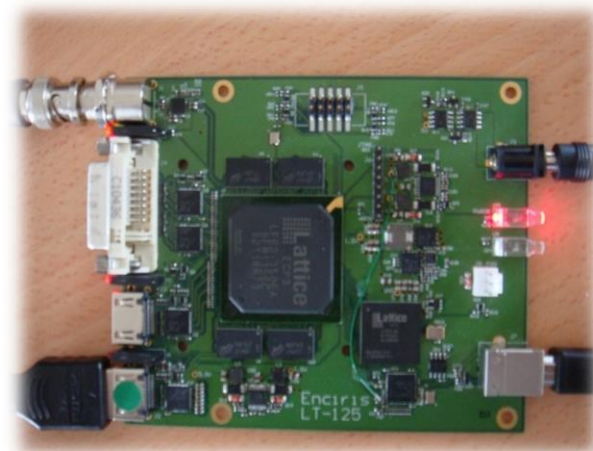


# Video decoding: SDI interface implementation & H.264/AVC bitstream decoder hardware architecture design and implementation

Student Name: Mr. Vicheka PHOR  
Department: CAMSI  
School supervisor: Mr. Jacques JORDA  
Advisor: Mr. Phillip WEISSFLOCH  
Academic year: 2013-2014



# OUTLINE

## I. Objectives

## II. SDI video interface implementation

II.1 Tri-Rate SDI PHY IP Pass-through sample design

II.2 SDI video interface implementation on LT-125 board

## III. H.264/AVC bitstream decoder hardware architecture design and implementation

III.1 H.264/AVC data stream structure

III.2 H.264/AVC bitstream decoder hardware architecture

III.3 FLC, Exp-Golomb, CAVLD decoding

III.4 Development progress

III.5 Tools used in the FPGA development flow design

III.6 Simulation and results

## IV. Conclusion

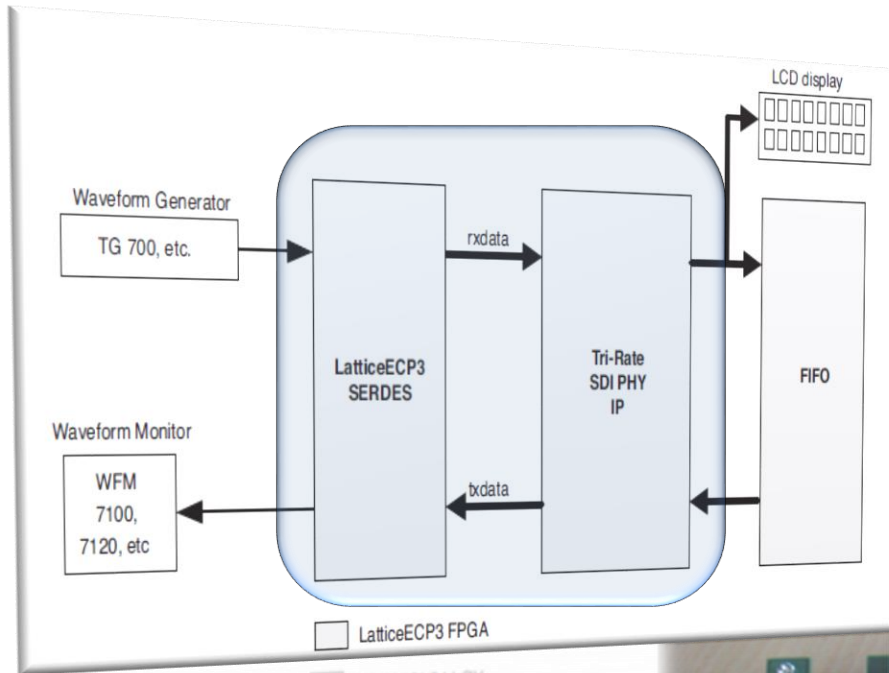
# I. Objectives

- SDI (Serial Digital Interface) video interface implementation:
  - implementation of a Lattice tri-rate SDI PHY IP core on a Lattice FPGA of the Enciris LT-125 board
  
- H.264/AVC bitstream decoder hardware architecture design and implementation

## II. SDI video interface implementation

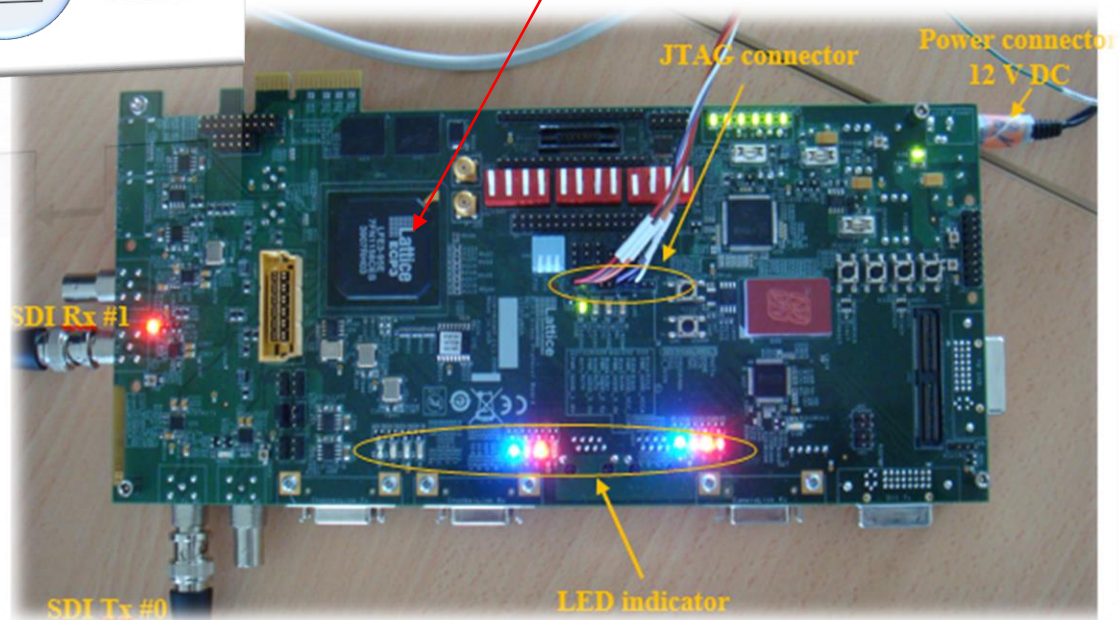
- Objective:
  - 3G/HD/SD SDI video input functionality on LT-125 board
- What was done:
  - Use of Lattice tri-rate SDI PHY IP core
  - Study of the Lattice tri-rate SDI PHY IP Pass-through demo
  - Adding of several blocks: Sync-signals, sdi\_422to444, and YCrCb to RGB converter block

# II.1 Tri-Rate SDI PHY IP Pass-through sample design

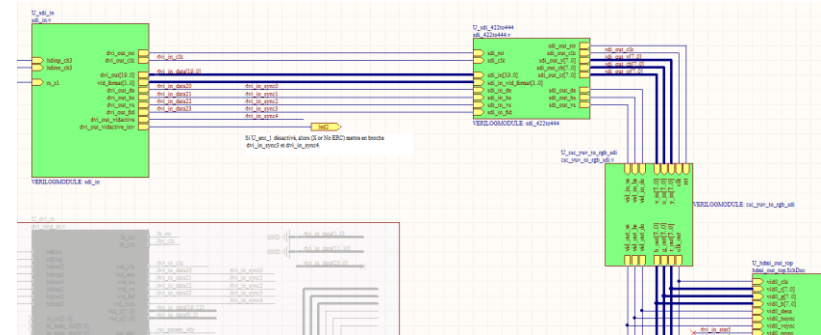
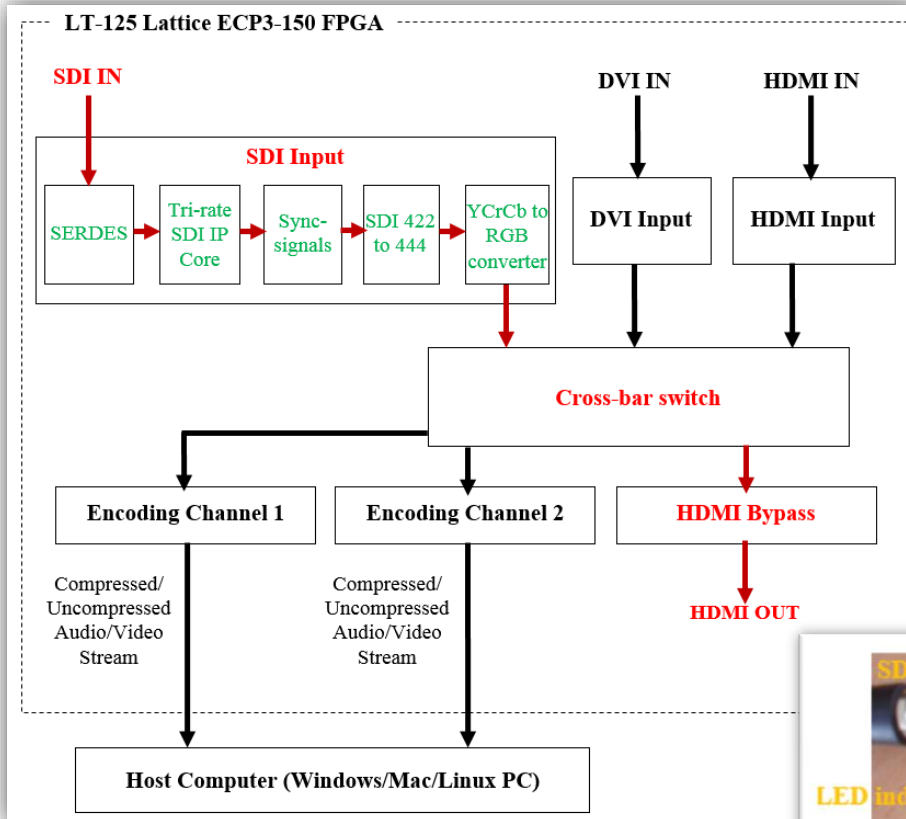


Number of registers:	834 / 71952 (1 %)
Number of SLICES:	896 / 46008 (2 %)
Number of LUT4s:	1425 / 92016 (2 %)

