FINAL YEAR PROJECT
FINAL PRESENTATION

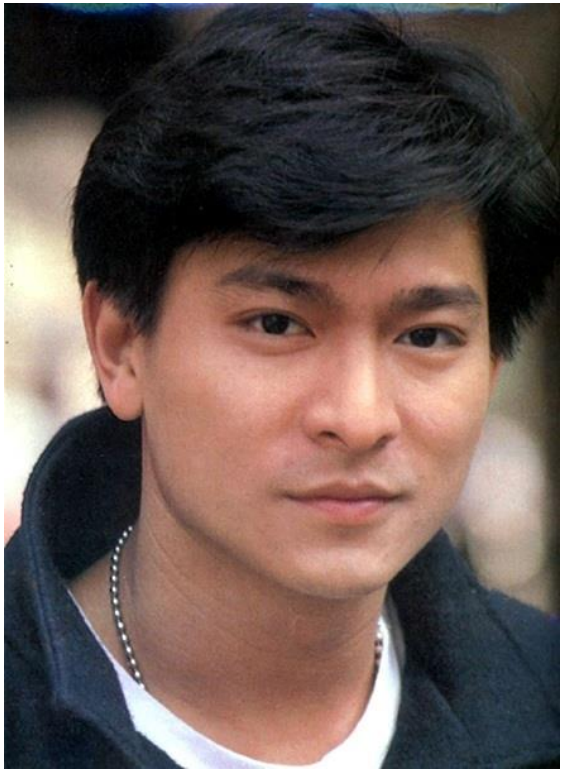
HARDWRAE-BASED FACE DETECTION

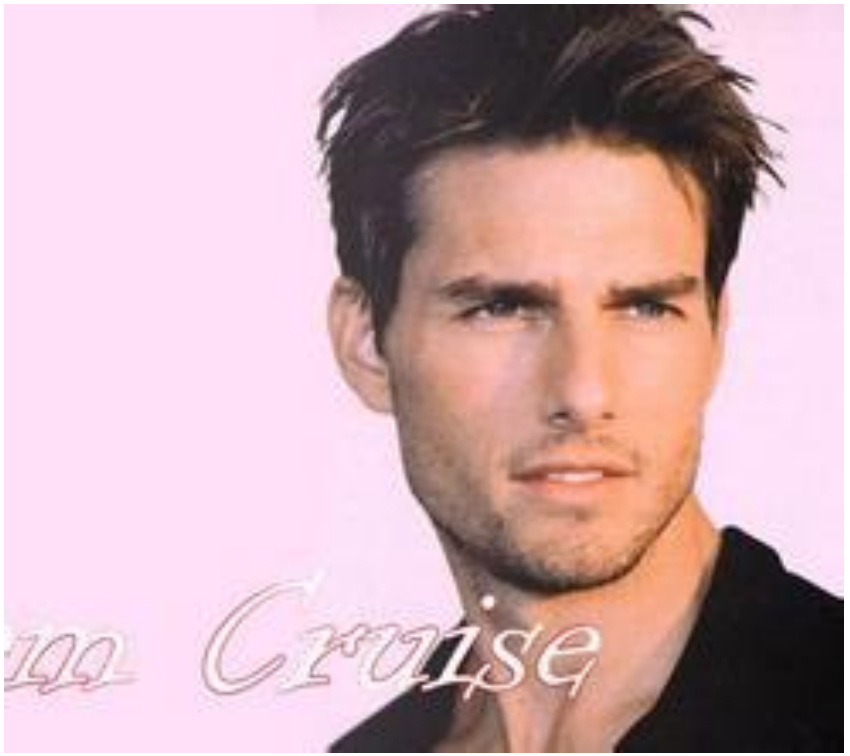
Supervisor: Dr. Benjamin Carrion Schafer
Student: Zhang Yunlei

