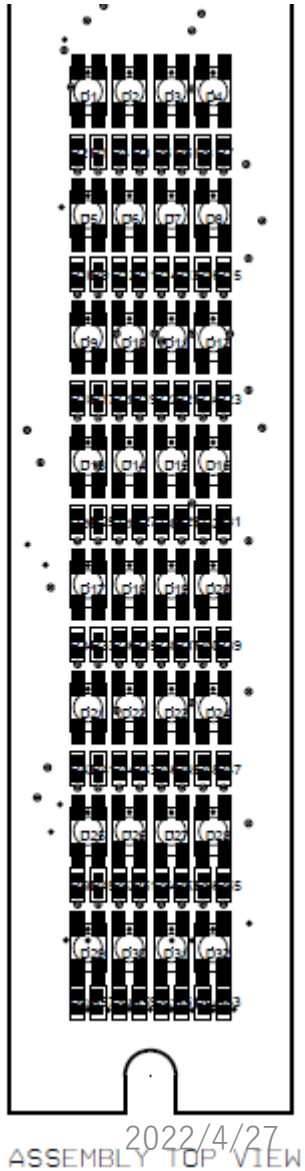
	1	2	3
受信データ	0xE3E3	0x5E5E	0x7676
実際に比較に使うデータ 4ビット毎にFLIPしている	0x7C7C	0xA7A7	0xE6E6

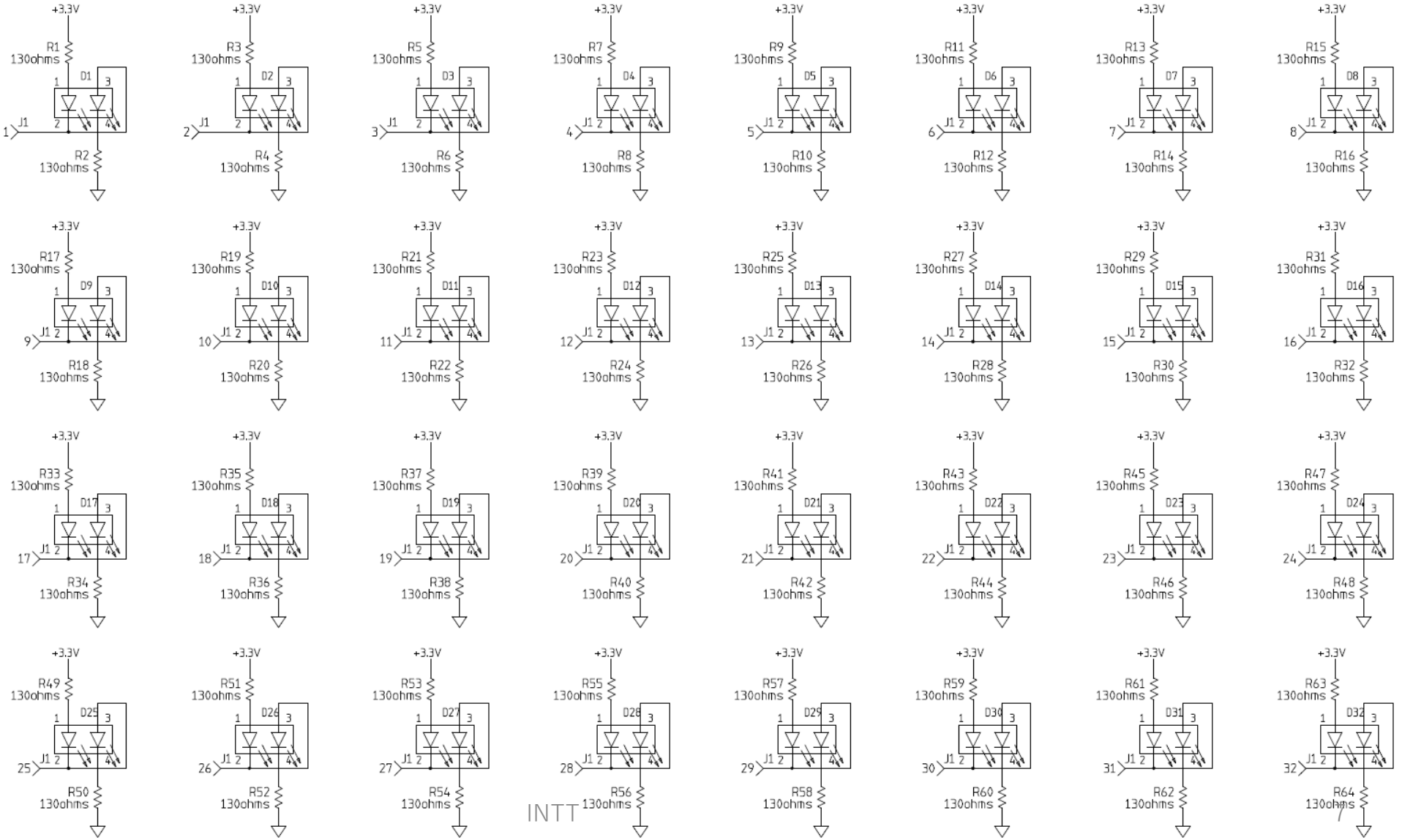
LUT4への接続がFLIPしているため、
4ビット毎にFLIPした条件で検出

```
COND_1_0_LUT : LUT4  generic map (INIT => X"0080") ^M
port map (
      O => COND_1_0,   -- LUT general output
      I0 => DATA_3BUF(15), -- LUT input
      I1 => DATA_3BUF(14), -- LUT input
      I2 => DATA_3BUF(13), -- LUT input
      I3 => DATA_3BUF(12)  -- LUT input
);^M
^M
```