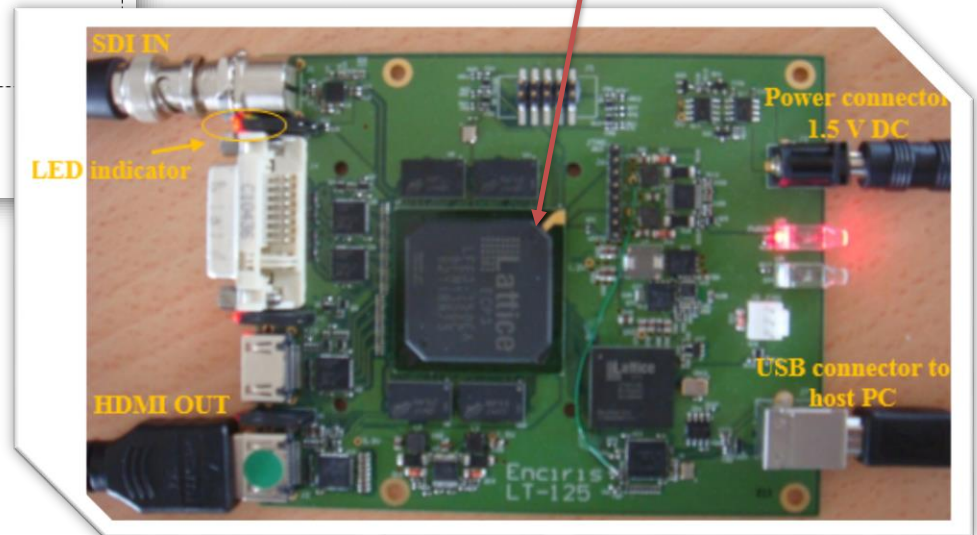
Lattice FPGA\_LFE3-95E-7FN1156CES



# II.2 SDI video interface implementation



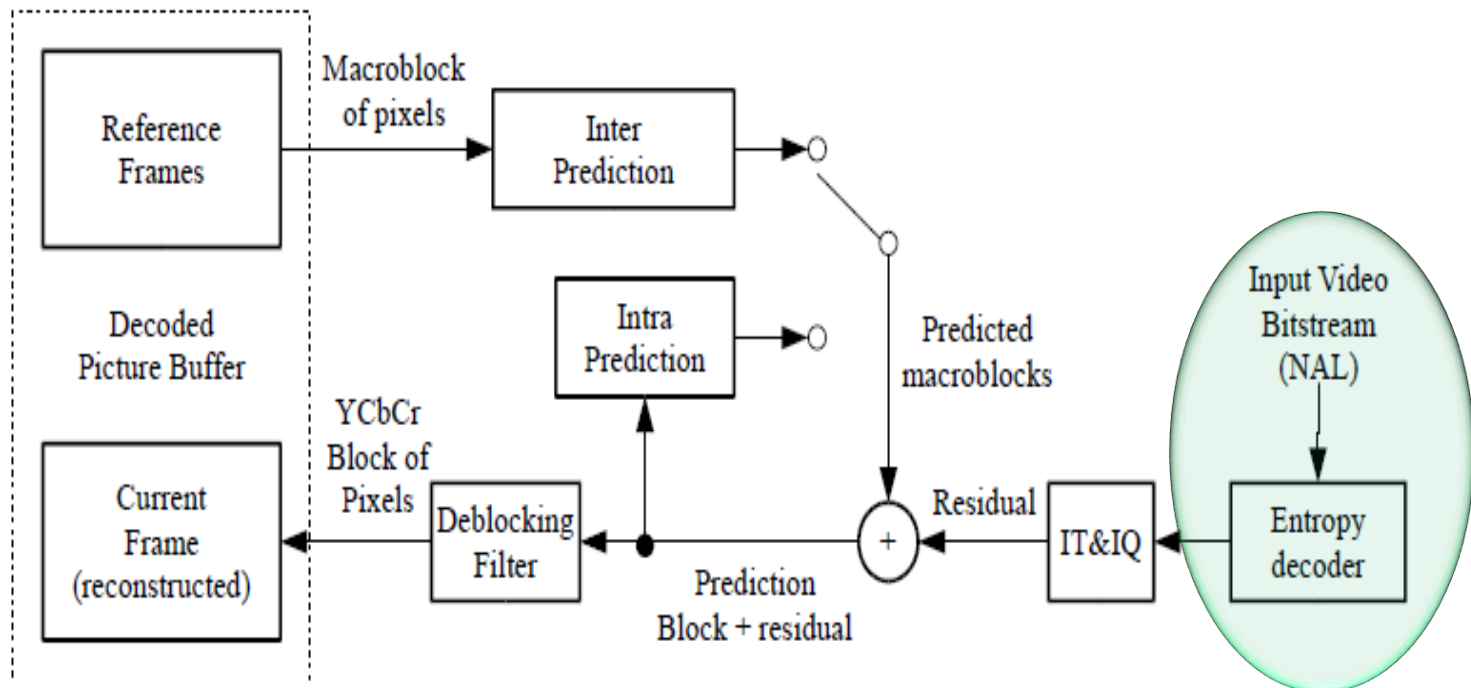
Lattice FPGA\_LFE3-150EA-6FN1156C



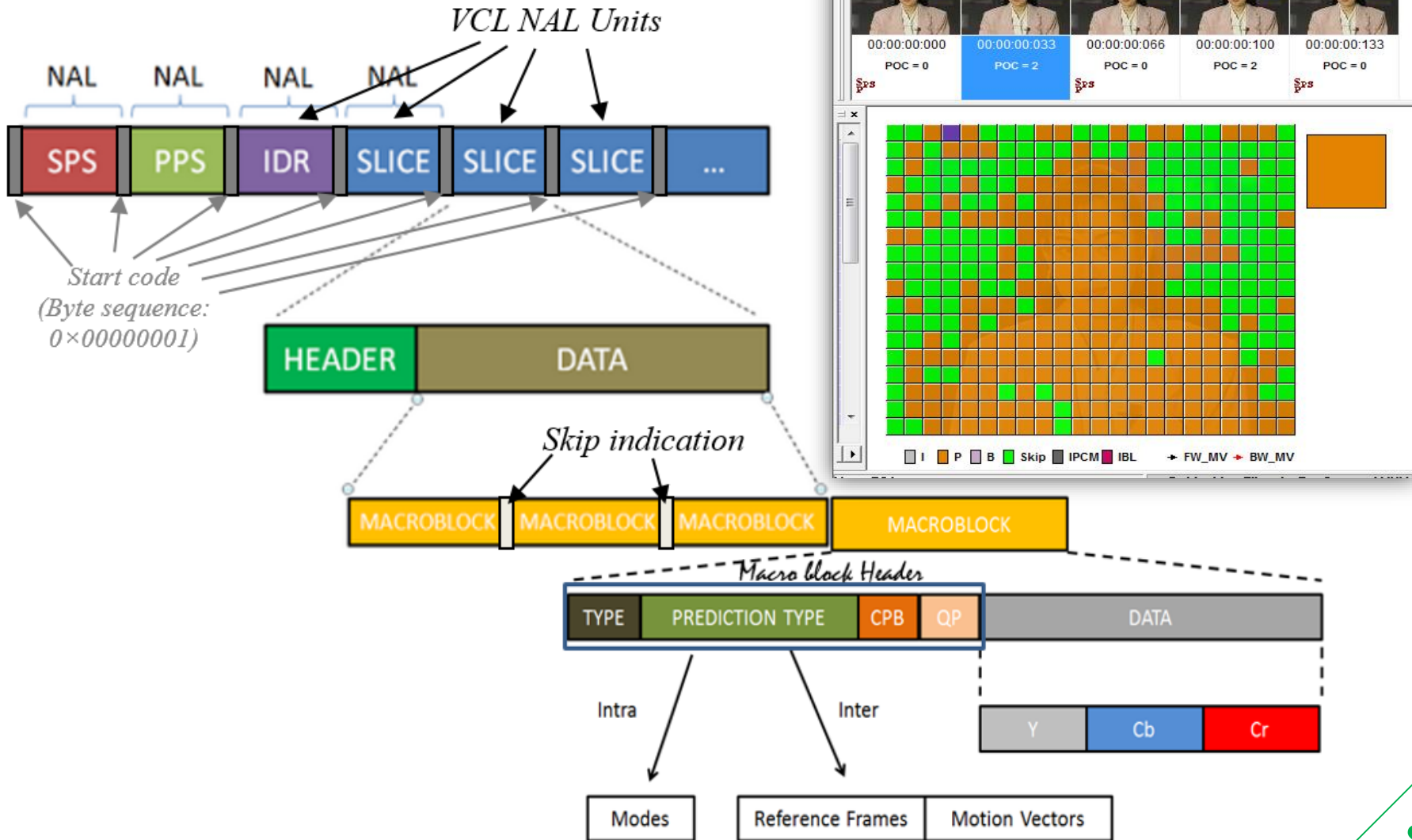
# III. H.264/AVC bitstream decoder hardware architecture design and implementation

## ➤ Objective:

- develop a custom H.264 bitstream decoder for Enciris encoder (high bit rate: 50 Mbit/s; 1080p30)

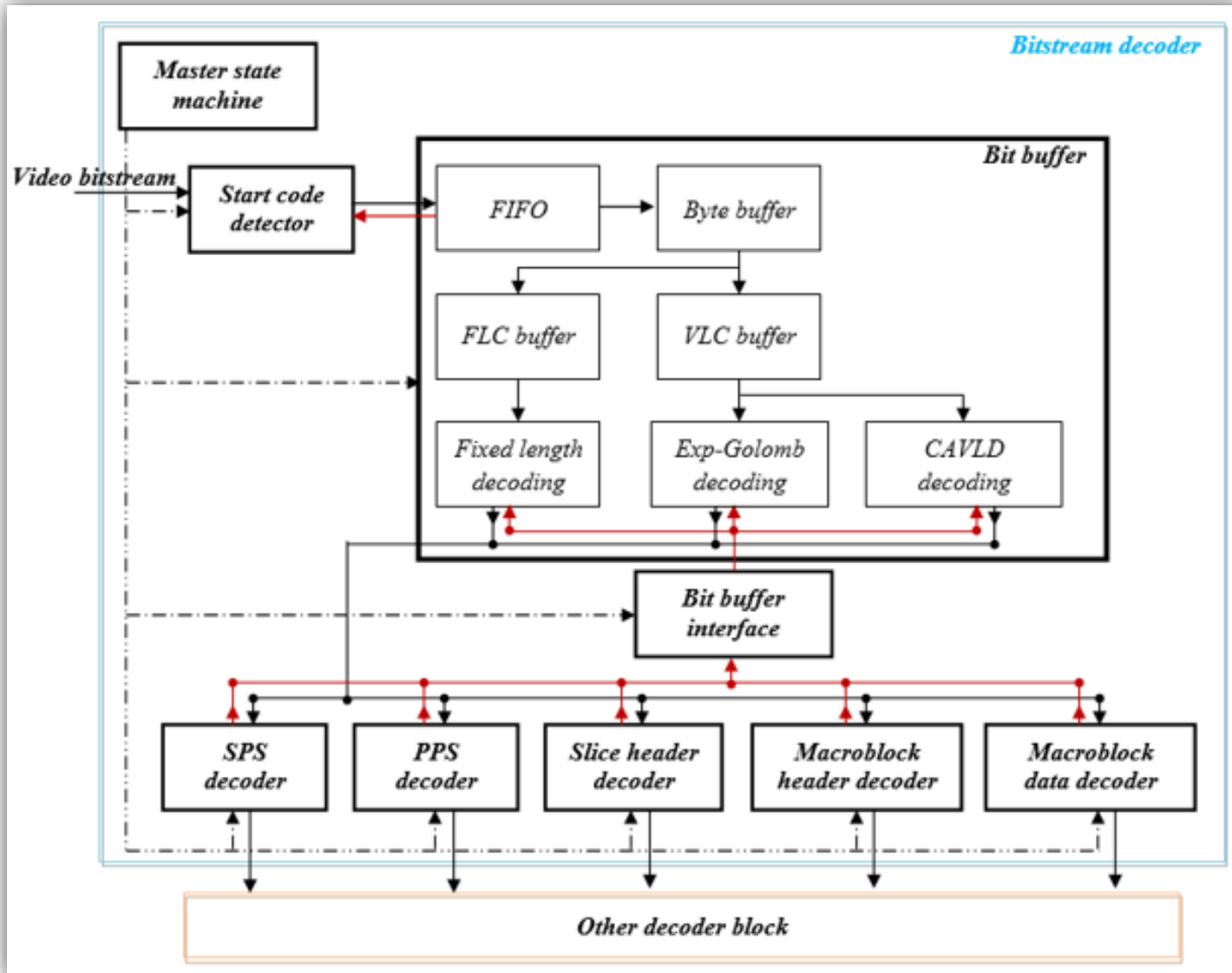


# III.1 H.264/AVC data stream structure





## III.2 H.264/AVC bitstream decoder hardware architecture



## III.3 FLC, Exp-Golomb, CAVLD decoding

FLC (5 bits)	
code_num	Codeword
0	00000
1	00001
2	00010
3	00011
4	00100
5	00101
6	00110
7	00111
8	01000
...	...

Exp-Golomb	
code_num	Codeword
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001
...	...

16 coefficients  $\rightarrow$  24 bits

**CAVLC**

4x4 block:

0	3	-1	0
0	-1	1	0
1	0	0	0
0	0	0	0




000010001110010111101101

24 bits  $\rightarrow$  16 coefficients

**CAVLD**

4x4 block:

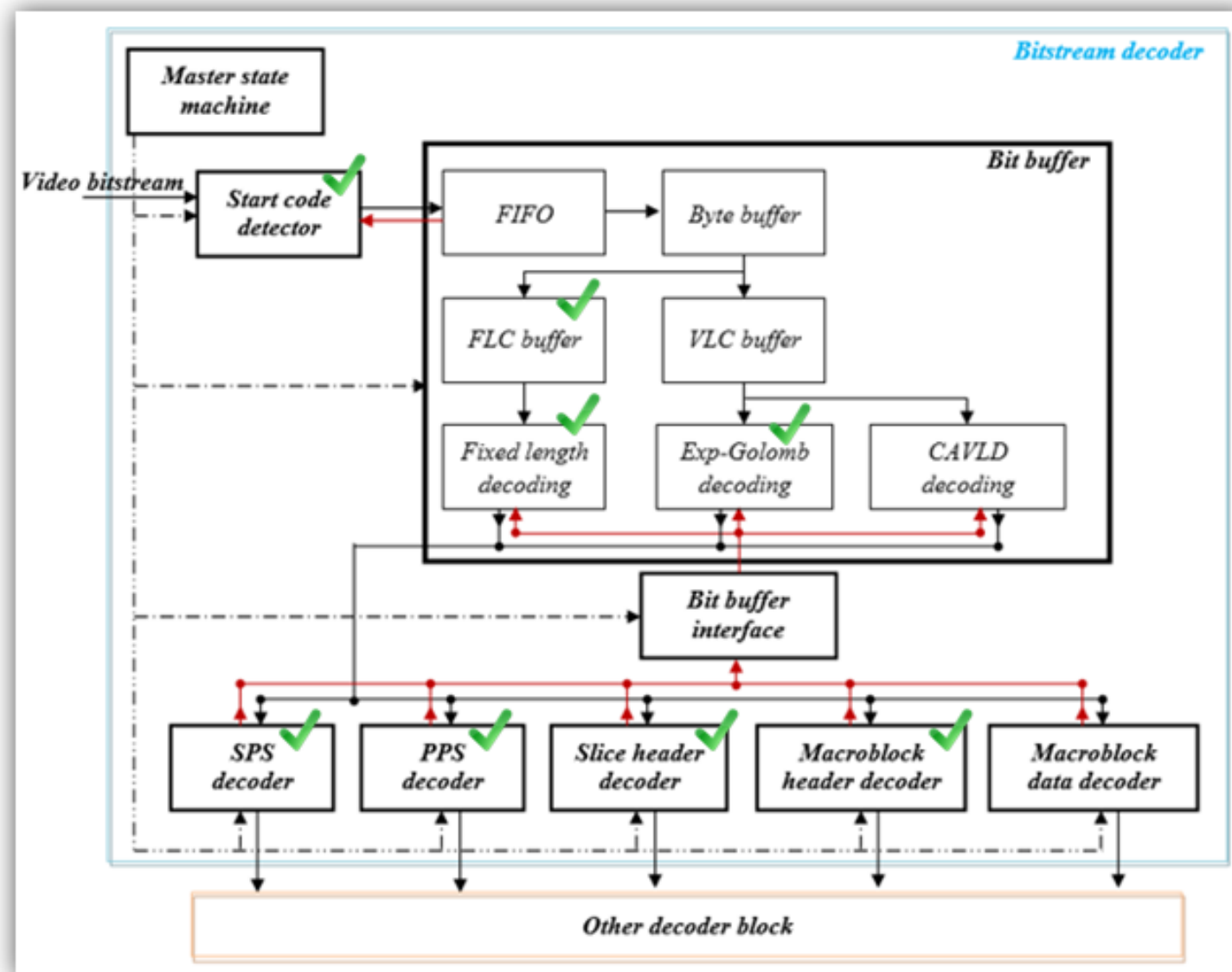
0	3	-1	0
0	-1	1	0
1	0	0	0
0	0	0	0

000010001110010111101101 

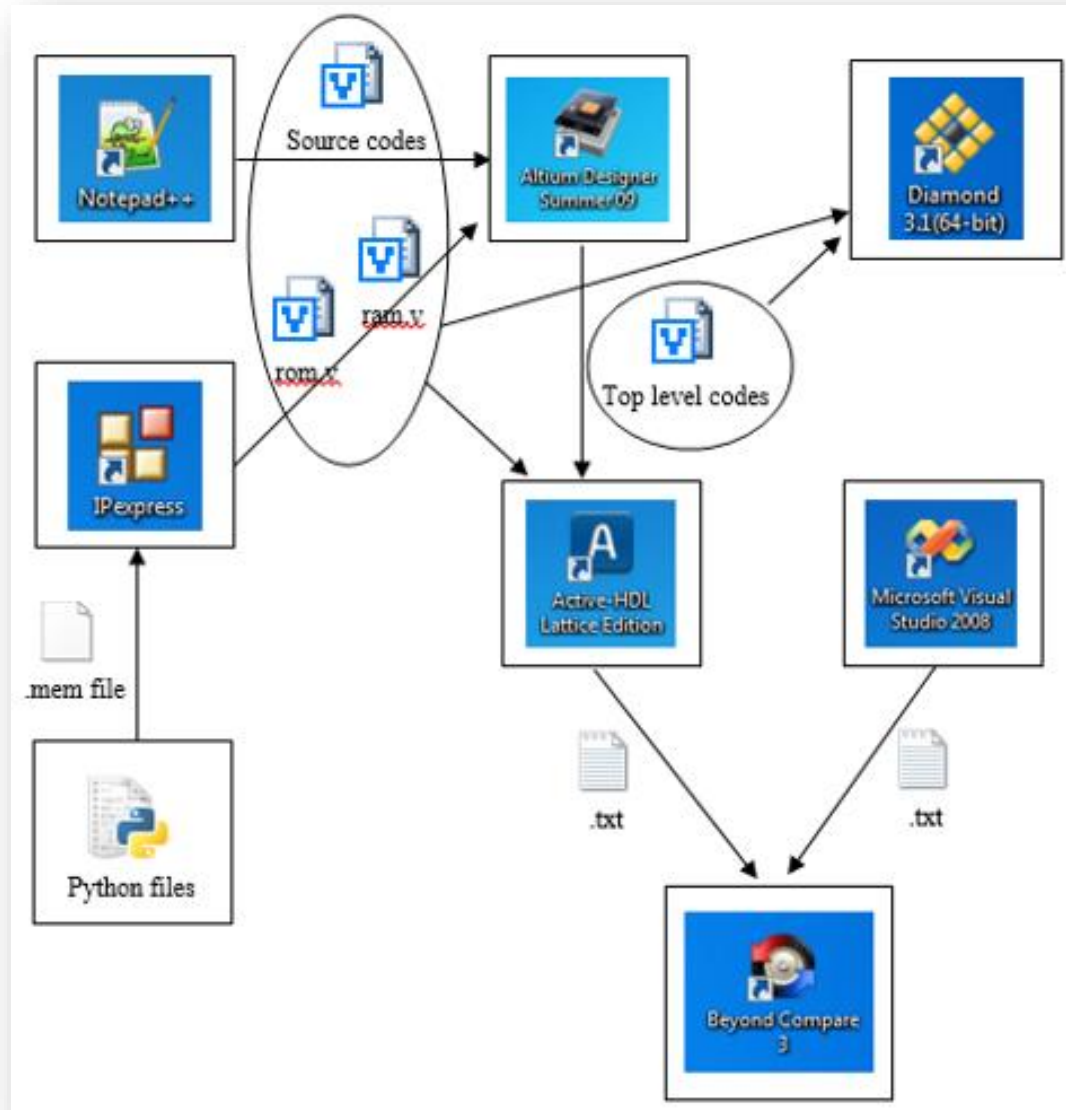
**CAVLC/CAVCD :**

different sets of variable-length codes are chosen depending on the statistics of recently-coded coefficients

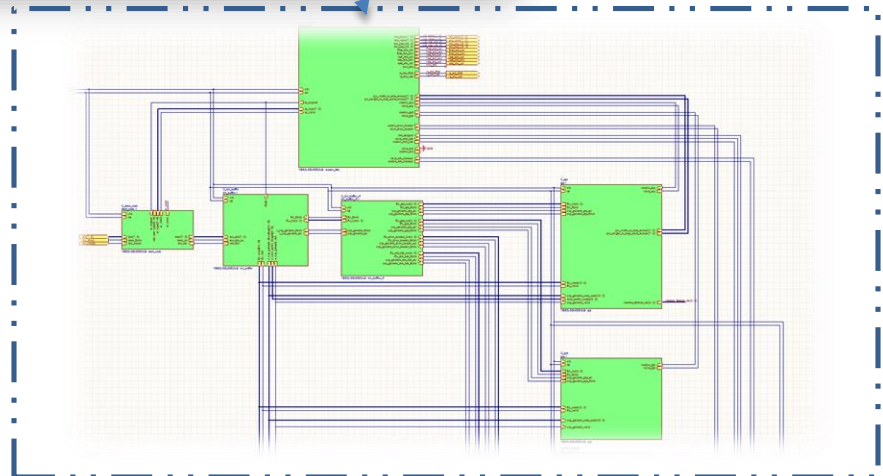
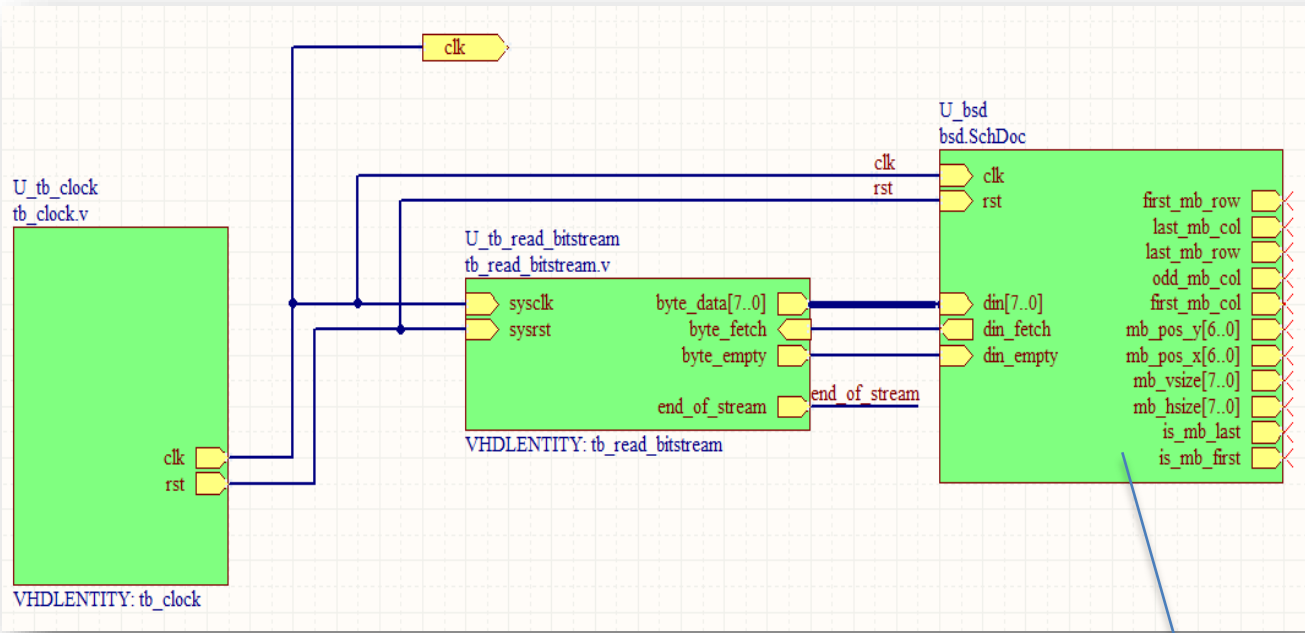
## III.4 Development progress



## III.5 Tools used in the FPGA development flow design



# III.6 Simulation and results



## III.6 Simulation and results

Example of a video H.264 test file used in the simulations in ActiveHDL

The screenshot displays the ActiveHDL simulation environment for an H.264 video test file. The top window shows a sequence of five video frames labeled 0 through 4, with frame types F, IDR, P, F, P, F, IDR. Below the frames, a timeline shows POC values: 0, 2, 0, 2, 0. The middle window shows a grid of macroblocks. The bottom-left window shows a table of sequence parameters for the IDR frame.

Field	Value
seq_parameter_set_rbsp()	
{	
profile_idc	100(0..)
constraint_set0_flag	false
constraint_set1_flag	false
constraint_set2_flag	false
constraint_set3_flag	false
}	

The bottom-right window shows a hex dump of the video data. The dump is in Hex mode and shows the following data:

```

00000000 00 00 00 01 67 64 00 1E AC 1B 1A 81 60 94 40 00 00 01 68 CE 3C 80 00
00000018 00 00 01 65 B8 00 04 00 00 0B A0 C6 01 A0 00 76 00 06 68 00 09 34 CF 32
00000030 6D 09 EC 68 51 3D 16 CB A0 5F 99 C2 E0 03 2D D8 00 CB 56 AD D2 70 01 96
00000048 94 00 1A 71 C1 8E 6D E9 F8 51 40 06 E6 3E CC F8 1D 1A E0 AF A0 B1 C6 6A
00000060 92 48 AF 00 06 A7 00 04 CC 6E D7 22 53 98 B8 00 CA 2F 80 03 4E DD 1E F4
00000078 9C 01 96 78 02 30 8E D0 C2 80 0F 75 A0 1A 79 D0 00 1A 6A CB 00 0B 07 00
00000090 02 17 9F 52 7E A5 00 33 90 7C B5 78 D6 FD 6A 19 38 03 2C 9C 01 96 DD 0F
000000A8 F5 27 00 65 93 80 CB 8E 74 9B DB 86 14 E8 00 1B A0 06 2E 46 26 F9 AA 4A
000000C0 A7 8C 95 6E B7 49 C0 19 69 40 06 C6 DD 5C 60 9C 06 59 38 03 2E BA 5F 0C
000000D8 28 03 A0 00 66 80 18 98 A2 C5 39 92 E4 18 53 6E C4 54 01 92 13 00 C4 B7
000000F0 62 3C 01 32 78 02 66 BA 7F 0C 28 02 A0 00 56 4A 00 44 C0 0B 9F 36 91 93
00000108 E9 79 A2 9A 05 C3 5D 02 3C 01 32 78 02 65 61 E9 D6 78 02 64 F0 04 CA E9
  
```

# III.6 Simulation and results

The screenshot displays the Active-HDL 9.3 simulation environment. The main window shows a waveform for the 'tb\_bsd' testbench. The signal list on the left includes various control signals like `tb_bsdU_bsdICLK`, `tb_bsdU_bsdDIN`, and `tb_bsdU_bsdDIN_EMPTY`, as well as data signals like `tb_bsdU_bsdNamedSignal_FIRST_MB_COL` and `tb_bsdU_bsdPinSignal_U_bit_buffer_ifc_word`. The waveform shows digital signals over time, with a cursor at 3400 ns. The console at the bottom shows the following output:

```

> run 100 ns
# KERNEL: stopped at time: 3200 ns
> run 100 ns
# KERNEL: stopped at time: 3300 ns
> run 100 ns
# KERNEL: stopped at time: 3400 ns
>
  