Contents

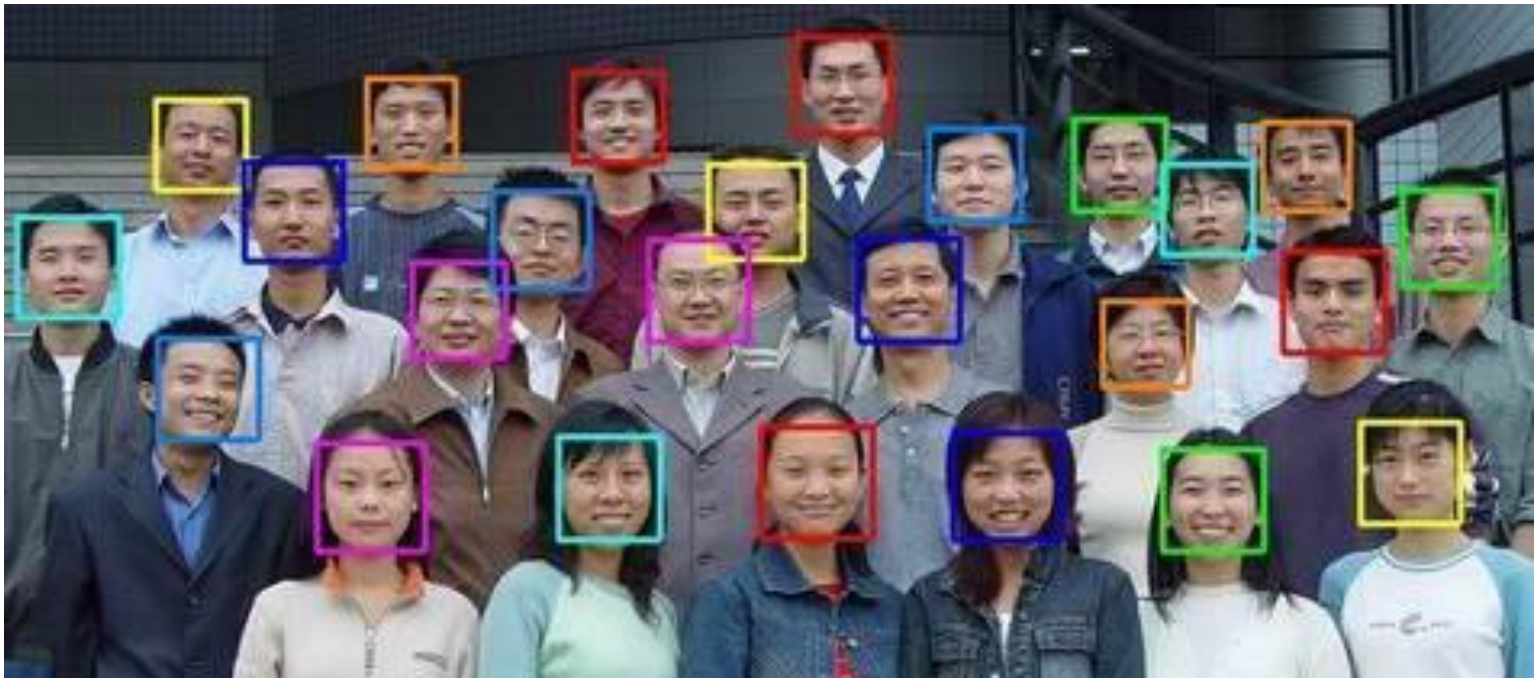
1. Introduction

Face is Identity





Face Detection (Location) is basis



Why Use Color Detection

- Three key points: speed, accuracy, stability
- Many methods: Principal Component Analysis (PCA), Neural Networks (NN), Support Vector Machines (SVM), Hough Transform (HT), Geometrical Template Matching (GTM), color analysis,etc

Why Color Detection?

Why Use Color Detection

- Real-time face detection requires large data flow
- Repetitive data accessing is impossible

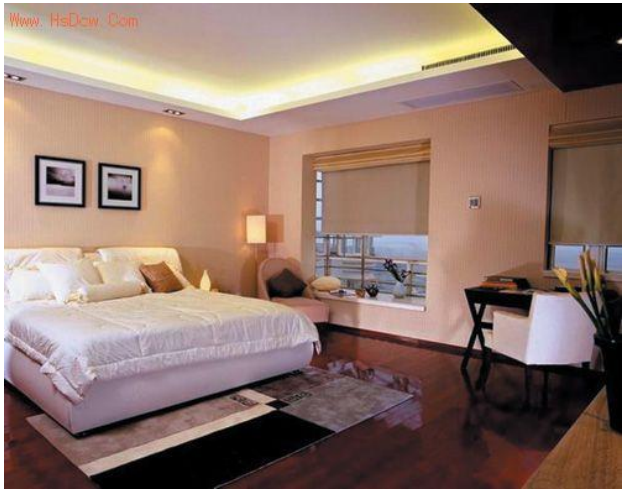
PROCESS DATA ON DATA PATH

Improve Accuracy

- However, simple color detection may not be reliable

=

OR



Improve Accuracy

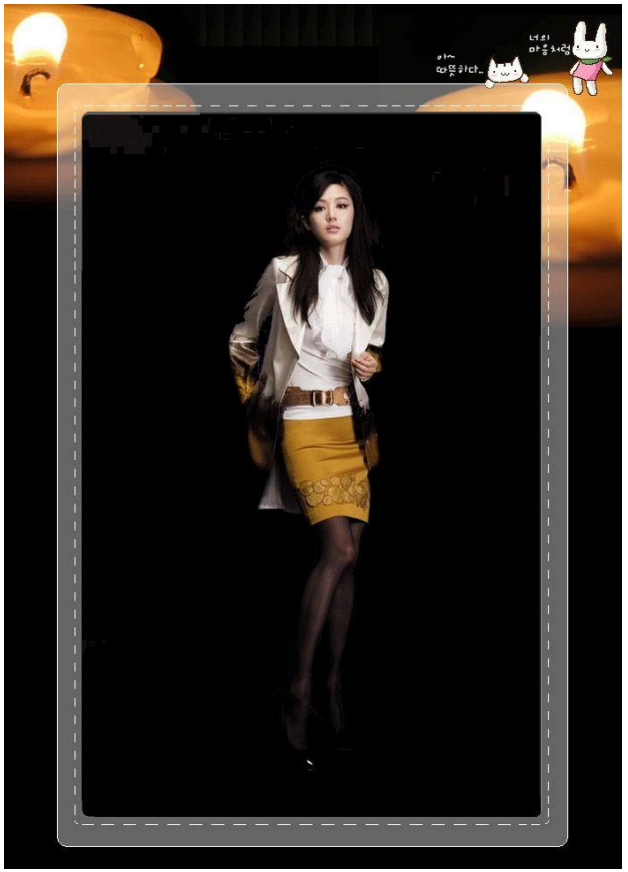
- Only head has hair on the top
- Hair helps locate face

