# **2D** of Processing and Visualization. Functional Elements **FE.**

## • *FE - **FVF (File-VideoFrame).***

**FVF** is FE that handles buffers of frames $\mathbf{Fr}_{(i,j,k..)}$ of free format **H*V** of double-buffer, that operates at a frame level in an in-sync operation of input-output of **16b** matrixes of images.

**FVF** is heart of parallel (two-phase) operation for all functional elements of the conveyor.

**FVF** consists of following parts:

- The **FVF** core itself – a system of separate control of two blocks of buffer memory with matrix organization, operating for data input/output consequently. The core allows both clock-dependant and clock-independent operations at frame level for data flows with different calculation time in the conveyor.

  Provides synchronous handling of both buffers at line level within the frame stream.

- The double buffer of frames $\mathbf{Fr}_{(i,j,k..)}$ – an image matrix – or the data itself. Each buffer is **32MB SRAM** based with **H*V*16b** format within **1k*16k** up to **16k*1k** range including intermediate format **4k*4k** where values **H** and **V** are bound to **1k<H<16k** and **16k<V<1k** accordingly in $\mathbf{2^n}$ increment for each value.

  The size of a matrix can not be less size of the largest display frame.

- FE **ZoneBlnk** - buffer of frame zone $\mathbf{Fz}_{(i,j,k..)}$ – matrix of one-bit matrixes that describe depth zones (up to 7 zones of interests and 1 zone to set blanking on/of). The pixel value in matrix receives one out of two possible values:

  =1 – shows that the given pixel is displayed, and

  =0 – shows that the given pixel is displayed at given luminosity intensity to decrease visualization zone.

  The buffer is **SRAM** based with the format from **H*V*2b** up to **H*V*8b**, within **1k*16k** up to **16k*1k** range including intermediate format **4k*4k** where values **H** and **V** are bound to **1k<H<16k** and **16k<V<1k** accordingly in $\mathbf{2^n}$ increment for each value.

- Functional unit **IN_LUT16** at core input **FVF** for normalization of data after each development phase performed by previous FE.

## • *FE – **IN_LUT16.***

### *Assignment:*

Functions of input **16b** nonlinear conversion **In_LUT16** include **LUT-**conversions and from hardware point of view represent static memory with raw data at input address the results of of conversion (memory data) at output address:

The following conversion functions are stored in memory of – **In_LUT16** :

- **DICLUT** – extraction of DICOM file, $\mathbf{Fr_j = f_{DICLUT} (Fr_i)}$;
- **LnLUT** - Napierian logarithm $\mathbf{Log_n}$, $\mathbf{Fr_j = f_{LNLUT} (Fr_i)}$;
- **S-LUT** - S-shaped function, $\mathbf{Fr_j = f_{S\text{-}LUT} (Fr_i)}$;
- **NegLUT / PosLUT** - Negative/Positive, $\mathbf{Fr_j = f_{NEGPOS} (Fr_i)}$;
- **ADAPTx** – programmed conversion tasks of conversion (numerical array). $\mathbf{Fr_j = f_{ADAPTx} (Fr_i)}$,

**LUT -** ***Look Up Table*** is an array of **16b** positive integers $2*2^n$ bytes (two bytes on each value **LUT**), where **8<=n<=16** is a quantity of bits per pixel used in image.

The functions **In_LUT16** can be applied sequentially in combination with/without filters and soon, thus performing such processes as image normalization, spectrum adjustment, preliminary calculations for in-between-frames operation (subtraction) etc.

### *Implementation:*

**IN_LUT16** is a functional element with direct **LUT (Look Up Table)** conversions of input data stream to output data converted with functions calculated by management system that has 65536 ($2^{16}$) values of each of the static tables loaded dynamically.

FE **IN_LUT16** is **SRAM 1*64k*16b** up to **8*64k*16b** memory and serves to be loaded with one of **LUT** tables. Load operations are performed by management system of conveyor through **ConfgBUS** (configuration bus).

It is possible to load from 1 to 8 of **IN_LUT16** tables simultaneously (minimum number of tables is 2), with instant useof only one **LUT** table.

**IN_LUT16** is designed to convert/build up already processed image spectrum back to original depth (full **16b**) image.

- *FE – MatrxFil.*

## Performance:

Filters with **3x3** matrix (as a result of multiplication of **16b** / **8b** image matrix by filter matrix) simultaneously build up spectrum of the source frame, its maximum and minimum values, for example: $\mathbf{Fr_j = Fr_i * Fil_i}$, where

$\mathbf{Fr}$ – matrix of positive 16b integers - frame itself,
$\mathbf{Fr_j}$ – intermediate current frame,
$\mathbf{Fr_i}$ – intermediate result of frame conversion,
$\mathbf{Fil_i}$ – currently selected **3x3** filter, where matrix elements are **16b** integers - ***short***.

Spectrum of the image - histogram is an array of 32b of integers - ***integer***.
The size of spectrum array is limited to = **4*(2n +2)** bytes (four bytes for each spectrum value), n - is number of bitper pixel in image.