```

## III.6 Simulation and results

Simulation results for SPS layer in Active HDL

Trace file created by  
JM ITU-T C code

▶ /tb_bsd/U_bsd/U_sps/rst	0
▶ /tb_bsd/U_bsd/U_sps/clk	1
▶ /tb_bsd/U_bsd/U_sps/enable_seq	1
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_state	12
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_profile_idc	100
▶ /tb_bsd/U_bsd/U_sps/r_constraint_set0_flag	0
▶ /tb_bsd/U_bsd/U_sps/r_constraint_set1_flag	0
▶ /tb_bsd/U_bsd/U_sps/r_constraint_set2_flag	0
▶ /tb_bsd/U_bsd/U_sps/r_constraint_set3_flag	0
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_reserved_zero_4bits	0
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_level_idc	30
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_seq_parameter_set_id_sps	0
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_chroma_format_idc	1
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_bit_depth_luma_minus8	0
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_bit_depth_chroma_minus8	0
▶ /tb_bsd/U_bsd/U_sps/r_qpprime_y_zero_transform_bypass_flag	0
▶ /tb_bsd/U_bsd/U_sps/r_seq_scaling_matrix_present_flag	0
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_log2_max_frame_num_minus4	12
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_pic_order_cnt_type	0
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_log2_max_pic_order_cnt_lsb_minus4	12
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_num_ref_frames	1
▶ /tb_bsd/U_bsd/U_sps/r_gaps_in_frame_num_value_allowed_flag	0
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_pic_width_in_mbs_minus1	21
⊕ ▶ /tb_bsd/U_bsd/U_sps/r_pic_height_in_map_units_minus1	17
▶ /tb_bsd/U_bsd/U_sps/r_frame_mbs_only_flag	1
▶ /tb_bsd/U_bsd/U_sps/r_direct_8x8_inference_flag	0
▶ /tb_bsd/U_bsd/U_sps/r_frame_cropping_flag	0
▶ /tb_bsd/U_bsd/U_sps/r_vui_parameters_present_flag	0
▶ /tb_bsd/U_bsd/U_sps/r_valid_seq	1

Line	Signal	Value
1		
2	SPS: profile_idc	01100100 (100)
3	SPS: constrained_set0_flag	0 ( 0)
4	SPS: constrained_set1_flag	0 ( 0)
5	SPS: constrained_set2_flag	0 ( 0)
6	SPS: constrained_set3_flag	0 ( 0)
7	SPS: reserved_zero_4bits	0000 ( 0)
8	SPS: level_idc	00011110 ( 30)
9	SPS: seq_parameter_set_id	1 ( 0)
10	SPS: chroma_format_idc	010 ( 1)
11	SPS: bit_depth_luma_minus8	1 ( 0)
12	SPS: bit_depth_chroma_minus8	1 ( 0)
13	SPS: lossless_qpprime_y_zero_flag	0 ( 0)
14	SPS: seq_scaling_matrix_present_flag	0 ( 0)
15	SPS: log2_max_frame_num_minus4	0001101 ( 12)
16	SPS: pic_order_cnt_type	1 ( 0)
17	SPS: log2_max_pic_order_cnt_lsb_minus4	0001101 ( 12)
18	SPS: num_ref_frames	010 ( 1)
19	SPS: gaps_in_frame_num_value_allowed_flag	0 ( 0)
20	SPS: pic_width_in_mbs_minus1	000010110 ( 21)
21	SPS: pic_height_in_map_units_minus1	000010010 ( 17)
22	SPS: frame_mbs_only_flag	1 ( 1)
23	SPS: direct_8x8_inference_flag	0 ( 0)
24	SPS: frame_cropping_flag	0 ( 0)
25	SPS: vui_parameters_present_flag	0 ( 0)



## III.6 Simulation and results

Trace file created by *AtiveHDL*

Trace file created by *JM ITU-T C code*

```

1 SPS: profile_idc 01100100 (100)
2 SPS: constrained_set0_flag 0 ( 0)
3 SPS: constrained_set1_flag 0 ( 0)
4 SPS: constrained_set2_flag 0 ( 0)
5 SPS: constrained_set3_flag 0 ( 0)
6 SPS: reserved_zero_4bits 0000 ( 0)
7 SPS: level_idc 00011110 (30)
8 SPS: seq_parameter_set_id 1 ( 0)
9 SPS: chroma_format_idc 010 ( 1)
10 SPS: bit_depth_luma_minus8 1 ( 0)
11 SPS: bit_depth_chroma_minus8 1 ( 0)
12 SPS: lossless_qprime_y_zero_flag 0 ( 0)
13 SPS: seq_scaling_matrix_present_flag 0 ( 0)
14 SPS: log2_max_frame_num_minus4 0001101 (12)
15 SPS: pic_order_cnt_type 1 ( 0)
16 SPS: log2_max_pic_order_cnt_lsb_minus4 0001101 (12)
17 SPS: num_ref_frames 010 ( 1)
18 SPS: gaps_in_frame_num_value_allowed_flag 0 ( 0)
19 SPS: pic_width_in_mbs_minus1 000010110 (21)
20 SPS: pic_height_in_map_units_minus1 000010010 (17)
21 SPS: frame_mbs_only_flag 1 ( 1)
22 SPS: direct_8x8_inference_flag 0 ( 0)
23 SPS: frame_cropping_flag 0 ( 0)
24 SPS: vui_parameters_present_flag 0 ( 0)
25
26
27 PPS: pic_parameter_set_id 1 ( 0)
28 PPS: seq_parameter_set_id 1 ( 0)
29 PPS: entropy_coding_mode_flag 0 ( 0)
30 PPS: bottom_field_pic_order_in_frame_present_flag 0 ( 0)
31 PPS: num_slice_groups_minus1 1 ( 0)
32 PPS: num_ref_idx_l0_default_active_minus1 1 ( 0)
33 PPS: num_ref_idx_l1_default_active_minus1 1 ( 0)
34 PPS: weighted_pred_flag 0 ( 0)
35 PPS: weighted_bipred_idc 00 ( 0)
36 PPS: pic_init_qp_minus26 1 ( 0)
37 PPS: pic_init_qs_minus26 1 ( 0)
38 PPS: chroma_qp_index_offset 1 ( 0)
39 PPS: deblocking_filter_control_present_flag 1 ( 1)
40 PPS: constrained_intra_pred_flag 0 ( 0)
41 PPS: redundant_pic_cnt_present_flag 0 ( 0)
42
43 SH: first_mb_in_slice 1 ( 0)
44 SH: slice_type 011 ( 2)
45 SH: pic_parameter_set_id 1 ( 0)
46 SH: frame_num 0000000000000000 ( 0)
47 SH: idr_pic_id 0000000000000001 ( 0)
48 SH: pic_order_cnt_lsb 0000000000000000 ( 0)

```

```

1 SPS: profile_idc 01100100 (100)
2 SPS: constrained_set0_flag 0 ( 0)
3 SPS: constrained_set1_flag 0 ( 0)
4 SPS: constrained_set2_flag 0 ( 0)
5 SPS: constrained_set3_flag 0 ( 0)
6 SPS: reserved_zero_4bits 0000 ( 0)
7 SPS: level_idc 00011110 (30)
8 SPS: seq_parameter_set_id 1 ( 0)
9 SPS: chroma_format_idc 010 ( 1)
10 SPS: bit_depth_luma_minus8 1 ( 0)
11 SPS: bit_depth_chroma_minus8 1 ( 0)
12 SPS: lossless_qprime_y_zero_flag 0 ( 0)
13 SPS: seq_scaling_matrix_present_flag 0 ( 0)
14 SPS: log2_max_frame_num_minus4 0001101 (12)
15 SPS: pic_order_cnt_type 1 ( 0)
16 SPS: log2_max_pic_order_cnt_lsb_minus4 0001101 (12)
17 SPS: num_ref_frames 010 ( 1)
18 SPS: gaps_in_frame_num_value_allowed_flag 0 ( 0)
19 SPS: pic_width_in_mbs_minus1 000010110 (21)
20 SPS: pic_height_in_map_units_minus1 000010010 (17)
21 SPS: frame_mbs_only_flag 1 ( 1)
22 SPS: direct_8x8_inference_flag 0 ( 0)
23 SPS: frame_cropping_flag 0 ( 0)
24 SPS: vui_parameters_present_flag 0 ( 0)
25
26
27 PPS: pic_parameter_set_id 1 ( 0)
28 PPS: seq_parameter_set_id 1 ( 0)
29 PPS: entropy_coding_mode_flag 0 ( 0)
30 PPS: bottom_field_pic_order_in_frame_present_flag 0 ( 0)
31 PPS: num_slice_groups_minus1 1 ( 0)
32 PPS: num_ref_idx_l0_active_minus1 1 ( 0)
33 PPS: num_ref_idx_l1_active_minus1 1 ( 0)
34 PPS: weighted_pred_flag 0 ( 0)
35 PPS: weighted_bipred_idc 00 ( 0)
36 PPS: pic_init_qp_minus26 1 ( 0)
37 PPS: pic_init_qs_minus26 1 ( 0)
38 PPS: chroma_qp_index_offset 1 ( 0)
39 PPS: deblocking_filter_control_present_flag 1 ( 1)
40 PPS: constrained_intra_pred_flag 0 ( 0)
41 PPS: redundant_pic_cnt_present_flag 0 ( 0)
42
43 SH: first_mb_in_slice 1 ( 0)
44 SH: slice_type 011 ( 2)
45 SH: pic_parameter_set_id 1 ( 0)
46 SH: frame_num 0000000000000000 ( 0)
47 SH: idr_pic_id 1 ( 0)
48 SH: pic_order_cnt_lsb 0000000000000000 ( 0)

```

## III.6 Simulation and results

Design Summary

- Project
  - Project Summary
- Process Reports
  - Synplify Pro
  - Map
  - Place & Route
  - Signal/Pad
  - Bitstream
- Analysis Reports
  - Map Trace
  - Place & Route Trace
  - I/O Timing Analysis
- Tool Reports
  - I/O SSO Analysis
  - Hierarchy Parsing Report
  - Run BKM Check
  - PIO DRC
  - TCL Command Log
- Messages
  - All Messages
  - User Defined Filters

Lattice Mapping Report File for Design Module 'tb\_bsd'

**Design Information**

```

Command line: map -a LatticeECP3 -p LFE3-150EA -t FPBGA1156 -s 6 -oc
Commercial h264_dec_h264_dec.ngd -o h264_dec_h264_dec_map.ncd -pr
h264_dec_h264_dec.prf -mp h264_dec_h264_dec.mrp C:/Users/ENCIRIS/Desktop/KK
_INTERNSHIP_FILES/KK_Diamond/kk_diamond_h264_dec_v1/h264_dec.lpf -gui
Target Vendor: LATTICE
Target Device: LFE3-150EAFPBGA1156
Target Performance: 6
Mapper: ep5c00, version: Diamond (64-bit) 3.1.0.96
Mapped on: 08/14/14 13:56:21
          
```

**Design Summary**

```

Number of registers: 468 out of 115296 (0%)
PFU registers: 460 out of 111780 (0%)
PIO registers: 8 out of 3516 (0%)
Number of SLICES: 535 out of 74520 (1%)
SLICES as Logic/ROM: 529 out of 74520 (1%)
SLICES as RAM: 6 out of 14220 (0%)
SLICES as Carry: 61 out of 74520 (0%)
Number of LUT4s: 852 out of 149040 (1%)
Number of logic LUTs: 718
Number of distributed RAM: 6 (12 LUT4s)
Number of ripple logic: 61 (122 LUT4s)
Number of shift registers: 0
Number of PIO sites used: 12 out of 586 (2%)
Number of PIO FIXEDDELAY: 0
Number of PCS (SerDes): 0 out of 4 (0%) with bonded PIO sites
Number of DQS DLLs: 0 out of 2 (0%)
Number of PLLs: 0 out of 10 (0%)
Number of DLLs: 0 out of 2 (0%)
Number of block RAMs: 0 out of 372 (0%)
Number of CLKDIVs: 0 out of 2 (0%)
Number of GSRs: 1 out of 1 (100%)
JTAG used : No
Readback used : No
Oscillator used : No
Startup used : No
Notes:-
1. Total number of LUT4s = (Number of logic LUT4s) + 2*(Number of
distributed RAMs) + 2*(Number of ripple logic)
2. Number of logic LUT4s does not include count of distributed RAM and
ripple logic.
          
```

Lattice FPGA\_LFE3-150EA-6FN1156C

Number of registers:	468 / 115296 (0 %)
Number of SLICES:	535 / 74520 (1 %)
Number of LUT4s:	852 / 149040 (1 %)

### Preference Summary

- FREQUENCY PORT "CLK" 150.000000 MHz (0 errors)  
0 items scored, 0 timing errors detected.  
Report: 684.932MHz is the maximum frequency for this preference.
- FREQUENCY NET "CLK\_c" 150.000000 MHz (0 errors)  
4096 items scored, 0 timing errors detected.  
Report: 198.334MHz is the maximum frequency for this preference.

