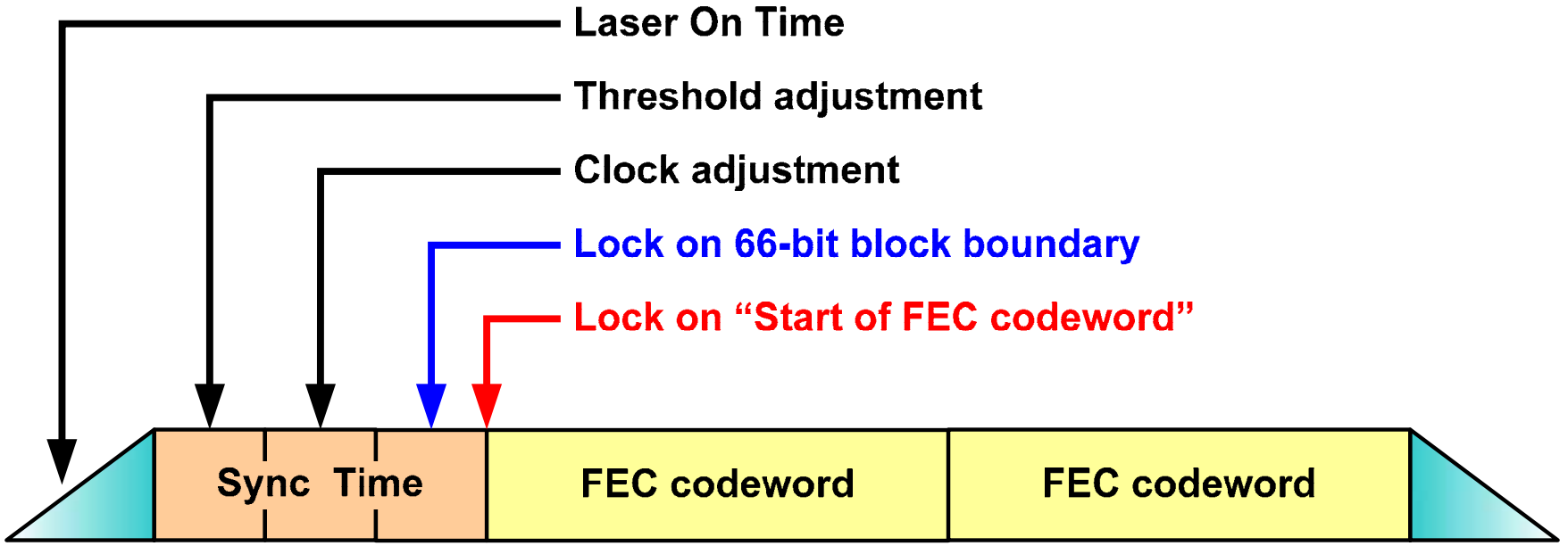
Upstream Clock Synchronization

Glen Kramer
glen.kramer@teknovus.com

Synchronization Issues

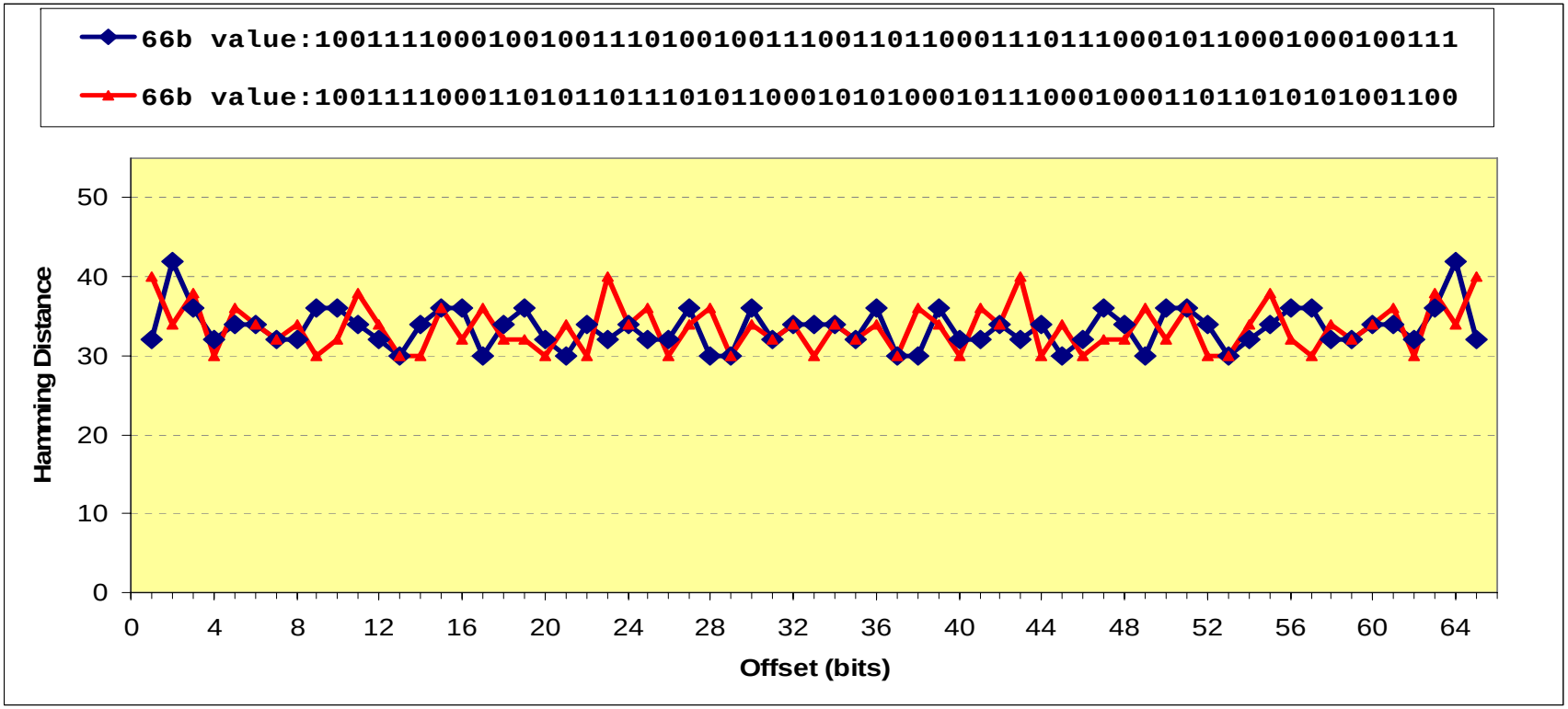
- OLT adjusts gain and recovers clock during special burst preamble (SyncTime)
- OLT cannot use FEC until it knows where FEC codewords start
- OLT should be able to lock on 66-bit block boundary and on FEC codeword boundary on pre-FEC (uncorrected) data



How to Lock on 66-bit Block Boundary?

- For sync time, use special 66-bit pattern with low auto-correlation, s.t. any cyclic shift by n bits (n= 1..65) results in many errors

- Different DC-balanced 66-bit patterns exist that provide minimum Hamming distance = 30 for any bit shift