LED基板のコネクタピンとLED位置の対応



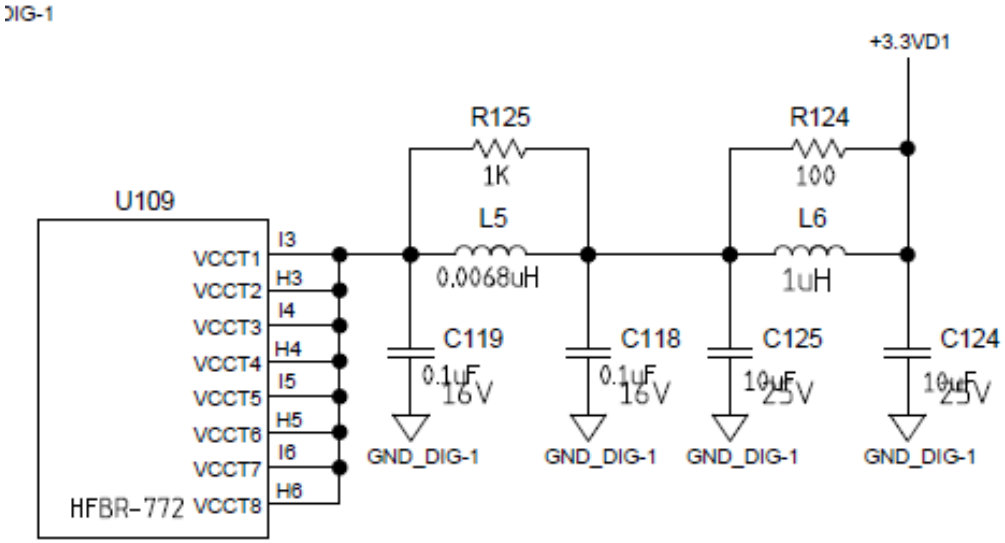


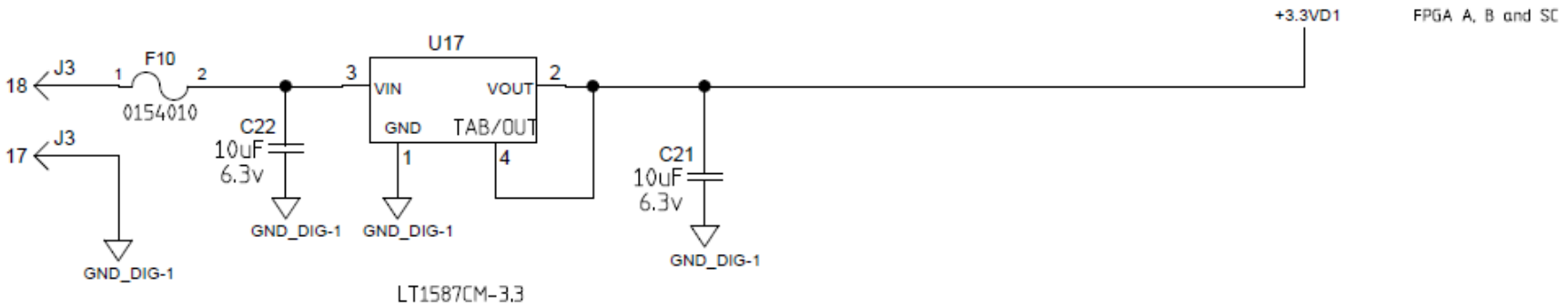


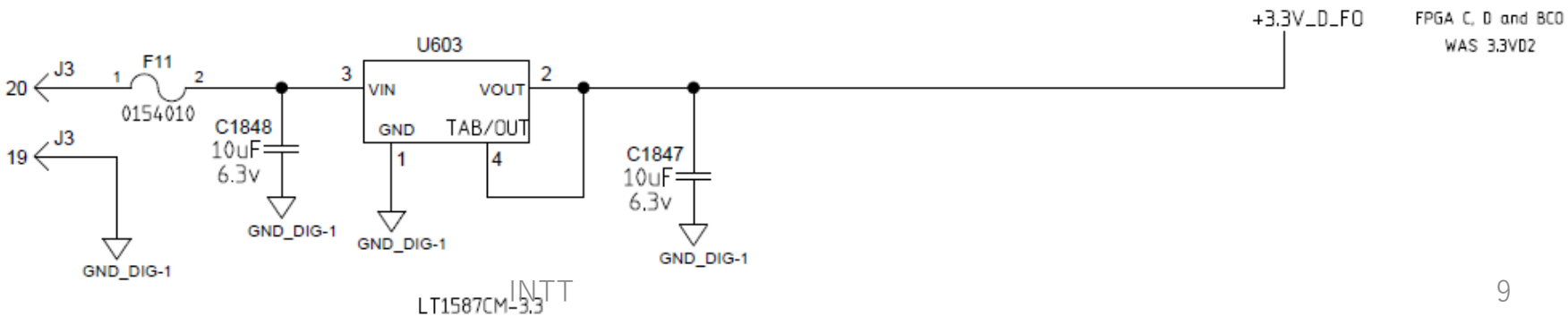
LED基板のコネクタピンとLED位置の対応

コネクタピン	LED番号	コネクタピン	LED番号	コネクタピン	LED番号	コネクタピン	LED番号	コネクタピン	LED番号
1	D1	9	D9	17	D17	25	D25	33	+3.3V
2	D2	10	D10	18	D18	26	D26	34	+3.3V
3	D3	11	D11	19	D19	27	D27	35	+3.3V
4	D4	12	D12	20	D20	28	D28	36	+3.3V
5	D5	13	D13	21	D21	29	D29	37	GND
6	D6	14	D14	22	D22	30	D30	38	GND
7	D7	15	D15	23	D23	31	D31	39	GND
8	D8	16	D16	24	D24	32	D32	40	GND

LED基板の電源


DIG-1
+3.3VD1
R125
1K
R124
100
U109
L5
0.0068uH
L6
1uH
VCCT1
VCCT2
VCCT3
VCCT4
VCCT5
VCCT6
VCCT7
VCCT8
HFBR-772