Design can run on the maximum frequency of **198.334 MHz** with the clock constraint of 150 MHz

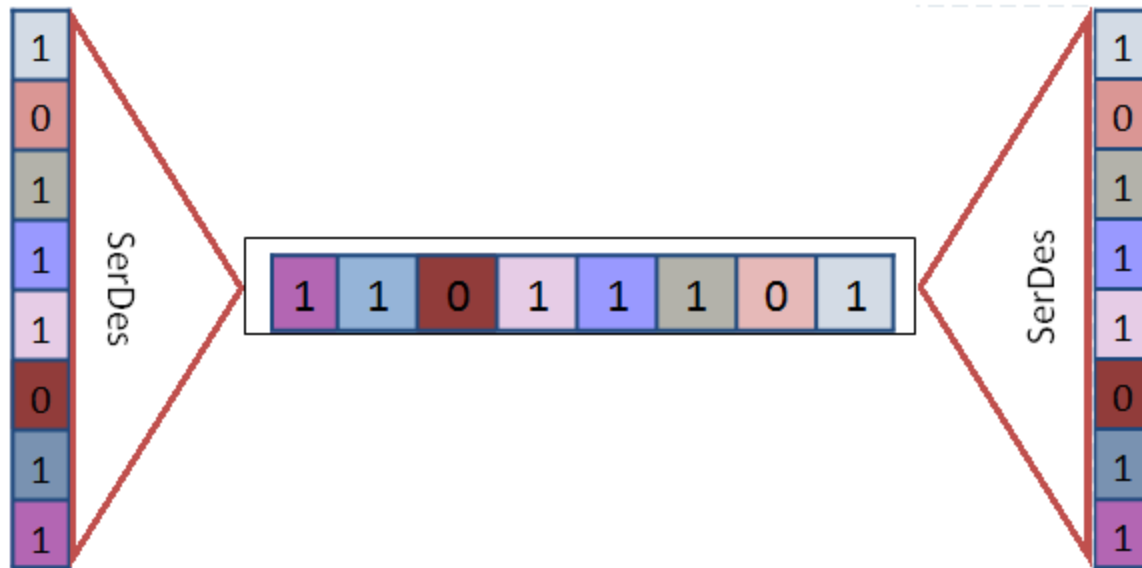
## IV. Conclusion

- Accomplished the first task on the implementation of a Lattice tri-rate SDI PHY IP core on the Enciris LT-125 board, provided an SMPTE 3G/HD/SD SDI video input functionality on the board.
- Because of lacking time and because of the complexity of CAVLD decoding, we can at the present manage to design the bitstream decoder which can decode the SPS, PPS, Slice header, and Macroblock header layer of the bitstream and which can run on the Lattice FPGA device - LFE3-150EA-6FN1156C with the maximum frequency of 198.334 MHz where the clock constraint is 150 MHz.

**THANKS FOR YOUR ATTENTION**

# **ANNEXE**

# SERDES



# IPexpress - SERDES

The screenshot shows the Lattice FPGA Module -- PCS configuration window. The window is titled "Lattice FPGA Module -- PCS" and has a "Configuration" tab selected. The main area is divided into two panes. The left pane shows a list of pins for the PCS module, with "[19:0] txdata\_ch0" selected. The right pane shows the "Channel Protocol" configuration table.

Channel	RX and TX	RX Only	TX Only	Disable Channel	Protocol	Low Speed Data Ports
Channel0	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	SDI	<input checked="" type="checkbox"/>
Channel1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	G8B10B	<input type="checkbox"/>
Channel2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	G8B10B	<input type="checkbox"/>
Channel3	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	SDI	<input checked="" type="checkbox"/>

Buttons at the bottom: Generate, Close, Help.

# IPexpress - Tri-rate SDI IP Core

The screenshot displays the configuration interface for the Lattice IP Core -- Tri-Rate SDI PHY v1.5. The window is divided into two main sections: a block diagram on the left and a configuration panel on the right.

**Block Diagram (Tri-Rate SDI PHY):**

- Inputs:** rxdata[19:0], rx\_clk, rx\_full\_clk, rx\_hd\_sdn, rx\_tg\_hdn, rstn, txdata[19:0].
- Outputs:** pd\_out[19:0], pdo\_clk, vid\_active, vid\_format[1:0], frame\_format[2:0], trs\_out, field, vblank, hblank, ln1\_out[10:0], eav\_error, sav\_err, y1\_crc\_error, c1\_crc\_error, rx\_rate[2:0], pd\_in[19:0], pdi\_clk, tx\_half\_clk, trs\_in, hd\_sdn\_in.

**Configuration Panel (PHY / Custom Format / Advanced):**

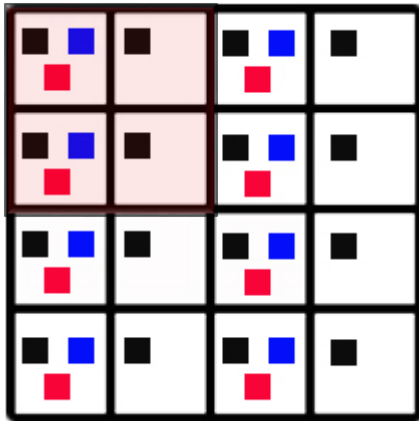
- PHY Function:** Radio buttons for Tx, Rx, and Both. **Both** is selected.
- 3G Level-B option:**  Enable 3G Level-B.
- 3G/HD Transmit Options:**
  - LN Insertion:** Radio buttons for Off (selected) and On.
  - VPID insertion:** Radio buttons for Off (selected) and On.
  - CRC Insertion:** Radio buttons for Off (selected) and On.
- SD Transmit Options:**
  - LDR path for SD
  - Include PLL for LDR
  - 10 bits mode for SD Tx
  - Separate data input for SD
  - SD data width:** Radio buttons for 8 bits, 10 bits (selected), and Dynamic 8/10 bits.
- 3G/HD Receive option:** VPID extraction radio buttons for Off (selected) and On.
- SD Receive options:**  10 bits mode for SD Rx.
- Optional ports:**  Clock enable port.

At the bottom of the window, there are three buttons: **Generate**, **Close**, and **Help**.

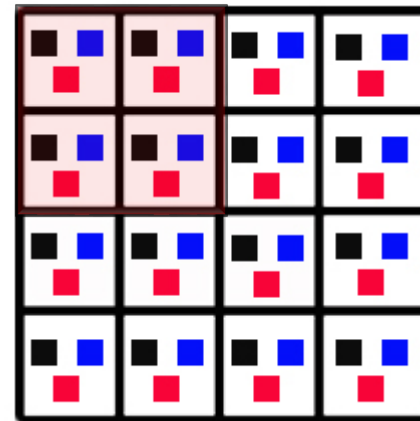


# YCbCr 422 vs 444 sample

422

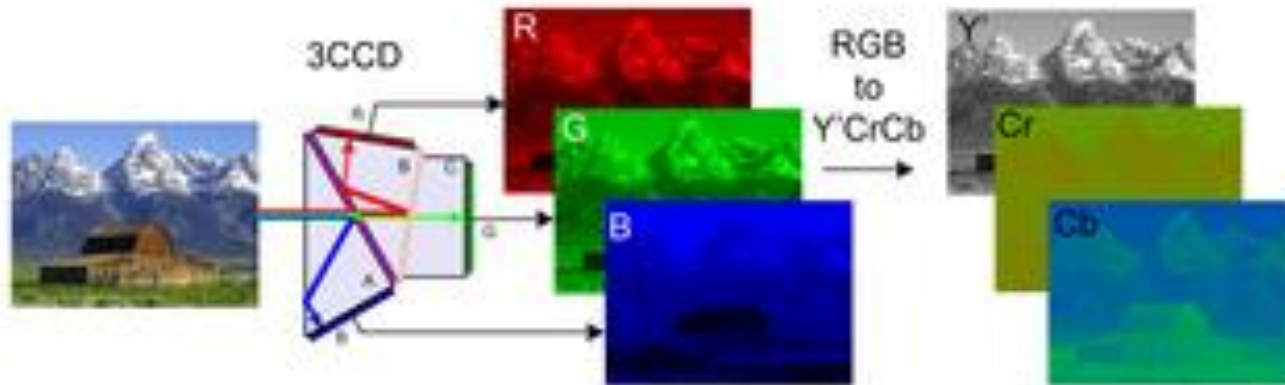


444

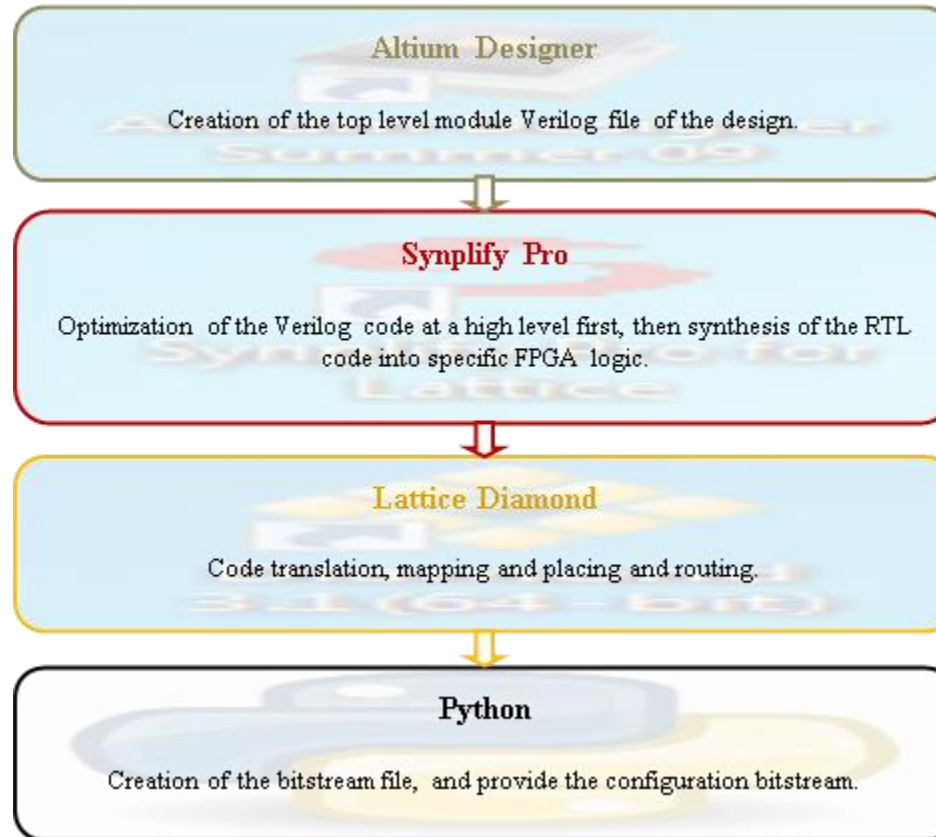


■ : Y  
■ : C<sub>b</sub>  
■ : C<sub>r</sub>

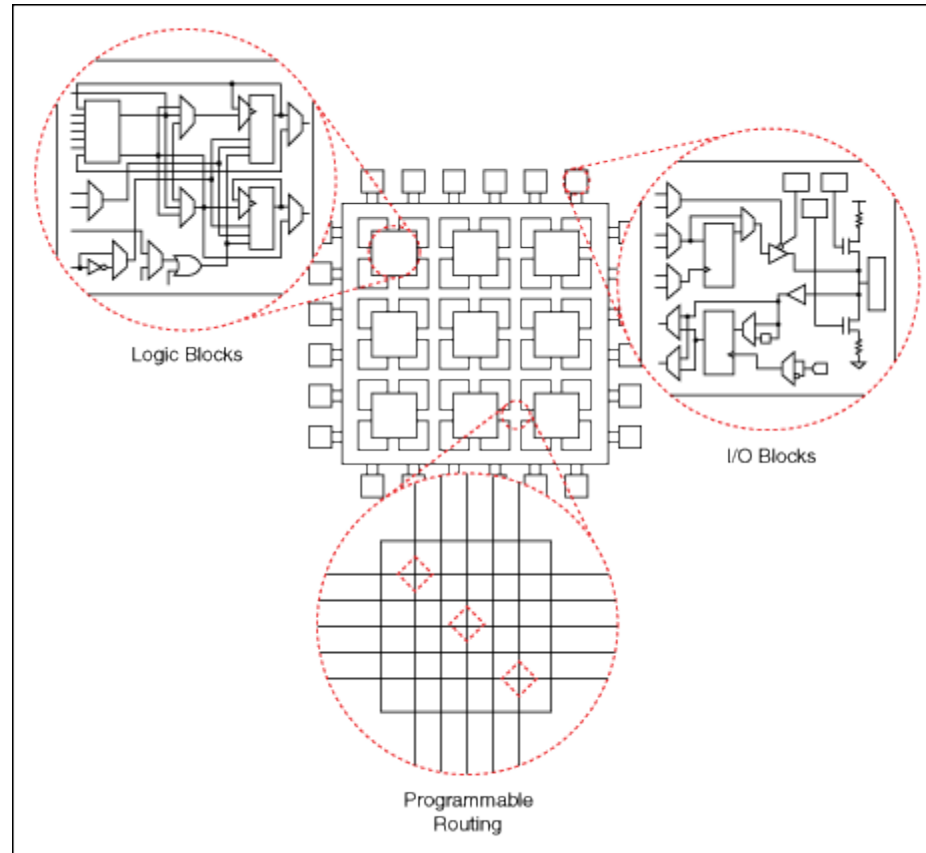
# RGB to $YCbCr$



# Flow of bitstream implementation



# Introduction to FPGA Hardware Concepts (FPGA Module)



# LUT

A lookup table (LUT) is used to transform the input data into a more desirable output format.