Example of Sync Pattern

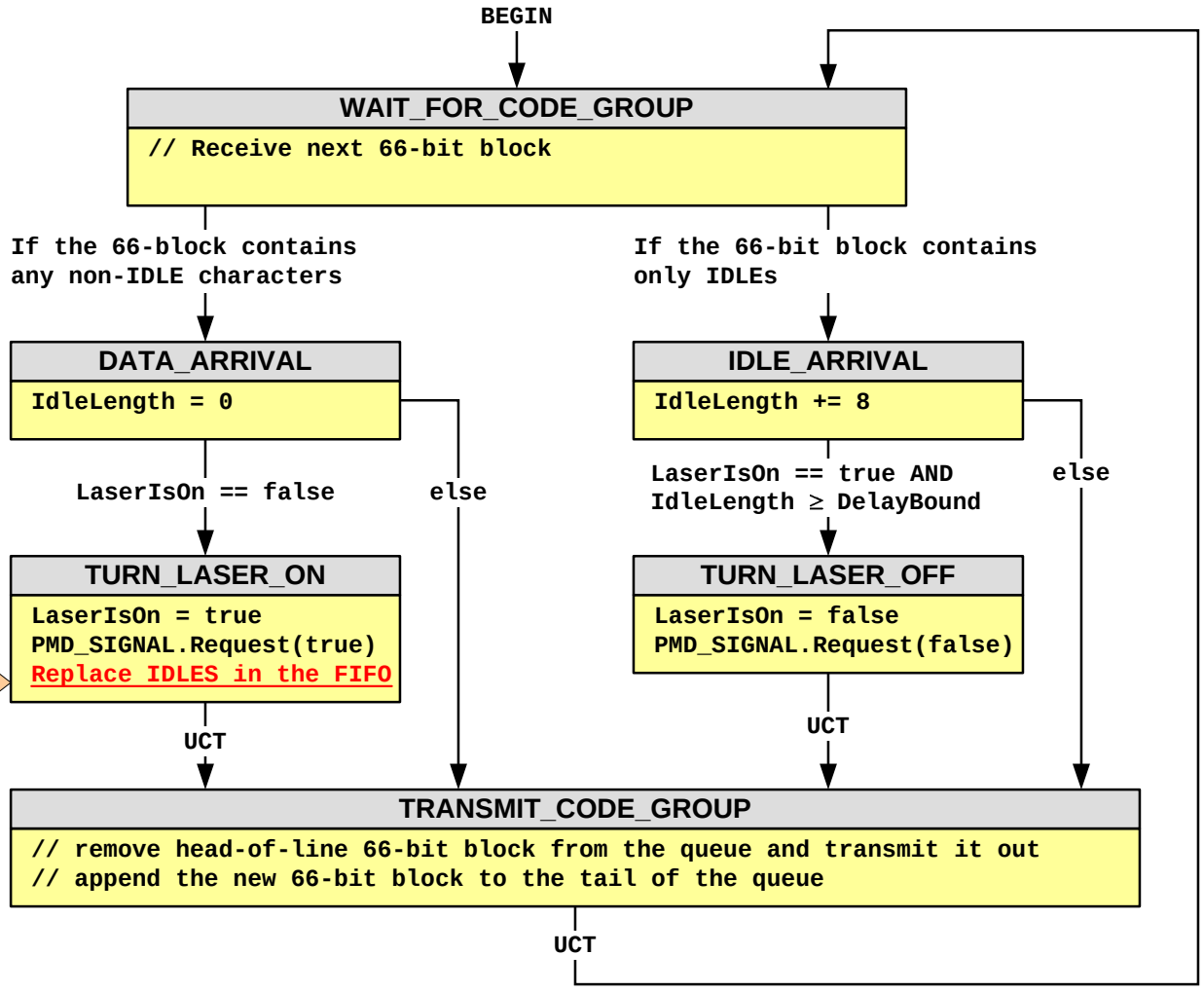
Input Data (first RS transfer / second RS transfer)	Sync [0]	Sync [1]	Bit fields [2]							[65]
$D_0D_1D_2D_3/D_4D_5D_6D_7$	0	1	D_0 [0] [7]	D_1 [0] [7]	D_2 [0] [7]	D_3 [0] [7]	D_4 [0] [7]	D_5 [0] [7]	D_6 [0] [7]	D_7 [0] [7]
$Z_0Z_1Z_2Z_3/Z_4Z_5Z_6Z_7$	1	0	0x1e "01111000"	C_0 [0] [6]	C_1 [0] [6]	C_2 [0] [6]	C_3 [0] [6]	C_4 [0] [6]	C_5 [0] [6]	C_6 [0] [6]	C_7 [0] [6]
$Z_0Z_1Z_2Z_3/S_4D_5D_6D_7$	1	0	0x33	C_0	C_1	C_2	C_3		D_5	D_6	D_7
$S_0D_1D_2D_3/D_4D_5D_6D_7$	1	0	0x78	D_1	D_2	D_3	D_4	D_5	D_6	D_7	
$T_0Z_1Z_2Z_3/Z_4Z_5Z_6Z_7$	1	0	0x87		C_1	C_2	C_3	C_4	C_5	C_6	C_7
$D_0T_1Z_2Z_3/Z_4Z_5Z_6Z_7$	1	0	0x99	D_0		C_2	C_3	C_4	C_5	C_6	C_7
$D_0D_1T_2Z_3/Z_4Z_5Z_6Z_7$	1	0	0xaa	D_0	D_1		C_3	C_4	C_5	C_6	C_7
$D_0D_1D_2T_3/Z_4Z_5Z_6Z_7$	1	0	0xb4	D_0	D_1	D_2		C_4	C_5	C_6	C_7
$D_0D_1D_2D_3/T_4Z_5Z_6Z_7$	1	0	0xcc	D_0	D_1	D_2	D_3		C_5	C_6	C_7
$D_0D_1D_2D_3/D_4T_5Z_6Z_7$	1	0	0xd2	D_0	D_1	D_2	D_3	D_4		C_6	C_7
$D_0D_1D_2D_3/D_4D_5T_6Z_7$	1	0	0xe1	D_0	D_1	D_2	D_3	D_4	D_5		C_7
$D_0D_1D_2D_3/D_4D_5D_6T_7$	1	0	0xff	D_0	D_1	D_2	D_3	D_4	D_5	D_6	