Use hair as assistance

Improve Accuracy

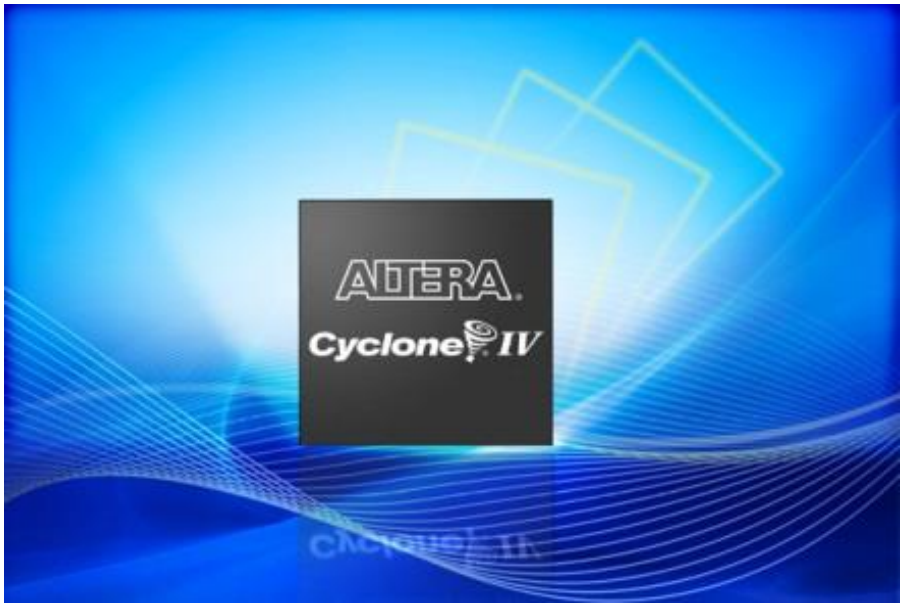
- How about black background?

Motion Detection!!



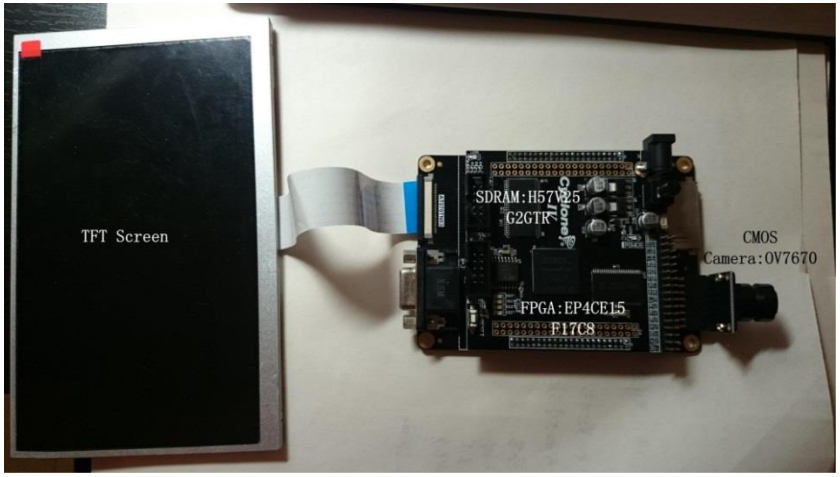
Why FPGA

- Parallel processing
- More powerful functions in further research



1. Introduction

- Hardware:
  - Altera Cyclone IV (EP4CE15F17C8)
  - TFT Screen (AT070TN83) (800*480, RGB565)
  - CMOS Camera (OV7670) (640*480, RGB565)
  - SDRAM(H57V2562GTR)
- Software:
  - Quartus II
  - Modelsim
- HDL:
  - Verilog