A lookup table (LUT) is a fast way to realize a complex function in digital logic. The address is the function input, and the value at that address is the function output. The advantage is that computing the function only takes a single memory lookup regardless of the complexity of the function, so is very fast. The disadvantage is that it takes memory, especially if you need high resolution for the function input.

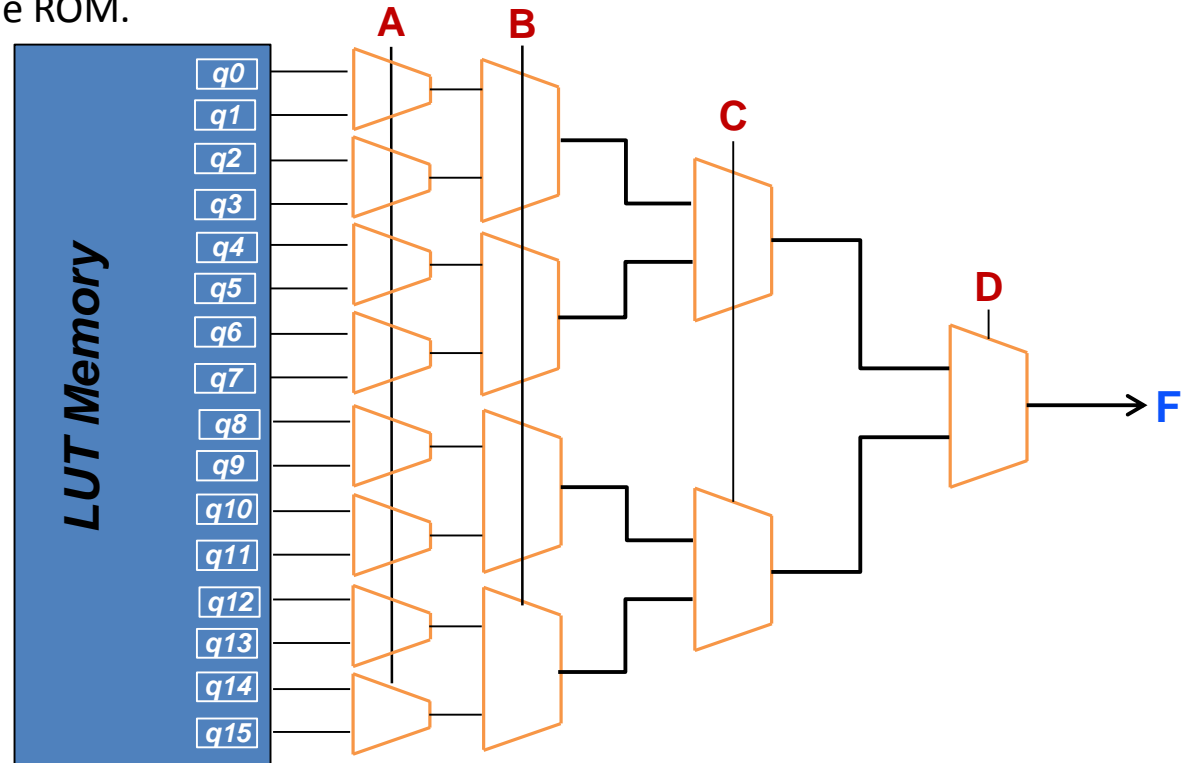
For example, SIN is often implemented as a table lookup. If 10 bit angles are good enough resolution, then the whole function can be implemented as a lookup table with 1024 entries. (Actually in the case of SIN, only 1/4 cycle is stored then negated or indexed backwards depending on the actual quadrant, but that is a aside specific to SIN).

The function input can also be a combination of different input variables with the result expressed as a single integer. For example, a 4 x 4 bit multiply can be implemented as a lookup table of 256 values. The 8-bit table address can be the two 4-bit input values concatenated.

# The look-up-table (LUT)

- Building the PFU from the inside out...

- Nearly all FPGAs are based on a Look-Up-Table plus Register. Most are a LUT4. Aka LUT4+REG.
- A 4-input LUT is just a 16-bit ROM, with 4 'address' bits (ABCD) and a 'data' bit (F).
- By programming the ROM, any 4 input logic functions can be formed.
  - Or it can be a simple ROM.



('q' values are programmable SRAM memory bits that are determined through the design synthesis process)

# Building 'Slices'

Pairs of LUT+REG are grouped together with extra RAM/Ripple logic to form SLICES.

## Each Slice consists of:

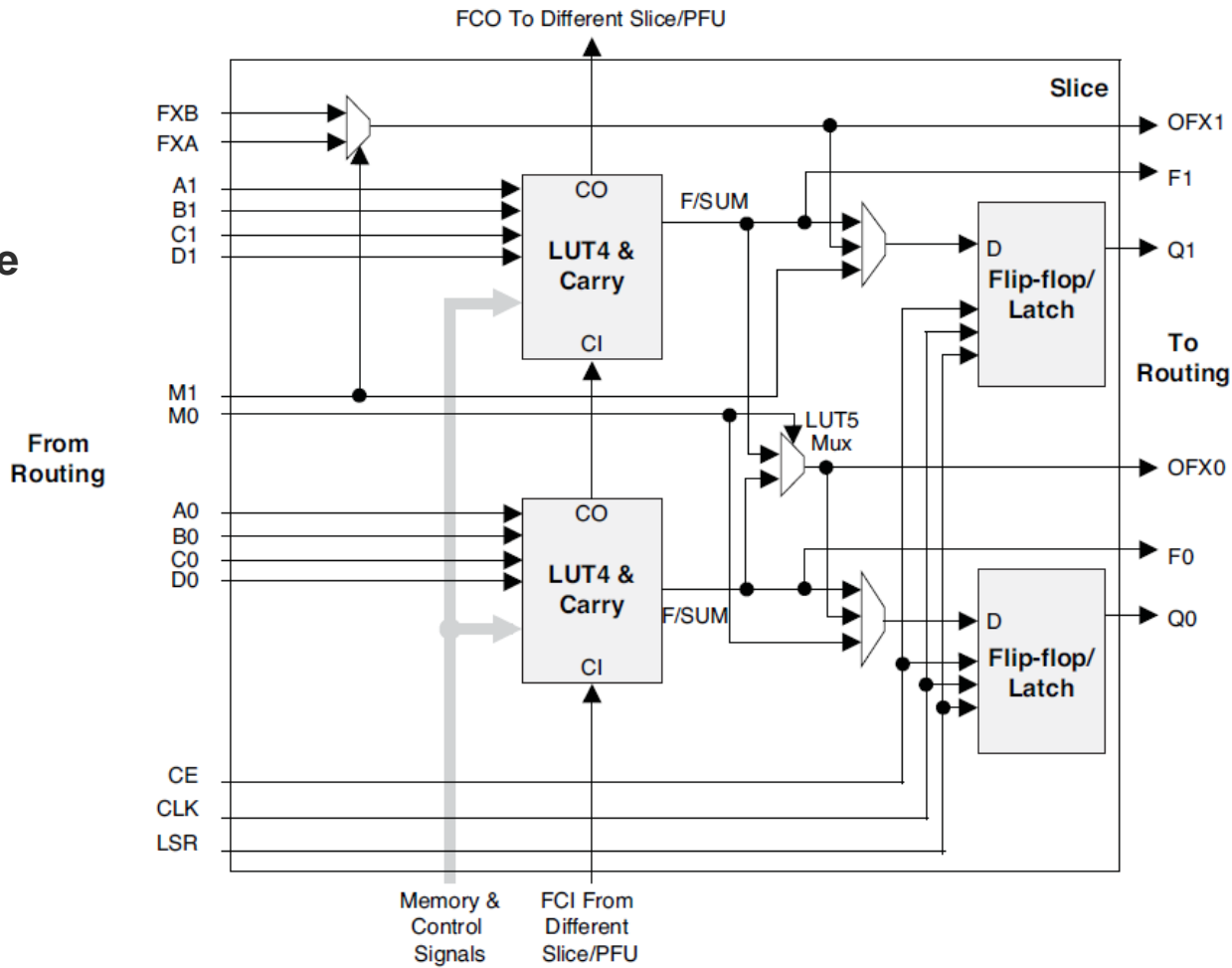
- Two 4-Input LUTs
- Two Registers
- Arithmetic Logic circuits
- Circuitry to support simple RAM mode

## Slice Inputs:

- LUT Inputs: A, B, C, D
- Multi-Purpose Inputs: M
- Fast Carry Input: FCI
- Register Control Inputs: CLK, CE, LSR

## Slice Outputs:

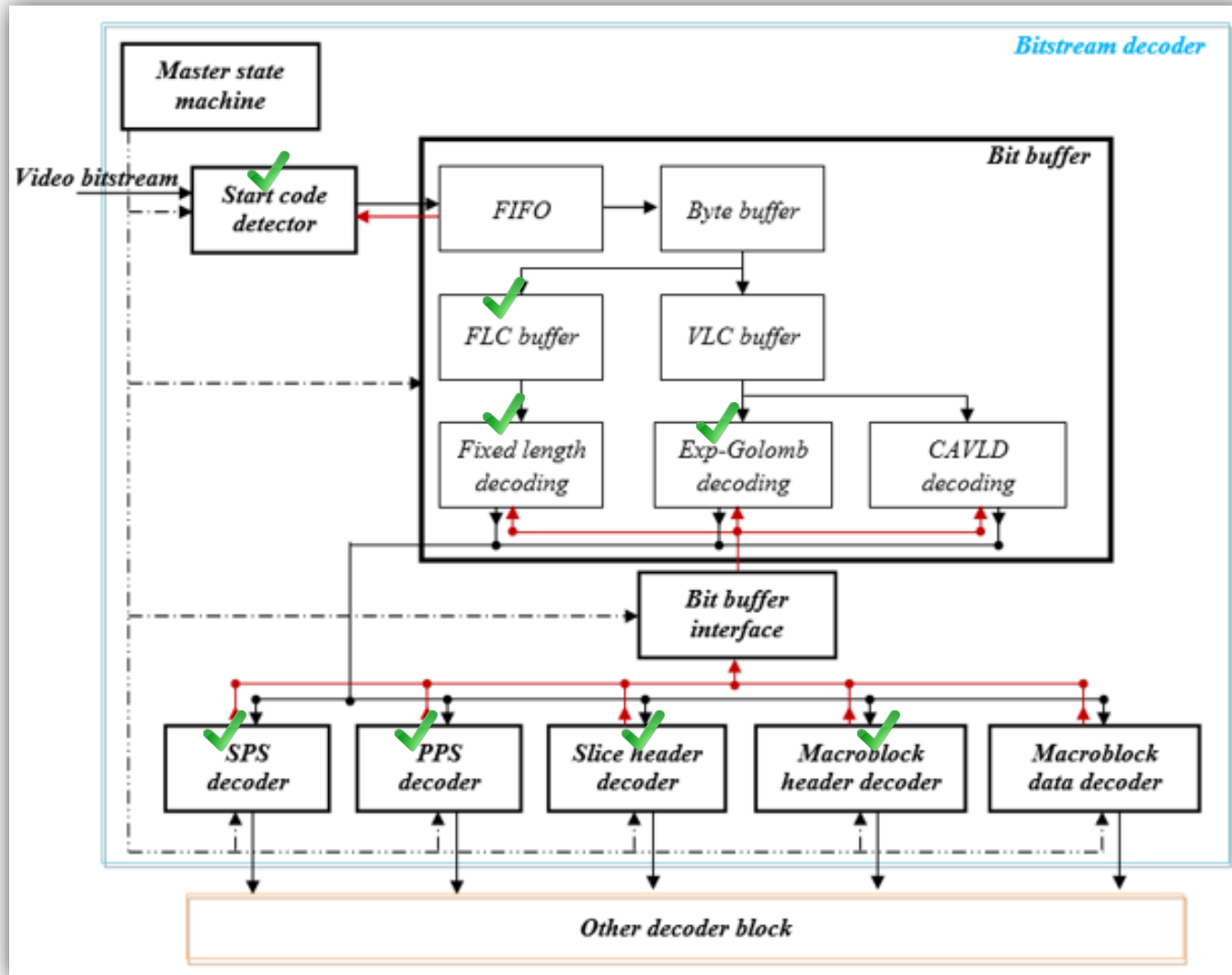
- LUT Outputs: F
- Register Outputs: Q
- Wide Function Outputs: OFX
- Fast Carry Output: FCO



For Slices 0 and 1, memory control signals are generated from Slice 2 as follows:

- WCK is CLK
- WRE is from LSR
- DI[3:2] for Slice 1 and DI[1:0] for Slice 0 data from Slice 2
- WAD [A:D] is a 4-bit address from slice 2 LUT input