I3
H3
I4
H4
I5
H5
I6
H6
C119
0.1uF
16V
C118
0.1uF
16V
C125
10uF
25V
C124
10uF
25V
GND_DIG-1
GND_DIG-1
GND_DIG-1
GND_DIG-1


+3.3VD1
FPGA A, B and SC
U17
18
J3
F10
0154010
17
J3
C22
10uF
6.3v
VIN
VOUT
GND
TAB/OUT
C21
10uF
6.3v
GND_DIG-1
GND_DIG-1
GND_DIG-1
GND_DIG-1
LT1587CM-3.3

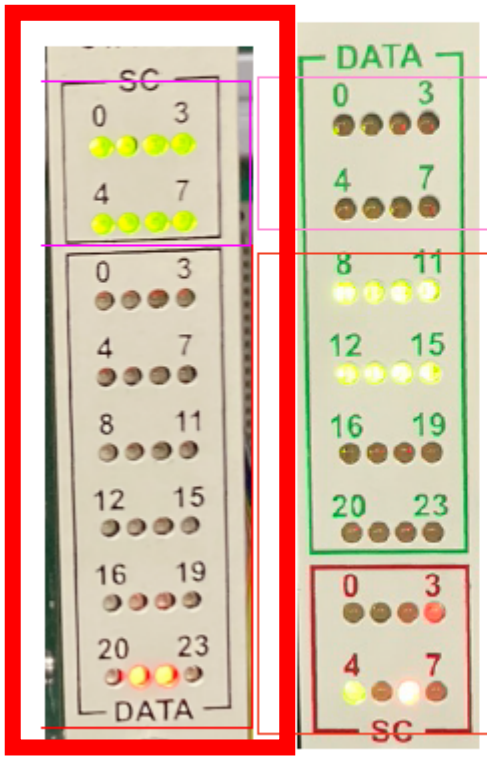

+3.3V_D_FO
FPGA C, D and BC0
WAS 3.3VD2
U603
20
J3
F11
0154010
19
J3
C1848
10uF
6.3v
VIN
VOUT
GND
TAB/OUT
C1847
10uF
6.3v
GND_DIG-1
GND_DIG-1
GND_DIG-1
GND_DIG-1
LT1587CM-3.3

Detail: FEM LED position and the meaning

Numbers which are written on FEM board panel are **different** from numbers of FPGA codes.