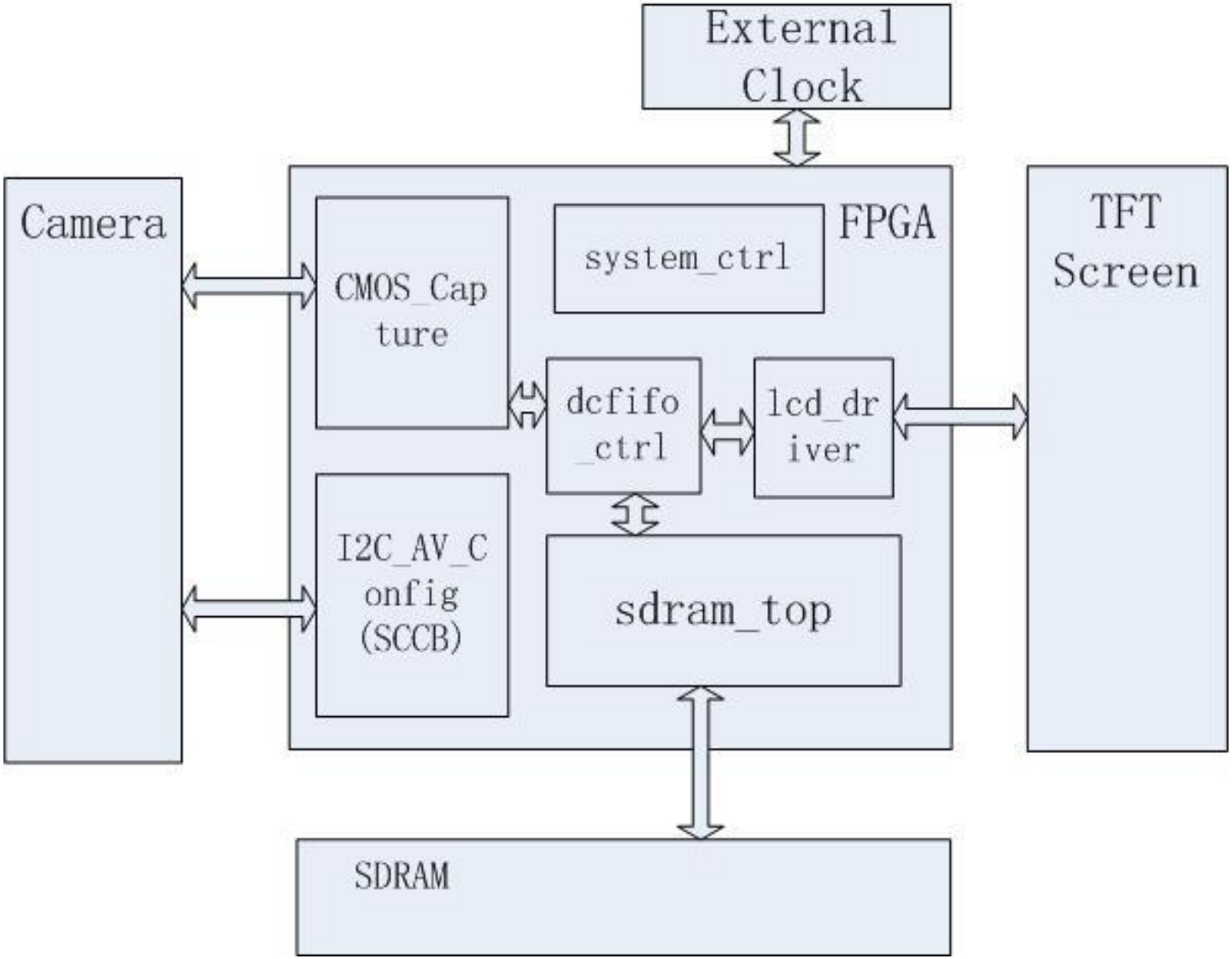
2. Design Methodology & Result

2.1 Design Objective

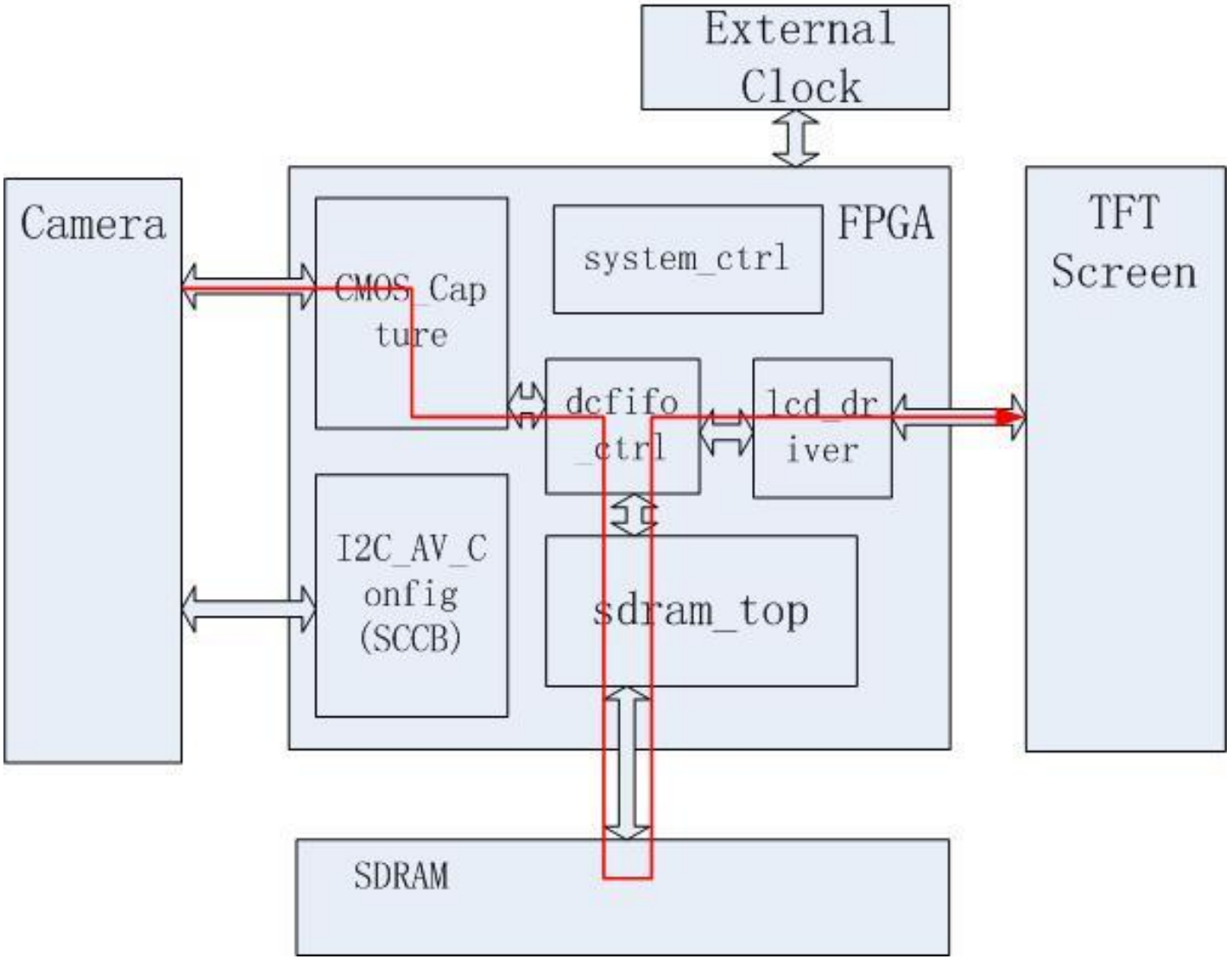
- Step 1: Real-time video display
- Step 2: Real-time face detection based on face color and hair color
- Step 3: Real-time face detection with the help of motion detection