# H.264/AVC bitstream decoder hardware architecture





# CAVLD decoding

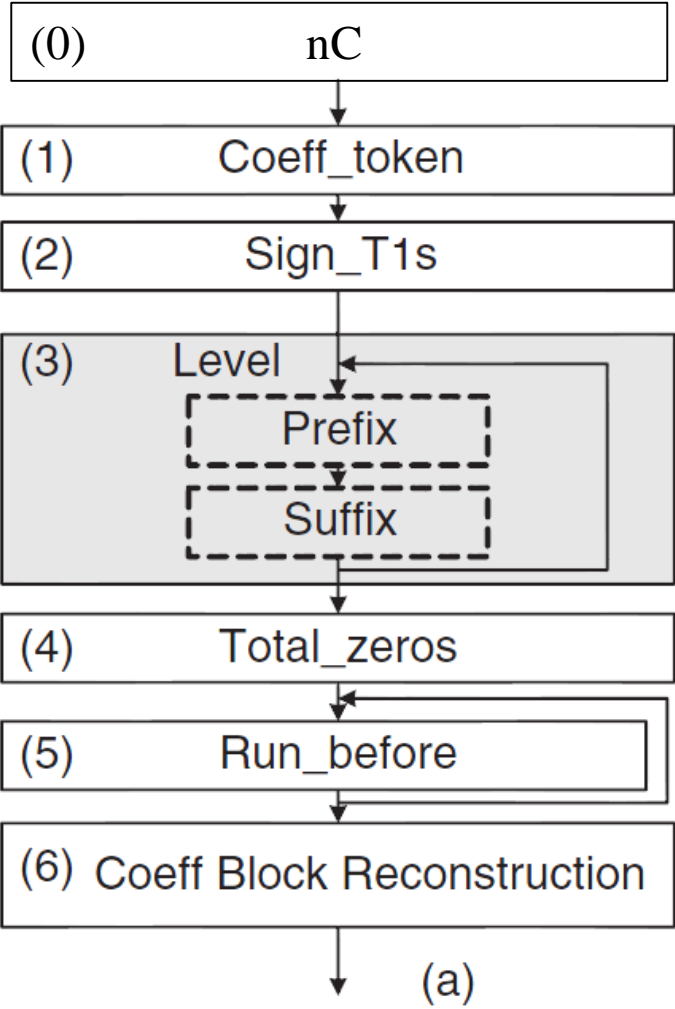


Table 9-5 – coeff\_token mapping to TotalCoeff(coeff\_token) and TrailingOnes(coeff\_token)

TrailingOnes (coeff_token)	TotalCoeff (coeff_token)	$0 \leq nC < 2$	$2 \leq nC < 4$	$4 \leq nC < 8$	$8 \leq nC$	$nC == -1$	$nC == -2$
0	0	1	11	1111	0000 11	01	1
0	1	0001 01	0010 11	0011 11	0000 00	0001 11	0001 111
1	1	01	10	1110	0000 01	1	01
0	2	0000 0111	0001 11	0010 11	0001 00	0001 00	0001 110
1	2	0001 00	0011 1	0111 1	0001 01	0001 10	0001 101
2	2	001	011	1101	0001 10	001	001
0	3	0000 0011 1	0000 111	0010 00	0010 00	0000 11	0000 0011 1
1	3	0000 0110	0010 10	0110 0	0010 01	0000 011	0001 100
2	3	0000 101	0010 01	0111 0	0010 10	0000 010	0001 011
3	3	0001 1	0101	1100	0010 11	0001 01	0000 1
0	4	0000 0001 11	0000 0111	0001 111	0011 00	0000 10	0000 0011 0
1	4	0000 0011 0	0001 10	0101 0	0011 01	0000 0011	0000 0010 1
2	4	0000 0101	0001 01	0101 1	0011 10	0000 0010	0001 010
3	4	0000 11	0100	1011	0011 11	0000 000	0000 01

# Node-Leaf method algorithm

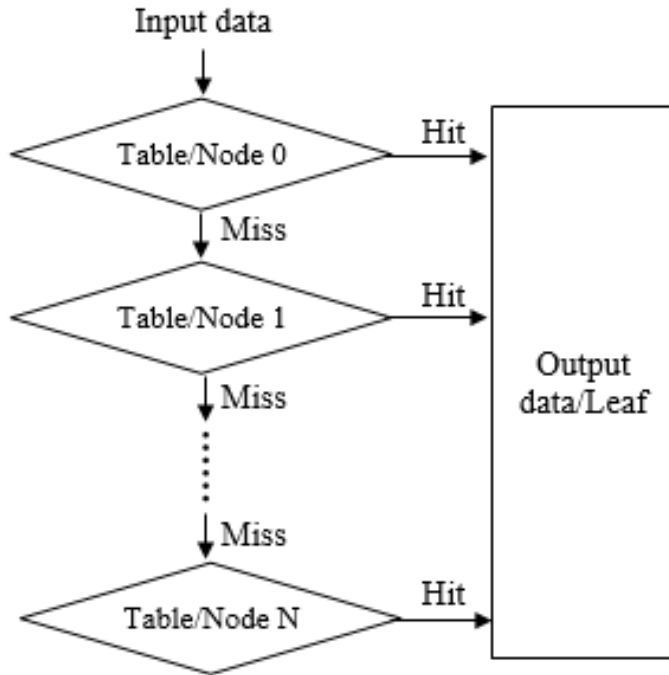


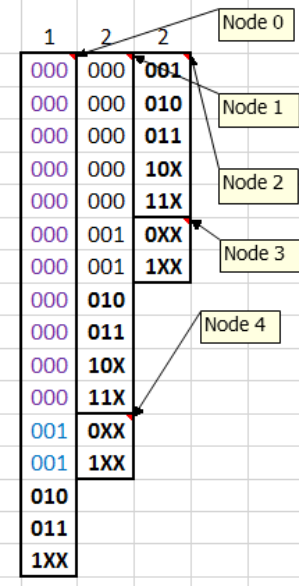
Table 9-7 : total\_zeros tables for 4x4 blocks with TotalCoeff(coeff\_token) 1 to 7

TotalCoeff(coeff\_token) = 1

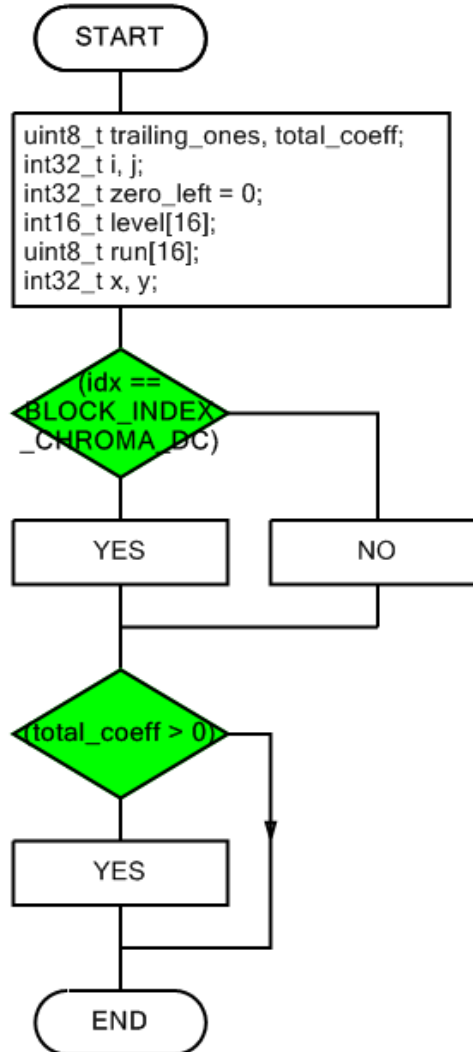
total_zeros	Len	Bin	Bin with don't cares (9 bits width)
15	9	000000001	000000001
14	9	000000010	000000010
13	9	000000011	000000011
12	8	00000010	00000010X
11	8	00000011	00000011X
10	7	0000010	0000010XX
9	7	0000011	0000011XX
8	6	000010	000010XXX
7	6	000011	000011XXX
6	5	00010	00010XXXX
5	5	00011	00011XXXX
4	4	0010	0010XXXXX
3	4	0011	0011XXXXX
2	3	010	010XXXXXX
1	3	011	011XXXXXX
0	1	1	1XXXXXXX

Number of table in that columns

5 x 8 entries



# CAVLD decoding



# (0) nC

(idx ==  
BLOCK\_INDEX  
\_CHROMA\_DC)

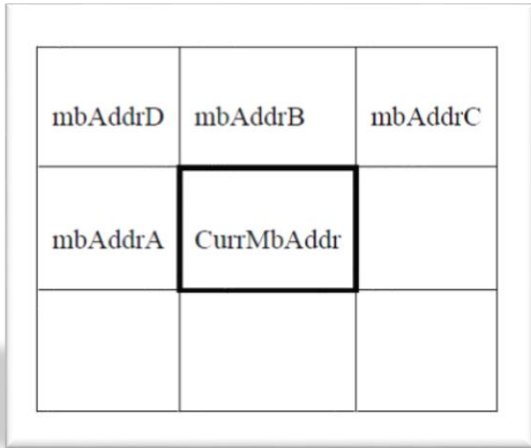
```
T264dec_mb_read_coff_token_t4(t, &trailing_ones, &total_coeff);
```

```
int32_t nC = 0;  
typedef void (*T264dec_mb_read_coff_token_t)(T264_t* t, uint8_t* trailing_ones, uint8_t* total_coeff);  
static const T264dec_mb_read_coff_token_t read_coeff[17] =
```

(idx ==  
BLOCK\_INDEX  
\_LUMA\_DC)

```
nC = T264_mb_predict_non_zero_code(t, 0);  
read_coeff[nC](t, &trailing_ones, &total_coeff);
```

```
nC = T264_mb_predict_non_zero_code(t, idx);  
read_coeff[nC](t, &trailing_ones, &total_coeff);  
assert(total_coeff != 255);  
assert(trailing_ones != 255);
```



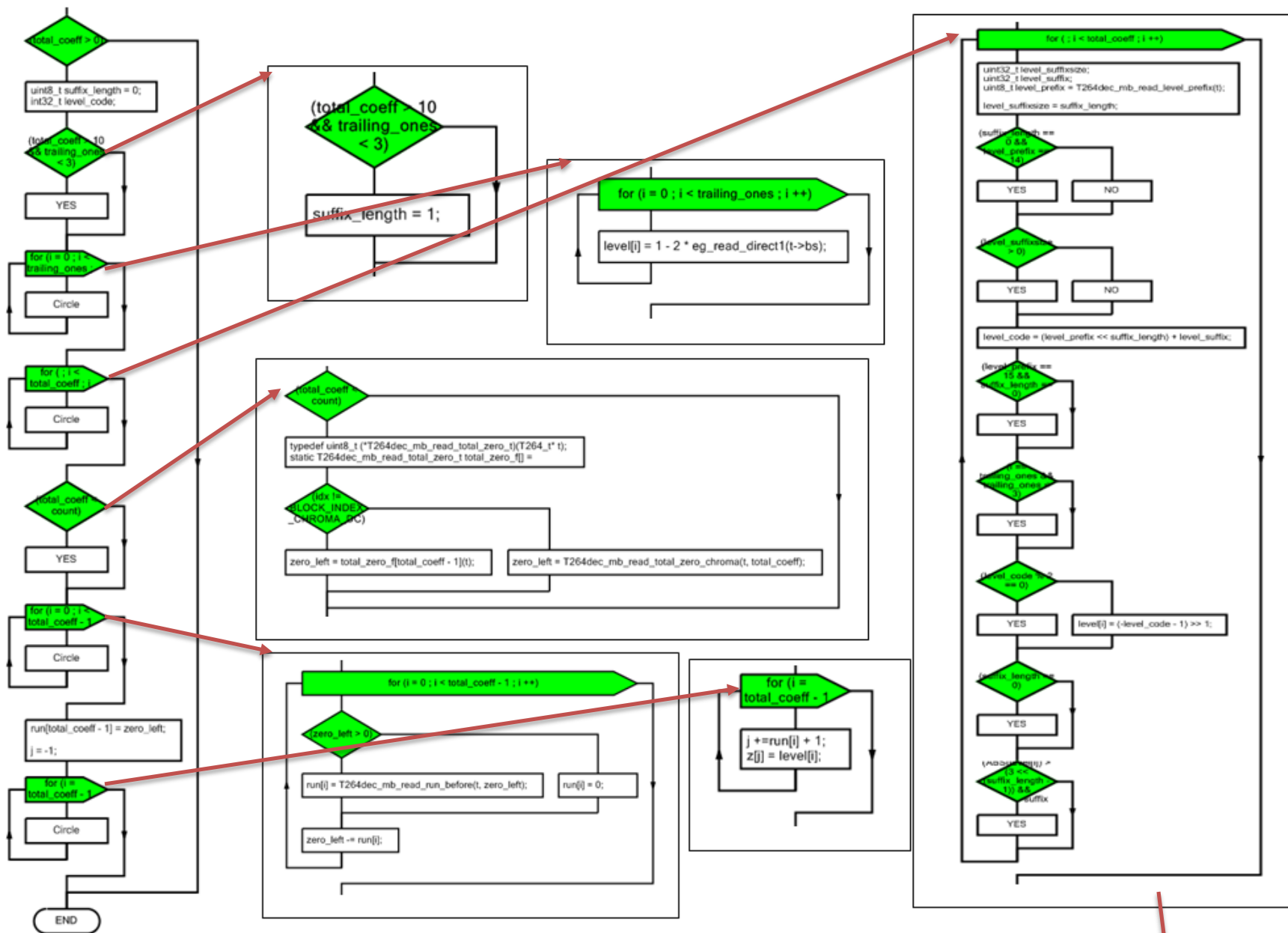
(idx < 16)

```
x = luma_inverse_x[idx];  
y = luma_inverse_y[idx];  
t->mb.nnz[luma_index[idx]] = total_coeff;  
t->mb.nnz_ref[NNZ_LUMA + y * 8 + x] = total_coeff;
```