LED_OUT(7)~LED_OUT(0) of FPGA code (SlowControl_FEM.vhd)
=
No.0~7 of DATA on FEM board panel

LED_OUT(23)~LED_OUT(0) of FPGA code (FEM_top.vhd)
=
No.8~23 of DATA and No.0~7 of SC on FEM board panel



SlowControl_FEM.vhd
LED_OUT( )

7	6	5	4
3	2	1	0

FEM_top.vhd
LED_OUT( )

23	22	21	20
19	18	17	16
15	14	13	12
11	10	9	8
7	6	5	4
3	2	1	0

SYNC_OK	COMMAND_VME(2) = '1' and ((FEM_ADDR_VME = FEM_ADDR_REF) or FEM_ADDR_VME = x"F")	COMMAND_VME(1) = '1' and ((FEM_ADDR_VME = FEM_ADDR_REF) or FEM_ADDR_VME = x"F")	COMMAND_VME(0) = '1' and ((FEM_ADDR_VME = FEM_ADDR_REF) or FEM_ADDR_VME = x"F")
FEM_LVL1_DELAY(3 downto 1)	FEM_LVL1_DELAY(3 downto 1)	FEM_LVL1_DELAY(3 downto 1)	FEM_COMB_MODE

Station-1	SYNC_OK_3_3	SYNC_OK_3_2	SYNC_OK_3_1	SYNC_OK_3_0	Station-0
Station-3	SYNC_OK_2_3	SYNC_OK_2_2	SYNC_OK_2_1	SYNC_OK_2_0	Station-2
Station-3	SYNC_OK_1_3	SYNC_OK_1_2	SYNC_OK_1_1	SYNC_OK_1_0	Station-2
Station-1	SYNC_OK_0_3	SYNC_OK_0_2	SYNC_OK_0_1	SYNC_OK_0_0	Station-0
	ENPTY	BUSY	MODE	Z	

bottom

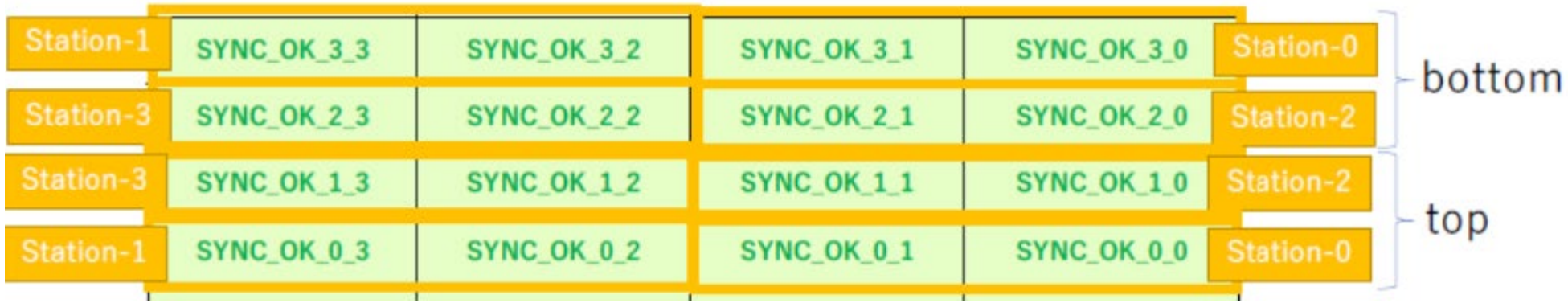
top

ENPTY and BUSY flash when calibration test is running

Codes in previous page is consistent with Mika Shibata's FEM-IB debugging (LED_Instruction.pdf)

接続関係

15 Station-3	14 Station-3	13 Station-2	12 Station-2
11 Station-0	10 Station-0	9 Station-1	8 Station-1



D 8	LED_OUT<15>	SYNC_OK_1_3	DATA_IN_1_1[31:16]	Station3
D 9	LED_OUT<14>	SYNC_OK_1_2	DATA_IN_1_1[15:0]	Station3
D 10	LED_OUT<13>	SYNC_OK_1_1	DATA_IN_1_0[31:16]	Station2
D 11	LED_OUT<12>	SYNC_OK_1_0	DATA_IN_1_0[15:0]	Station2
D 12	LED_OUT<11>	SYNC_OK_0_3	DATA_IN_0_1[31:16]	Station0
D 13	LED_OUT<10>	SYNC_OK_0_2	DATA_IN_0_1[15:0]	Station0
D 14	LED_OUT<9>	SYNC_OK_0_1	DATA_IN_0_0[31:16]	Station1
D 15	LED_OUT<8>	SYNC_OK_0_0 INTT	DATA_IN_0_0[15:0]	Station1