2.2 Real-time Video Display

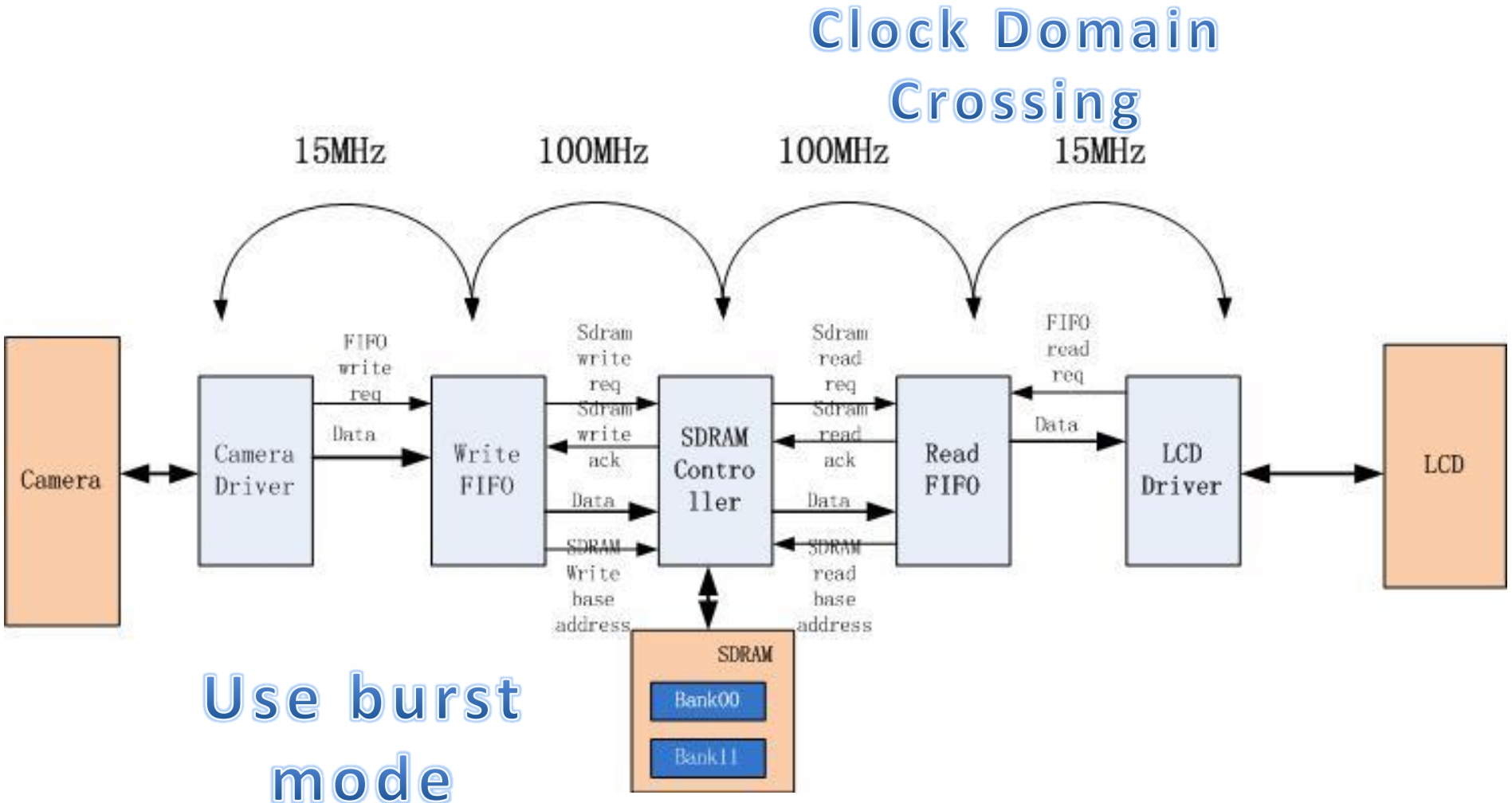
Block Diagram of Video Display Module



Data Path of Video Display Module

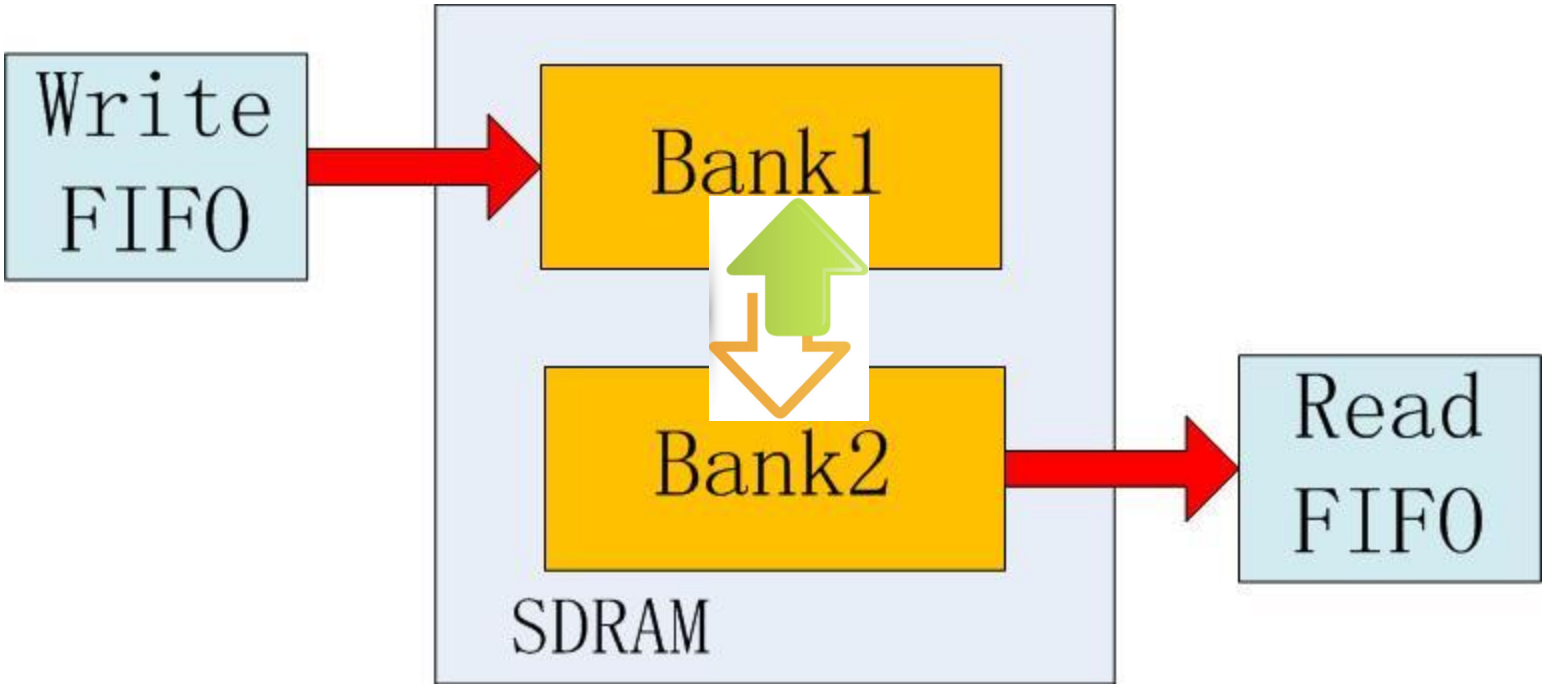


Data Path of Video Display Module