$Z_0Z_1Z_2Z_3/Z_4Z_5Z_6Z_7$	1	0	0x1e	0x6b	0x2e	0x23	0x45	0x0e	0x31	0x2b	0x19

- **No reserved values**
- **Maximum run length = 3 (except code field)**
- **DC-balanced**

Generating Sync Pattern

- Generating special sync pattern is trivial



In this state, the laser is being turned on. The data Detector's FIFO contains exactly the number of IDLES sufficient to cover laser_on + SyncTime.

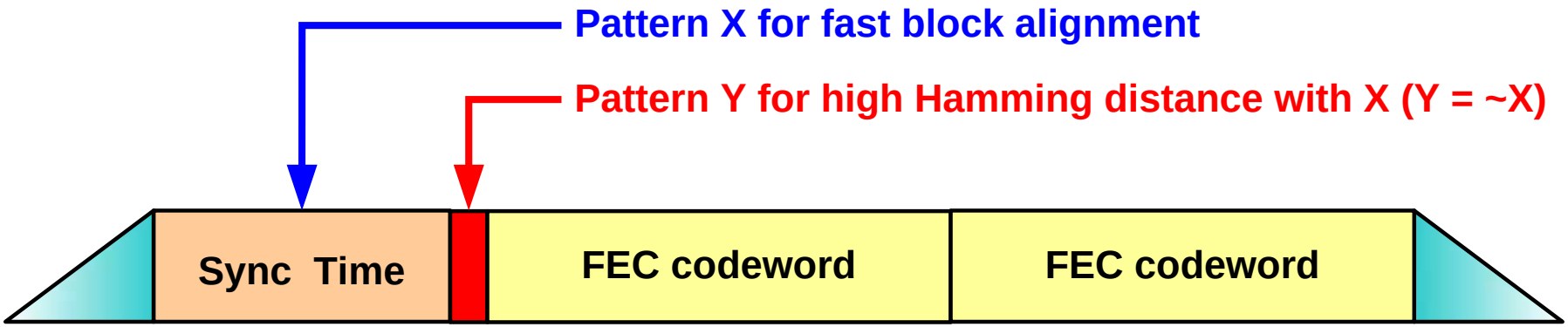
Data Detector can replace all stored IDLEs with a special sync pattern

How to Delineate FEC Codewords?

- The FEC decoder needs to know precisely where FEC codewords start
- It is enough to find the start of the first codeword in the burst. All consecutive codewords will be delineated automatically.
- To find the start of the first codeword, it is enough to find the end of sync sequence