## Operation:

**MatrxFil** is FE with matrix transformation of an input data obtained synchronously from **FVF** units, i.e. performs multiplying of **16b 3*3** filter matrix by frame matrix.
**MatrxFil** is a set of high-speed parallel **16b** multipliers and **24b** summators.

The crux of this architecture consists in a synchronous operation all FE, including **MatrxFil** provided by single FE **FrSynchr**, that synchronizes all conveyor operations, regardless of the fact that there may occur specially programmed shifts in line/frame at stream processing in FE **MatrxFil**.
Description of calculations performed by FE **MatrxFil** for each pixel in image map:

- Choice of values around of pixel processed $\mathbf{a_{ij}}$ - calculation of **3*3** matrix coordinates for matrix element (calculation of coordinates for 9 elements – for the pixel itself and for neighboring values)

- Summing up elements of filter matrix $S_m = \sum_{j=1,i=1}^{3} (\mathbf{m_{ji}})$, where $\mathbf{m_{ji}}$ are matrix elements

  (the sum of 9 values is performed only once when selecting (changing) values of filter matrix);

- Multiplication of matrixes $P_{KL} = (\mathbf{a_{ij}} * \mathbf{m_{ji}})$ , where $\mathbf{a_{ij}}$ are elements of image matrix
  (Calculation of 9 pair multiplication per each image pixel);

- Summing up values of image matrix $S_p = \sum_{k=1, L=1}^{3} (P_{KL})$, (sum of 9 values);

- Getting new value for each pixel $p_{KL} = S_p / S_m$.

- Compilation of initial image array on the basis of all pixel values of $p_{KL}$ as a result of processing by FE **MatrxFil** where the result is subject for conveyor control system for further alterations, calculations and recording of the newly created array to **IN_LUT16** stored in FE **FVF**, that follows the conveyor.

- Calculation of mixed value for each pixel of value based on the previous and subsequent values of a pixel at filtration operation.
  For the whole frame it will look as follows $\mathbf{Fr_j = k_i * Fr_i + (1-k_i) * Fr_{i\&f}}$.

- Calculation of an intermediate image spectrum of a frame, with maximum and minimum pixel values for subsequent use in **In_LUT16** to calculate **LUT** tables and correction values of an image at input to twin frame buffer.

Required average calculation speed (average pixel input) in **MatrxFil** makes **200-250 Mpixel/sec**.
Actual pixel input speed in **MatrxFil** (synchronous), makes about **160-200 Mpixel/sec** or in terms of instant data stream up to **320-400 MB/sec**. We also explore several considerations aimed to simplify calculations, using symmetry of filter matrix, special division methods with coefficients etc.
It is necessary to have in mind, that the conveyor performs several manipulations (matrix multiplying) sometimes in consequent-parallel mode of operation.

## • FE – LUTConv.

### Assignment:

Functions of nonlinear conversion are achieved by means of **8(10)b** output **Out_LUT8** conversion tables and are preformed by video adapter hardware:

- **Quad –** square function
- **Log$_n$** – Napierian logarithm;
- **Exp** – exponential;
- **Sin** – sinusoidal;
- **Cos** – cosine.

At availability of adapters with **10/12b RGB** or **B&W** output , all **8b** functions of **LUT** conversions are changed to **10/12b** output **LUT**-conversions accordingly.

### Implementation:

**LUTConv** is FE for direct table **LUT** conversion of **16b** output data for visualization through **10/12b** output frame buffers.

FE **LUTConv** is **SRAM** memory based with size from **1*64k*10/12b** up to **8*64k*10/12b**, used to upload **LUT** tables by , conveyor management system

Supports from 1 to 8 **LUTConv** tables simultaneously, with only one **LUT** table used in each instant
For each HR screen output monitor should be used one FE **LUTConv**.

## • FE – AddFIFO.

**AddFIFO** is FE representing stream summator with a **FIFO** instruction elements at one or two inputs of the summator, serving to ensure delays and synchronization for stream processes in parallel computing procedures.
Actually **FIFO** provides delays for indicated number of synchronization clock ticks.

## • FE – BUSComm.

**BUSComm** is FE for commutation of **16b** synchronous data buses, that provides interchange between functional elements and functions as a kind of transport arteries of the conveyor in image processing.

## • FE – FrSynchr.

**FrSynchr** is FE that synchronizes streams of frames (line-frame synchronous generator) and handles functional elements with frame organization – **FVF;** provides frame synchronization of visualization matrix of **H*V** format images.

## • FE – ConfigFE.

**ConfigFE –** is **FE**, that provides general configuring of other FE, sets processing parameters, manages different operations mode of FE, loads conversion function data **IN_LUT16** and **LUTConv**, communicating through **FE BUSComm** to implement architectural structure of systems for 2D processing and visualization.

## • FE - BUS.

**DateBUS** – **16b** one-direction data bus from one FE to another FE.

**FrSynBUS – (Frame Synchronization BUS) –** 28b synchronous bus for addressing **FVF**.

**ConfgBUS – (Configuration BUS) –** bus for configuring and loading parameters and data arrays.