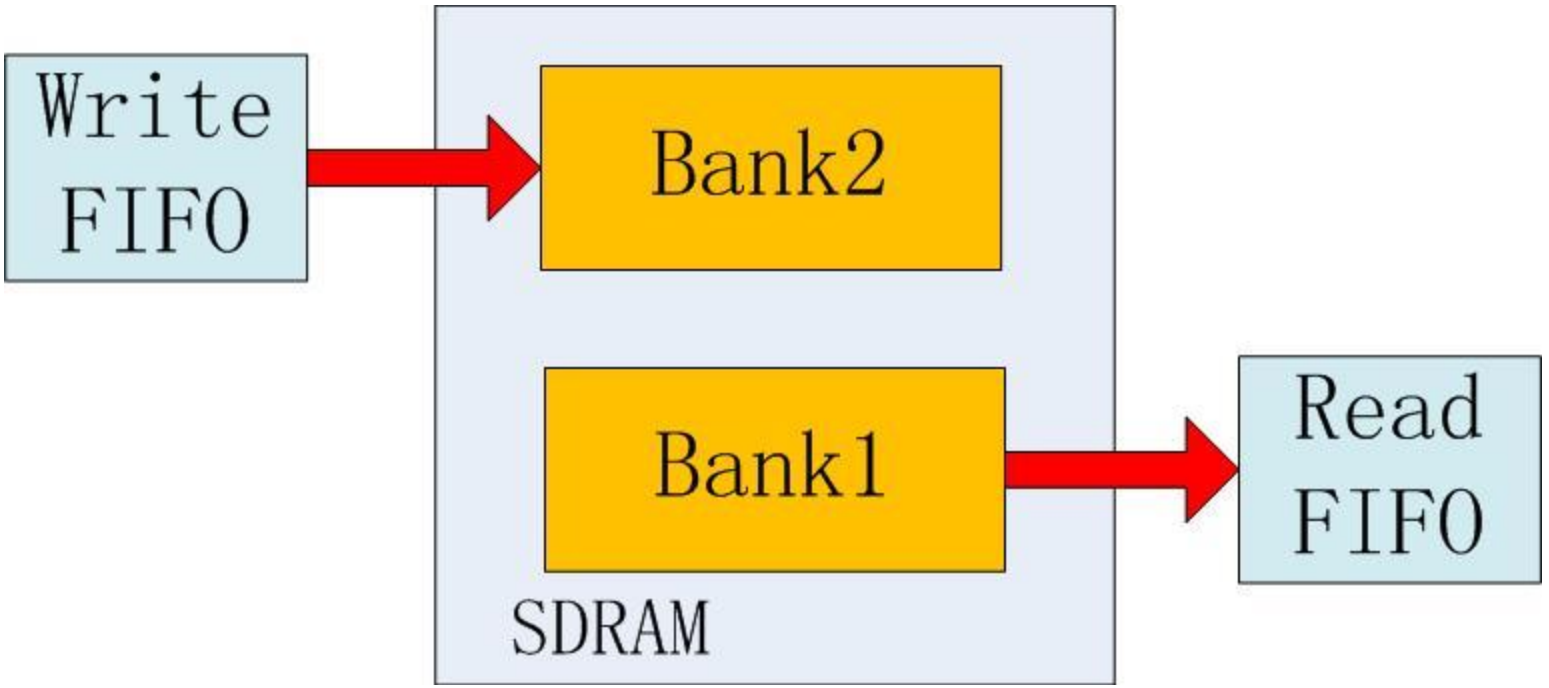
Ping-Pong Operation

- Single SDRAM based Ping-Pong Operation



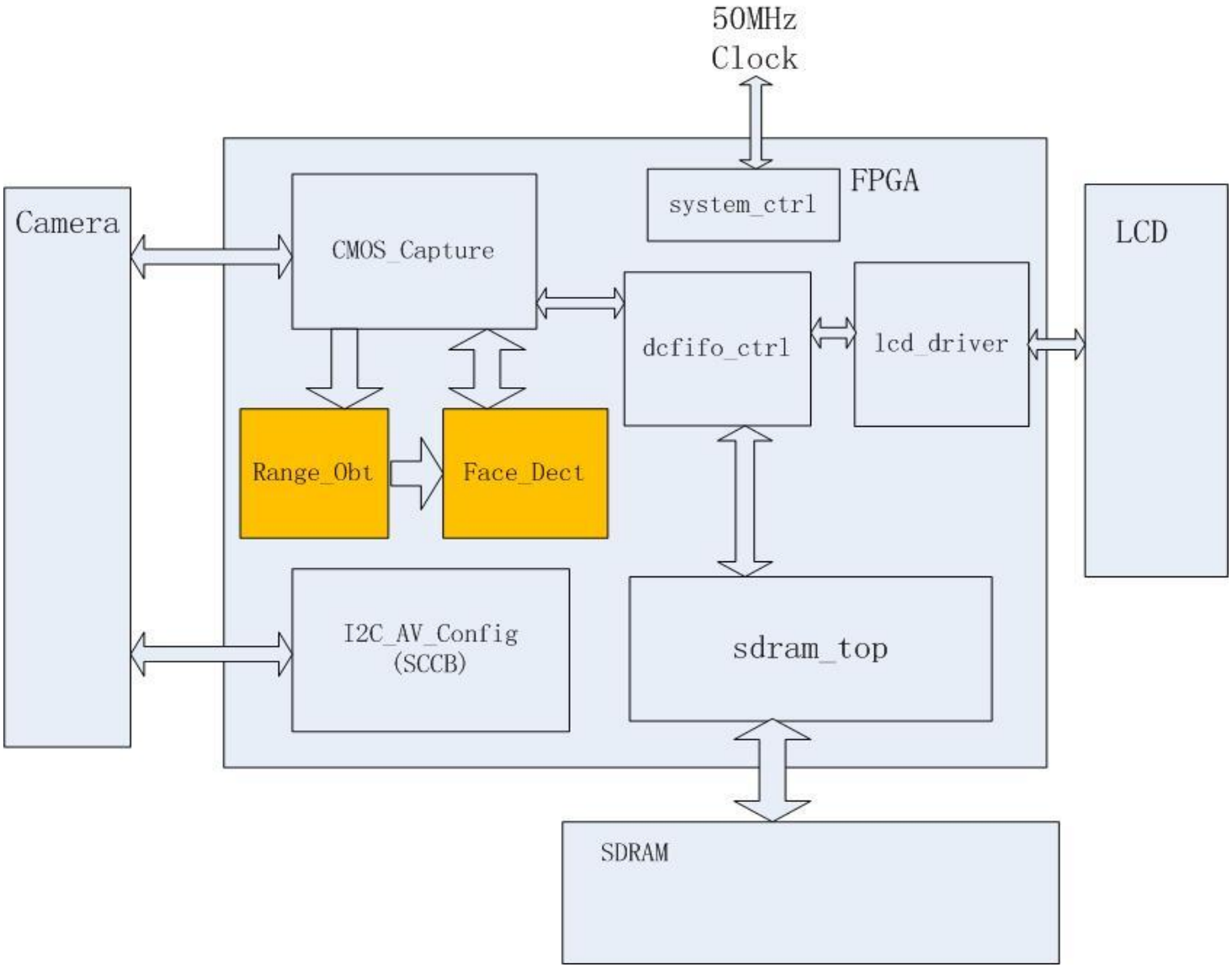
Ping-Pong Operation

- Single SDRAM based Ping-Pong Operation



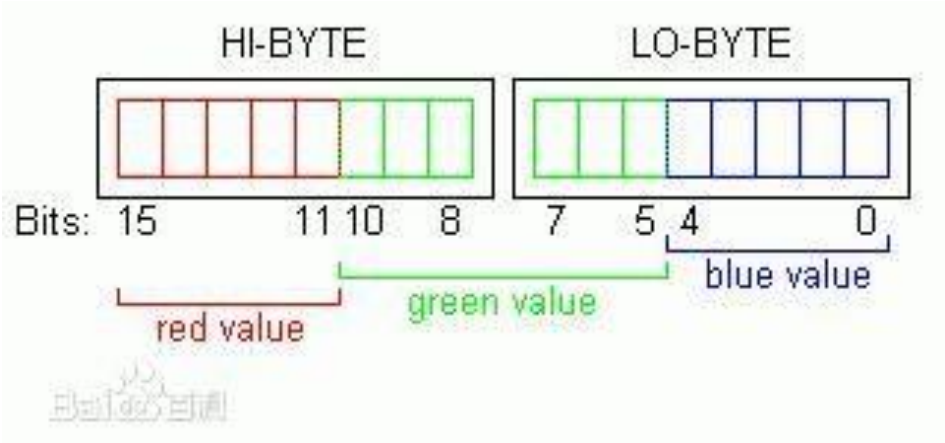