接続関係

15 Station-3	14 Station-3	13 Station-2	12 Station-2
11 Station-0	10 Station-0	9 Station-1	8 Station-1

BottomファイバーのLED

FEM_DATA_Fiber_Mappingと柴田さんの予想

LED位置				
	23 Station-1	22 Station-1	21 Station-0	20 Station-0
	19 Station-3	18 Station-3	17 Station-2	16 Station-2

今井さんの観測*

23 Station-2	22 Station-2	21 Station-3	20 Station-3
19 Station-0	18 Station-0	17 Station-1	16 Station-1

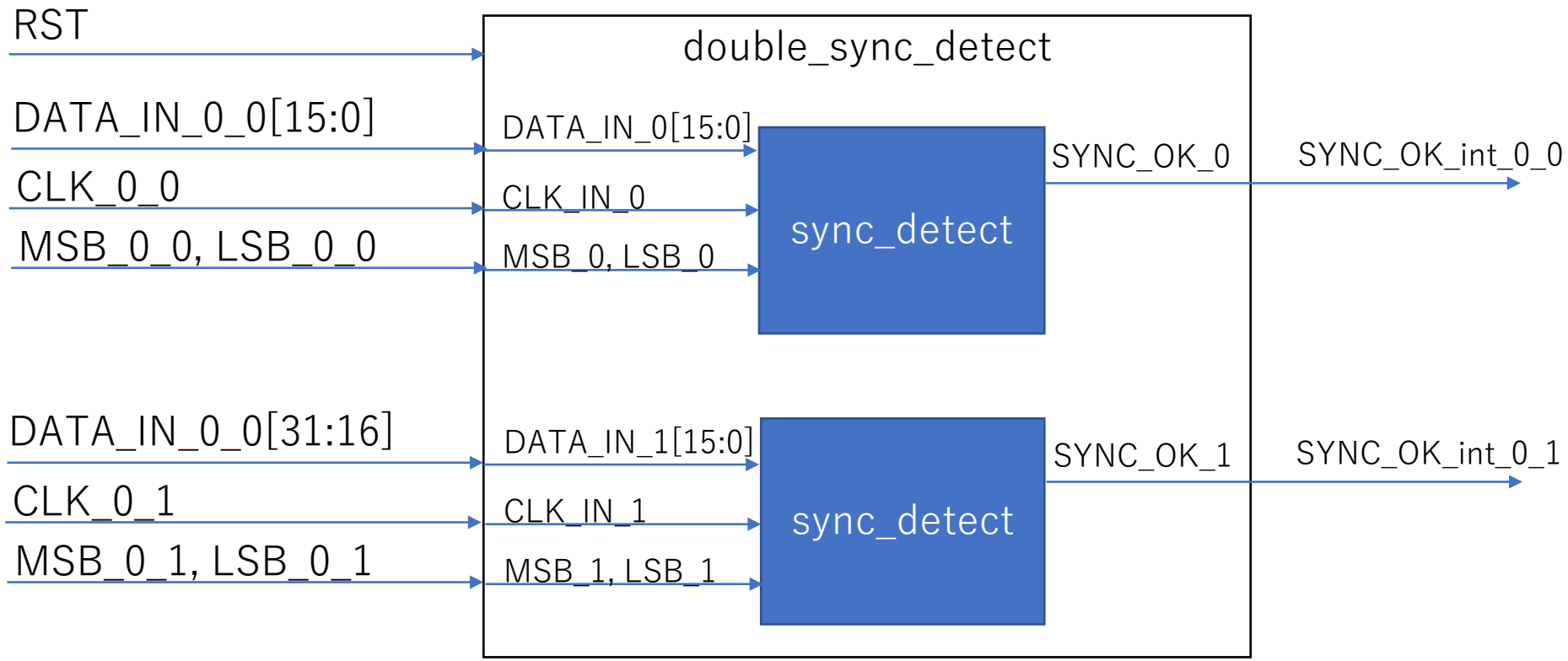
*LEDの不点灯時に、どのポートのキャリブレーションが失敗するかで相関をとった。