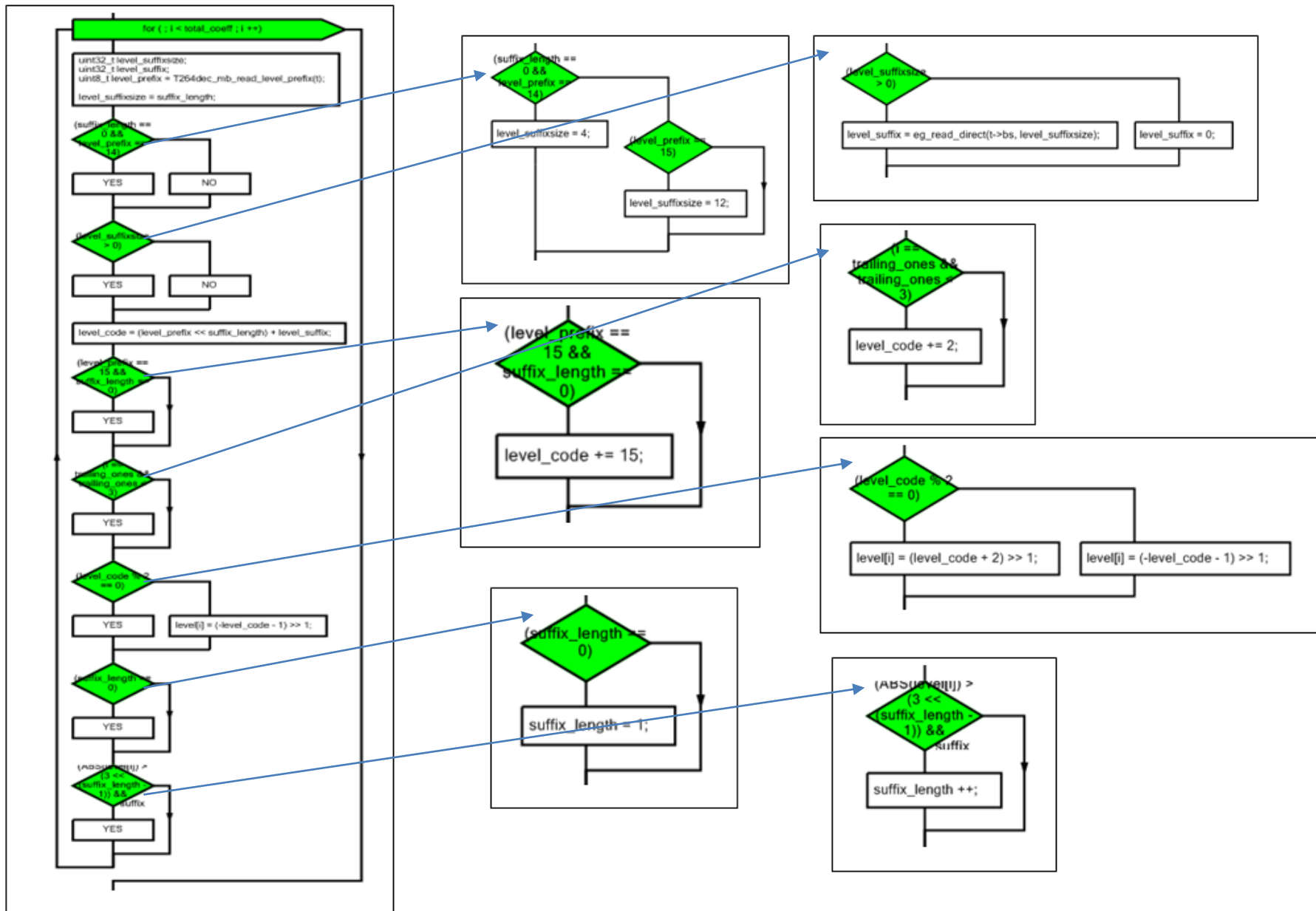
(idx

```
t->mb.  
x = (idx  
y = (idx  
t->mb.
```

# Step (1) to (6)



# Step (1) to (6)



# File .264 Analyse

Vega TSA - S:\exchange\\_EXCHANGE\\_vicka\KK\_INTERNSHIP\_FILES\kk\_active\_hdl\_dec\_h264\dec\_h264\src\kk\_test\_hp\_skip.264

File View Navigate Settings Window Help

Decoding Order [Icons] AU Num [Dropdown]

0 F IDR 1 F P 2 F IDR 3 F P 4 F IDR

00:00:00.000 POC = 0  
00:00:00.033 POC = 2  
00:00:00.066 POC = 0  
00:00:00.100 POC = 2  
00:00:00.133 POC = 0

S:\exchange\\_EXCHANGE\\_vicka\KK\_INTERNSHIP\_FILES\kk\_active\_hdl\_dec\_h264\dec\_h264\src\kk\_test\_hp\_skip.264

- H264 Video Stream
  - FileSize: 76.90 KB
  - Total AUs: 5
  - #Errors: 0
  - #Warnings: 0
  - CC Data: Not Present
  - Progressive Video

Field	Value
Profile	High
Level	3.0
Chroma Format	[4:2:0]
MBAFF Mode	FALSE
Resolution	352 x 288...
Max DPB Size	16
Coding Type	CAVLC
BitDepth Luma	8
BitDepth C	0

Field	Value
seq_parameter_set_rbsp()	
{	
profile_idc	100(0..)
constraint_set0_flag	false
constraint_set1_flag	false
constraint_set2_flag	false
constraint_set3_flag	false
}	

Field	Value
macroblock	
index	0(0x0)
StartAddress(bits)	276(0x114)
Length(bits)	369(0x171)
frame/field	frame
sliceNo	0(0x0)
slice_type	I_slice

Active SPS Active PPS

Vega TSA Deblocking Filter As Per Stream / YUV Dumping Disabled 00:00:00:133 100% AU: 0 Slice: 0 MB: 0

25 errors per page Start AU [Icons]

Number	File-offset	Stream-location	Section	Description
244	N.A.	N.A.	N.A.	Stream Parsing Started .....
245	N.A.	N.A.	N.A.	Stream Parsing End .....

Hex Binary [Icons] Edit Mode ASCII

```

00000000 00 00 00 01 67 64 00 1E AC 1B 1A 81 60 94 40 00 00 01 68 CE 3C 80 00
00000018 00 00 01 65 B8 00 04 00 00 0B A0 C6 01 A0 00 76 00 06 68 00 09 34 CF 32
00000030 6D 09 EC 68 51 3D 16 CB A0 5F 99 C2 E0 03 2D D8 00 CB 56 AD D2 70 01 96
00000048 94 00 1A 71 C1 8E 6D E9 F8 51 40 06 E6 3E CC F8 1D 1A E0 AF A0 B1 C6 6A
00000060 92 48 AF 00 06 A7 00 04 CC 6E D7 22 53 98 B8 00 CA 2F 80 03 4E DD 1E F4
00000078 9C 01 96 78 02 30 8E D0 C2 80 0F 75 A0 1A 79 D0 00 1A 6A CB 00 0B 07 00
00000090 02 17 9F 52 7E A5 00 33 90 7C B5 78 D6 FD 6A 19 38 03 2C 9C 01 96 DD 0F
000000A8 F5 27 00 65 93 80 CB 8E 74 9B DB 86 14 E8 00 1B A0 06 2E 46 26 F9 AA 4A
000000C0 A7 8C 95 6E B7 49 C0 19 69 40 06 C6 DD 5C 60 9C 06 59 38 03 2E BA 5F 0C
000000D8 28 03 A0 00 66 80 18 98 A2 C5 39 92 E4 18 53 6E C4 54 01 92 13 00 C4 B7
000000F0 62 3C 01 32 78 02 66 BA 7F 0C 28 02 A0 00 56 4A 00 44 C0 0B 9F 36 91 93
00000108 E9 79 A2 9A 05 C3 5D 02 3C 01 32 78 02 65 61 E9 D6 78 02 64 F0 04 CA E9
  
```

EN [Icons] 14:31 27/08/2014

# File .264 Analyse

Vega TSA - S:\exchange\\_EXCHANGE\\_vicheka\KK\_INTERNSHIP\_FILES\kk\_active\_hd1\_dec\_h264\dec\_h264\src\kk\_test\_hp\_skip.264

File View Navigate Settings Window Help

Decoding Order [Navigation Icons] AU Num [Dropdown]

0 F IDR 1 F P 2 F IDR 3 F P 4 F IDR

00:00:00:000 POC = 0  
00:00:00:033 POC = 2  
00:00:00:066 POC = 0  
00:00:00:100 POC = 2  
00:00:00:133 POC = 0

Sps

Field Value

Profile	High
Level	3.0
Chroma Format	[4:2:0]
MBAFF Mode	FALSE
Resolution	352 x 288...
Max DPB Size	16
Coding Type	CAVLC
BitDepth Luma	8
BitDepth C...	8
Frame Rate...	30.000000...
AUs in Seq# 1	2
AUs in Seq# 2	2
AUs in Seq# 3	1

Seq# 1-3

Legend: I (orange), P (purple), B (blue), Skip (green), IPCM (grey), IBL (pink)

FW\_MV BW\_MV

Field Value

macroblock	
index	3(0x3)
StartAddress(bits)	197004(0x3018c)
Length(bits)	12(0xc)
frame/field	frame
sliceNo	0(0x0)
slice_type	P_slice
mb_skip_run	0(0x0)
mb_skip_flag	
mb_type	P_I0_16x16
mb_field_decoding_flag	false
mb_qp_delta	0(0x0)
qp	15(0xf)
coded_block_pattern	16:: 010000
CodedBlockPatternLuma	0:: 0000
CodedBlockPatternChroma	1:: 01
transform_size_8x8_flag	false
base_mode_flag	
residual_prediction_flag	
intra_mb_ored	

Vega TSA      Deblocking Filter As Per Stream / YUV Dumping Disabled      00:00:00:133      100%      AU: 1      Slice: 0      MB: 3

25 errors per page      Start AU      Hex      Binary      Edit Mode      ASCII

Number	File-offset	Stream-location	Section	Description
244	N.A		N.A.	Stream Parsing Started .....
245	N.A		N.A.	Stream Parsing End .....

```

00006030 58 BE B5 F5 A9 3A BF D5 EF AE 55 D7 BE BD 7D 5B EA DF 5F 5F 56 AE AC FA
00006048 D7 D6 72 5E F7 43 BE AC 45 75 F1 11 C1 27 33 CE 20 F7 C2 3C 38 E5 43 32
00006060 11 0C 96 3D A2 BE BA 43 08 66 2F BB 37 47 E3 2C E5 1A 19 9D DC 10 09 D0
  
```

EN      14:37      27/08/2014



# Lattice FPGA LFE3-150EA-6FN1156C

**Project Properties**

h264\_dec

- LFE3-150EA-6FN1156C
  - h264\_dec
    - h264\_dec/source/bit\_buffer.v
    - h264\_dec/source/bit\_buffer\_if.v
    - h264\_dec/source/bsd.V
    - h264\_dec/source/h264\_function\_p
    - h264\_dec/source/master\_fsm.v
    - h264\_dec/source/mb\_hdr.v
    - h264\_dec/source/mb\_residual.v
    - h264\_dec/source/nC\_compute.v
    - h264\_dec/source/nnz\_1024x5\_dp\_1
    - h264\_dec/source/pps.v
    - h264\_dec/source/slice\_header.v
    - h264\_dec/source/sps.v
    - h264\_dec/source/start\_code.v
    - h264\_dec/source/tb\_bsd.V
    - h264\_dec/source/tb\_nc\_compute

Part Name: LFE3-150EA-6FN1156C

Family: LatticeECP3

Device: LFE3-150EA

Performance grade(\_): 6

Package type: FPBGA1156

Operating condition: COM

[Online Data Sheet for Device](#)

**Device Information:**

Voltage:	1.2V
LUT:	149040
Registers:	111780
EBR Bits:	6.85M
EBR Blocks:	372
Dist RAM:	296.25K
DSP:	80
PLL:	10
DLL:	2
PCS:	4
PIO Cells:	644
APIO:	72
PIO Pins:	586

OK Cancel

# Simulation results of block SPS in ActiveHDL