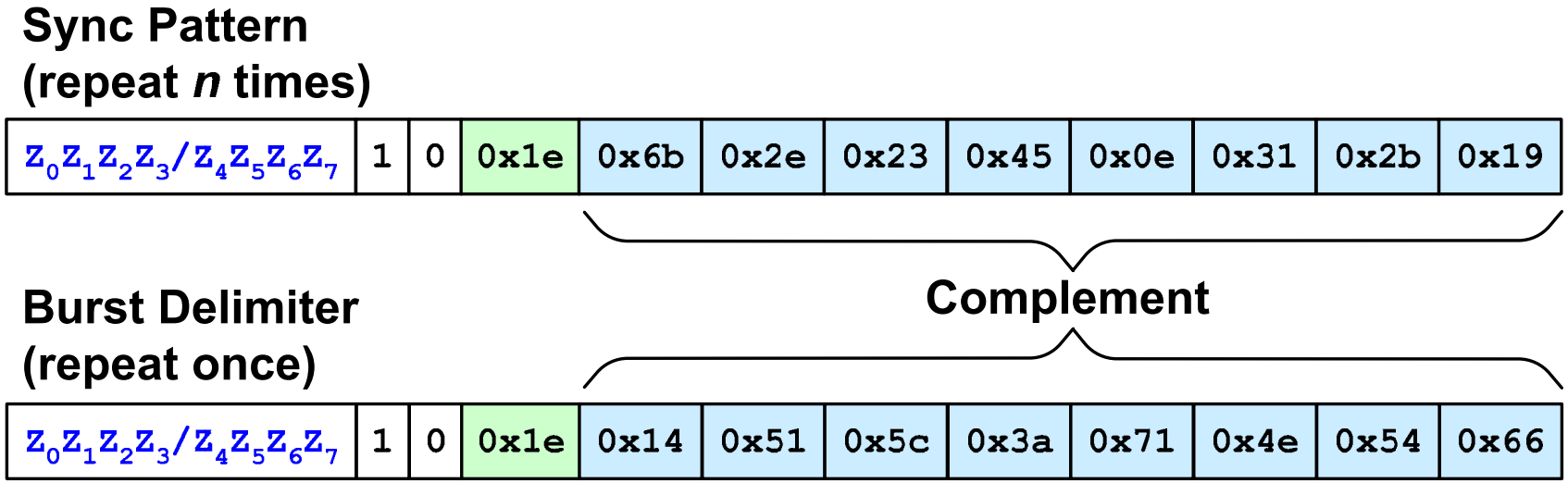
How to Find End of Sync?

- Make the last sync 66-bit block unique (call it burst delimiter).
- Choose the last block to have high Hamming distance with sync pattern
  - For example, burst delimiter = inverted sync pattern
- When burst delimiter is found with less then N bit errors, the receiver knows that the next block will be the beginning of a FEC codeword.



Burst Delimiter Pattern

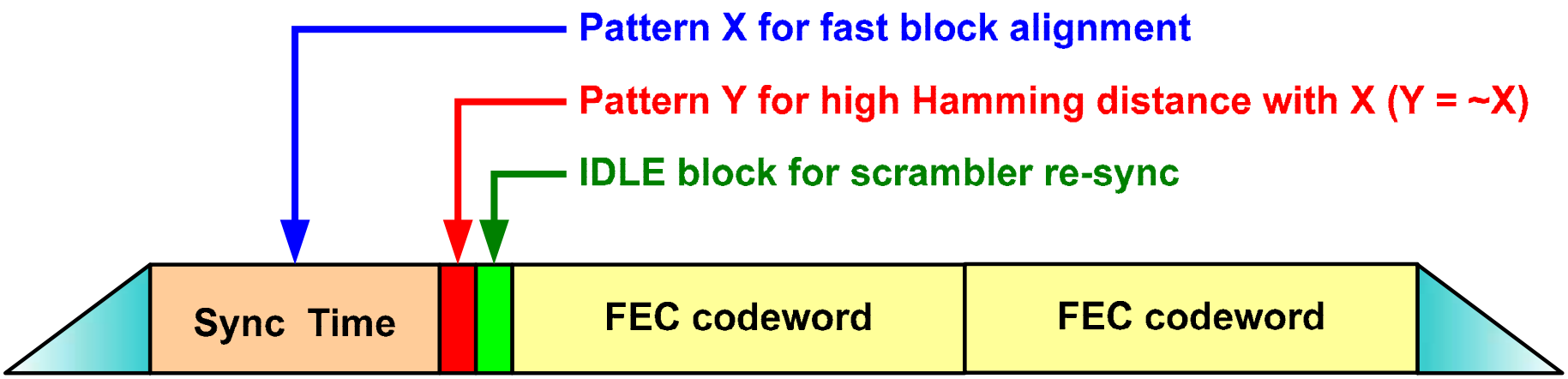
- Burst delimiter pattern should have large Hamming distance from Sync Pattern
- In the following example, the Hamming distance is 56