D 0	9	9	32	LED[22]	AG32	LED_OUT<23>	SYNC_OK_3_3	DATA_IN_3_1[31:16]	
D 1	10	10	31	LED[23]	AG33	LED_OUT<22>	SYNC_OK_3_2	DATA_IN_3_1[15:0]	
D 2	11	11	30	LED[4]	F33	LED_OUT<21>	SYNC_OK_3_1	DATA_IN_3_0[31:16]	
D 3	12	12	29	LED[10]	J34	LED_OUT<20>	SYNC_OK_3_0	DATA_IN_3_0[15:0]	
D 4	13	13	28	LED[0]	C33	LED_OUT<19>	SYNC_OK_2_3	DATA_IN_2_1[31:16]	
D 5	14	14	27	LED[20]	AF33	LED_OUT<18>	SYNC_OK_2_2	DATA_IN_2_1[15:0]	
D 6	15	15	26	LED[6]	G32	LED_OUT<17>	SYNC_OK_2_1	DATA_IN_2_0[31:16]	
D 7	16	16	25	LED[16]	AC34	LED_OUT<16>	SYNC_OK_2_0	DATA_IN_2_0[15:0]	
D 8	17	17	24	LED[5]	F34	LED_OUT<15>	SYNC_OK_1_3	DATA_IN_1_1[31:16]	Station3
D 9	18	18	23	LED[9]	H34	LED_OUT<14>	SYNC_OK_1_2	DATA_IN_1_1[15:0]	Station3
D 10	19	19	22	LED[19]	AE34	LED_OUT<13>	SYNC_OK_1_1	DATA_IN_1_0[31:16]	Station2
D 11	20	20	21	LED[13]	K33	LED_OUT<12>	SYNC_OK_1_0	DATA_IN_1_0[15:0]	Station2
D 12	21	21	20	LED[8]	H33	LED_OUT<11>	SYNC_OK_0_3	DATA_IN_0_1[31:16]	Station0
D 13	22	22	19	LED[18]	AE33	LED_OUT<10>	SYNC_OK_0_2	DATA_IN_0_1[15:0]	Station0
D 14	23	23	18	LED[15]	L34	LED_OUT<9>	SYNC_OK_0_1	DATA_IN_0_0[31:16]	Station1
D 15	24	24	17	LED[2]	D34	LED_OUT<8>	SYNC_OK_0_0	DATA_IN_0_0[15:0]	Station1

FEM SC-DATA FPGA間のAUXライン

FEM回路	DATA_FPGApin	DATA_FPGA	SC_FPGApin	SC_FPGA
Data_aux_sc[0]	M1	DATA_AUX_TO_SC_FPGA[0]	G2	DATA_AUX_TO_SC<0>
Data_aux_sc[1]	L1	DATA_AUX_TO_SC_FPGA[1]	J1	CS_FROM_MAIN
Data_aux_sc[2]	K1	DATA_AUX_TO_SC_FPGA[2]	J2	CS_SC_FEM
Data_aux_sc[3]	K2	DATA_AUX_TO_SC_FPGA[3]	L3	DO_SC_FEM
Data_aux_sc[4]	J1	DATA_AUX_TO_SC_FPGA[4]	L4	DI_SC_FEM
Data_aux_sc[5]	J2	DATA_AUX_TO_SC_FPGA[5]	L5	SCK_SC_FEM
Data_aux_sc[6]	M2	FEM_ID[0]	M1	DATA_AUX_FROM_SC[8]
Data_aux_sc[7]	M3	FEM_ID[1]	M2	DATA_AUX_FROM_SC[9]
Data_aux_sc[8]	G2	DATA_AUX_FROM_SC_FPGA[0]	G14	DATA_AUX_FROM_SC[0]
Data_aux_sc[9]	G3	DATA_AUX_FROM_SC_FPGA[1]	C16	DATA_AUX_FROM_SC[1]
Data_aux_sc[10]	G1	DATA_AUX_FROM_SC_FPGA[2]	B16	DATA_AUX_FROM_SC[2]
Data_aux_sc[11]	F1	DATA_AUX_FROM_SC_FPGA[3]	B14	DATA_AUX_FROM_SC[3]
Data_aux_sc[12]	E1	DATA_AUX_FROM_SC_FPGA[4]	A9	DATA_AUX_FROM_SC[4]
Data_aux_sc[13]	D1	DATA_AUX_FROM_SC_FPGA[5]	C8	DATA_AUX_FROM_SC[5]
Data_aux_sc[14]	D2	DATA_AUX_FROM_SC_FPGA[6]	D8	DATA_AUX_FROM_SC[6]
Data_aux_sc[15]	C2	DATA_AUX_FROM_SC_FPGA[7]	B7	DATA_AUX_FROM_SC[7]
FEM_CS	F14	FEM_ID[2]	G16	DATA_AUX_FROM_SC[10]
FEM_DI	K14	FEM_ID[3]	G15	DATA_AUX_FROM_SC[11]

FEMのSyncBlock

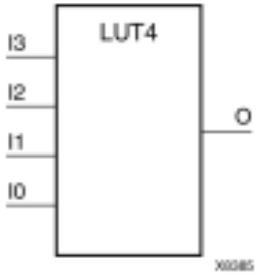


SYNC_OK_int_0_0などが

LUT4の動き

LUT4

プリミティブ：




```
LUT4_inst : LUT4
generic map (
   INIT => X"0000")
port map (
   O => O,    -- LUT general output
   I0 => I0, -- LUT input
   I1 => I1, -- LUT input
   I2 => I2, -- LUT input
   I3 => I3  -- LUT input
);
```

論理表

入力				出力
I3	I2	I1	I0	O
0	0	0	0	INIT[0]
0	0	0	1	INIT[1]
0	0	1	0	INIT[2]
0	0	1	1	INIT[3]
0	1	0	0	INIT[4]
0	1	0	1	INIT[5]
0	1	1	0	INIT[6]
0	1	1	1	INIT[7]
1	0	0	0	INIT[8]
1	0	0	1	INIT[9]
1	0	1	0	INIT[10]
1	0	1	1	INIT[11]
1	1	0	0	INIT[12]
1	1	0	1	INIT[13]
1	1	1	0	INIT[14]
1	1	1	1	INIT[15]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値				

入力4ビットで示される数値に対応する出力ビット(16bit)値を出力する

INIT=X”8000”と書くと
入力=15=0b_1111のとき、出力が1、それ以外で0

8b/10bのKコード

- 8b/10b変換
  - データ中にクロックを埋め込んだり、信号波形の鈍りを少なくしたい。
  - 生のデータでは0や1が長く続く場合がある。
    - クロックの埋め込めない。
    - 回路中の寄生容量などに電荷がたまり、応答が遅くなる時がある。
  - これを解決するため、 0や1が長く続かないように、8bを2bit拡張し、0,1を繰り返すようにデータを変換する(8b/10b変換)
- Kコード
  - 8b/10b変換で使われる制御用のコード
  - データから差別化できるシンボルであり、受信機が異なる制御管理に使用できるものです。例えば、復号化の前のバイト境界のアライメント、クロック補正テクニック、チャンネルボンディングによる複数レーン全体でのデータ一貫性などに対する制御です。
  - 8b/10b変換でも、Kコード変換と通常コード(Dコード)変換は変換則がちがうので、区別できる
- アラインメントとカンマ検出
  - 復号化が正しく機能するためには、受信機が符号化データの正しいワード境界を認識する必要があります。受信機は入力されるデータをスキャンして、符号化データの境界を定義する制御符号（8b/10bではアラインメントで使用されるカンマ）を探します。受信機がアラインメント符号を特定したら、符号化データのワード境界の概念ができるため、符号化データを復号化ロジックへ送り込むことができるのです。

KコードとDコードの変換の違い

- 8bを10bに変換するときの変換則が違う
- 例えば0xFE　の場合
- Kコード　：　0b_100001_0111　もしくは、100001_1000
- Dコード　：　0b_011110_0001　もしくは、100001_1110

D30.7	FEh	011110 0001	100001 1110				
EDB	K30.7	EnD Bad packet	1FE	100001	0111	011110	1000

Name	8bit	currentRD-	currentRD+
D00.0	00h	100111 0100	011000 1011
D01.0	01h	011101 0100	100010 1011
D02.0	02h	101101 0100	010010 1011
D03.0	03h	110001 1011	110001 0100
D04.0	04h	110101 0100	001010 1011
D05.0	05h	101001 1011	101001 0100
～			
D30.7	FEh	011110 0001	100001 1110
D31.7	FFh	101011 0001	010100 1110

表1 PCI Expressで使われるKコードの一覧

シンボル	Kコード	意　味	HEX値	送出するパターン（+）		送出するパターン（−）	
				abcdei	fghj	abcdei	fghj
COM	K28.5	Comma	1BC	110000	0101	001111	1010
STP	K27.7	Start TLP	1FB	001001	0111	110110	1010
SDP	K28.2	Start DLLP	15C	110000	1010	001111	0101
END	K29.7	END	1FD	010001	0111	101110	1000
EDB	K30.7	EnD Bad packet	1FE	100001	0111	011110	1000
PAD	K23.7	PAD	1F7	000101	0111	111010	1000
SKP	K28.0	Skip	11C	110000	1011	001111	0100
FTS	K28.1	Fast Training Sequence	13C	110000	0110	001111	1001
IDL	K28.3	Idle	17C	110000	1100	001111	0011