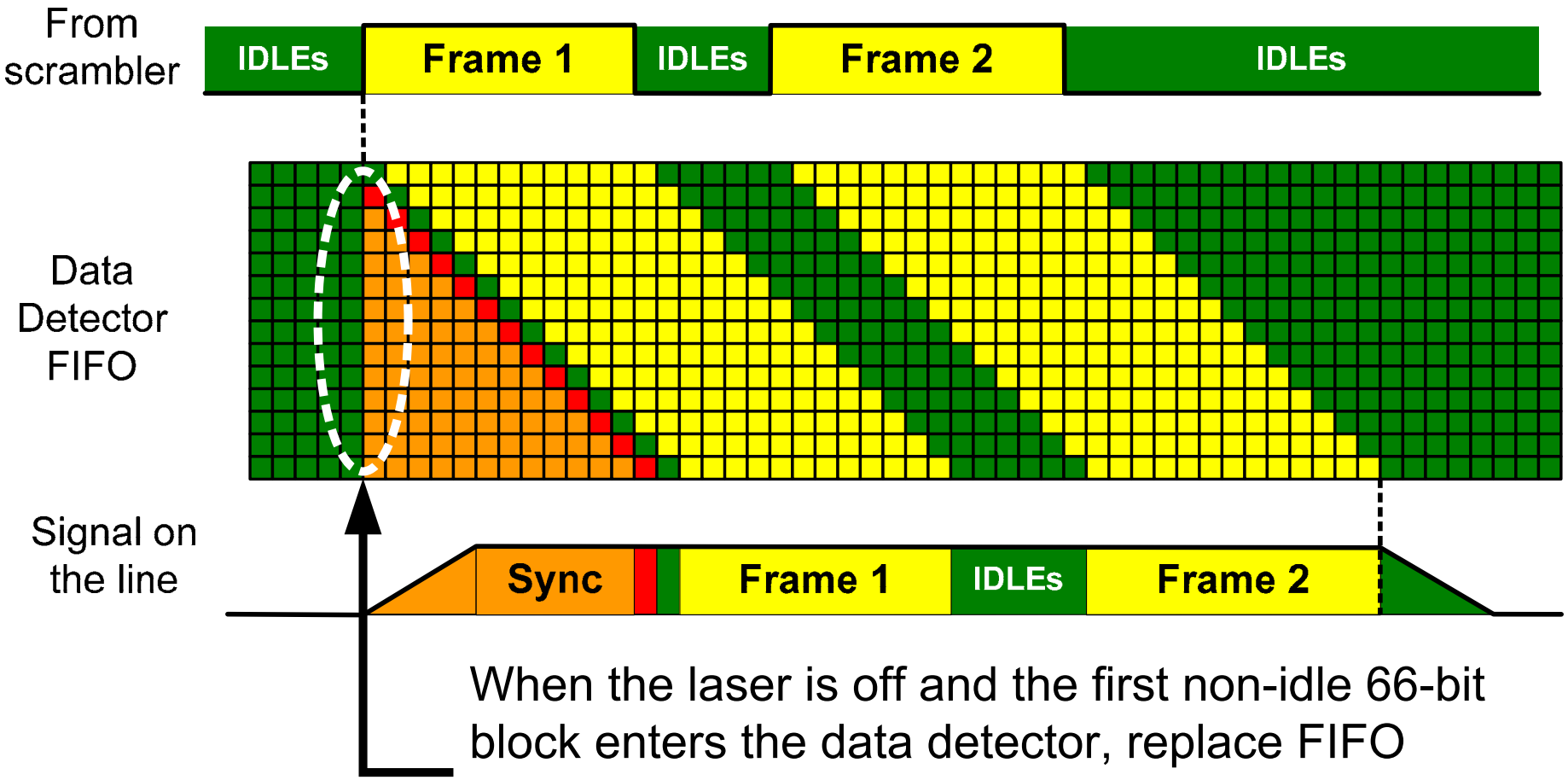
Considerations for Scrambler

- Sync pattern should not be scrambled to guarantee low auto-correlation. Data Detector is below scrambler, so it is OK.

- But the FEC codewords are scrambled. To allow receiver to synchronize the scrambler, there should be 1 scrambled IDLE 66-bit block before the FEC codeword.



Required Data Detector Modifications



When the laser is off and the first non-idle 66-bit block enters the data detector, replace FIFO contents with

sync pattern* + *burst delimiter* + *1 IDLE block

FIFO Replacement Procedure

- The replacement happens just before the first non-idle block is added to the FIFO

- If FIFO contains N 66-bit blocks…

  1. Replace blocks 2..N-1 by Sync Pattern (**X**)
  2. Replace block 1 by Burst Delimiter (**Y**)
  3. Keep block 0 unchanged