INTTでのKコードの使用

Table 3. Receive Status Signals

RKLSB	RKMSB	DECODED 20-BIT OUTPUT	
0	0	Valid data on RXD0 to RXD7	Valid data RXD8 to RXD15
0	1	Valid data on RXD0 to RXD7	K code on RXD8 to RXD15
1	0	K code on RXD0 to RXD7	Valid data on RXD8 to RXD15
1	1	K code on RXD0 to RXD7	K code on RXD8 to RXD15

Table 1. Transmit Data Controls

TKLSB	TKMSB	16-BIT PARALLEL INPUT	
0	0	Valid data on TXD0 to TXD7	Valid data TXD8 to TXD15
0	1	Valid data on TXD0 to TXD7	K code on TXD8 to TXD15
1	0	K code on TXD0 to TXD7	Valid data on TXD8 to TXD15
1	1	K code on TXD0 to TXD7	K code on TXD8 to TXD15

表1　PCI Express で使われるKコードの一覧

シンボル	Kコード	意　味	HEX値	送出するパターン(+)		送出するパターン(−)	
				abcdei	fghj	abcdei	fghj
COM	K28.5	Comma	1BC	110000	0101	001111	1010
STP	K27.7	Start TLP	1FB	001001	0111	110110	1010
SDP	K28.2	Start DLLP	15C	110000	1010	001111	0101
END	K29.7	END	1FD	010001	0111	101110	1000
EDB	K30.7	EnD Bad packet	1FE	100001	0111	011110	1000
PAD	K23.7	PAD	1F7	000101	0111	111010	1000
SKP	K28.0	Skip	11C	110000	1011	001111	0100
FTS	K28.1	Fast Training Sequence	13C	110000	0110	001111	1001
IDL	K28.3	Idle	17C	110000	1100	001111	0011

- SYNCのReset時にKコードを使用する。(LATCH_FPGAコマンド)
- データ送信時はDコードを使用
- INTTのSYNC判定はDコードのデータを用いている。

INTTでのデータファイバのSYNC方法

- ROC-FEM間のデータファイバのSYNCは、SERDESチップ毎(16ビット)に判定される。
  - 使用中のファイバ数：8本が独立にSYNC判定
- SYNC状態は、FEMのLEDで表示。
  - SYNC_OK=1：LED=緑。　SYNC_OK＝0：LED=OFF
- SYNCは、制御GUIのLATCH_FPGAボタンで行われる
  - LATCHコマンドがスローコントロール経由でROCに送られる。
  - 詳細な制御は次のページ。
  - SYNC時の手順
    - SYNC_OFF(K30.7) -> Alignment(CommaDetect)x2 -> SYNC判定ワード　3つ

CommaDetectについて

8.3.12 Comma Detect and 8-Bit/10-Bit Decoding

The TLK2711-SP has two parallel 8-bit/10-bit decode circuits. Each 8-bit/10-bit decoder converts 10-bit encoded data (half of the 20-bit received word) back into 8 bits. The comma-detect circuit is designed to provide for byte synchronization to an 8-bit/10-bit transmission code. When parallel data is clocked into a parallel-to-serial converter, the byte boundary that was associated with the parallel data is now lost in the serialization of the data. When the serial data is received and converted to parallel format again, a method is needed to recognize the byte boundary. Typically, this is accomplished through the use of a synchronization pattern. This is typically a unique pattern of 1s and 0s that either cannot occur as part of valid data or is a pattern that repeats at defined intervals. The 8-bit/10-bit encoding contains a character called the comma (b0011111 or b1100000), which is used by the comma-detect circuit on the TLK2711-SP to align the received serial data back to its original byte boundary. The decoder detects the comma, generating a synchronization signal aligning the data to their 10-bit boundaries for decoding; the comma is mapped into the LSB. The decoder then converts the data back into 8-bit data. The output from the two decoders is latched into the 16-bit register synchronized to the recovered parallel data clock (RXCLK) and output valid on the rising edge of the RXCLK.

NOTE

The TLK2711-SP only achieves byte alignment on the 0011111 comma.

Decoding provides two additional status signals, RKLSB and RKMSB. When RKLSB is asserted, an 8-bit/10-bit K code is received and the specific K code is presented on the data bits RXD0 to RXD7; otherwise, an 8-bit/10-bit D code is received. When RKMSB is asserted, an 8-bit/10-bit K code is received and the specific K-code is presented on data bits RXD8 to RXD15; otherwise, an 8-bit/10-bit D code is received (see Table 3). The valid K codes the TLK2711-SP; decodes are provided in Table 4. An error detected on either byte, including K codes not in Table 4, causes that byte only to indicate a K0.0 code on the RKxSB and associated data pins, where K0.0 is known to be an invalid 8-bit/10-bit code. A loss of input signal causes a K31.7 code to be presented on both bytes, where K31.7 is also known to be an invalid 8-bit/10-bit code.