2.3 Real-time Face Detection Based on Face Color and Hair Color

Block Diagram of Face Detection Module



Data Format and Color Space

- Simple face color detection
- RGB565



- Range for R,G,B: [0, 31]
- Selected face color space:

$$5 < R < 17 \,\&\, 5 < G < 17 \,\&\, 5 < B < 17 \,\&\, R - B > 1 \,\&\, R - G > 1$$

Color Space Testing

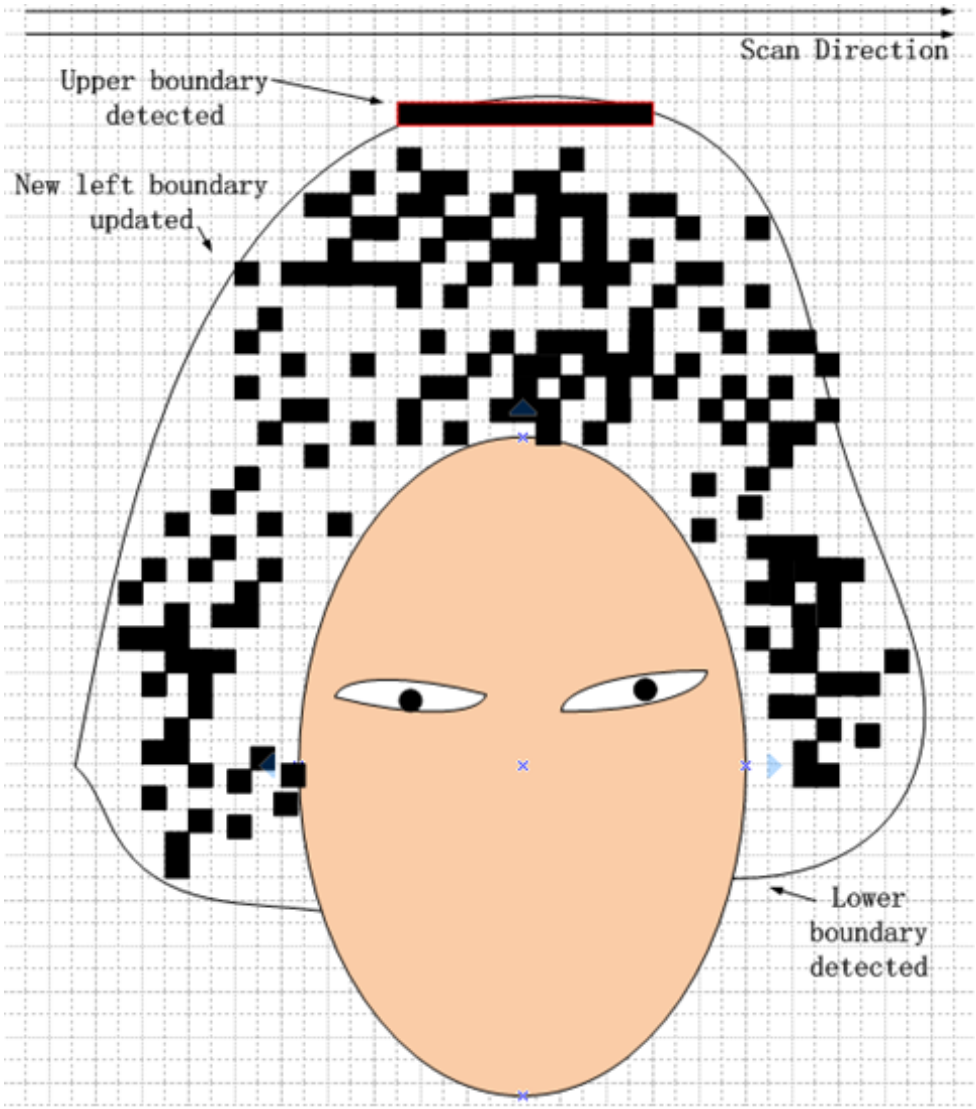
Color Space of Hair

- Hair color space is selected as following

$$R < 9 \,\&\, G < 9 \,\&\, B < 9;$$

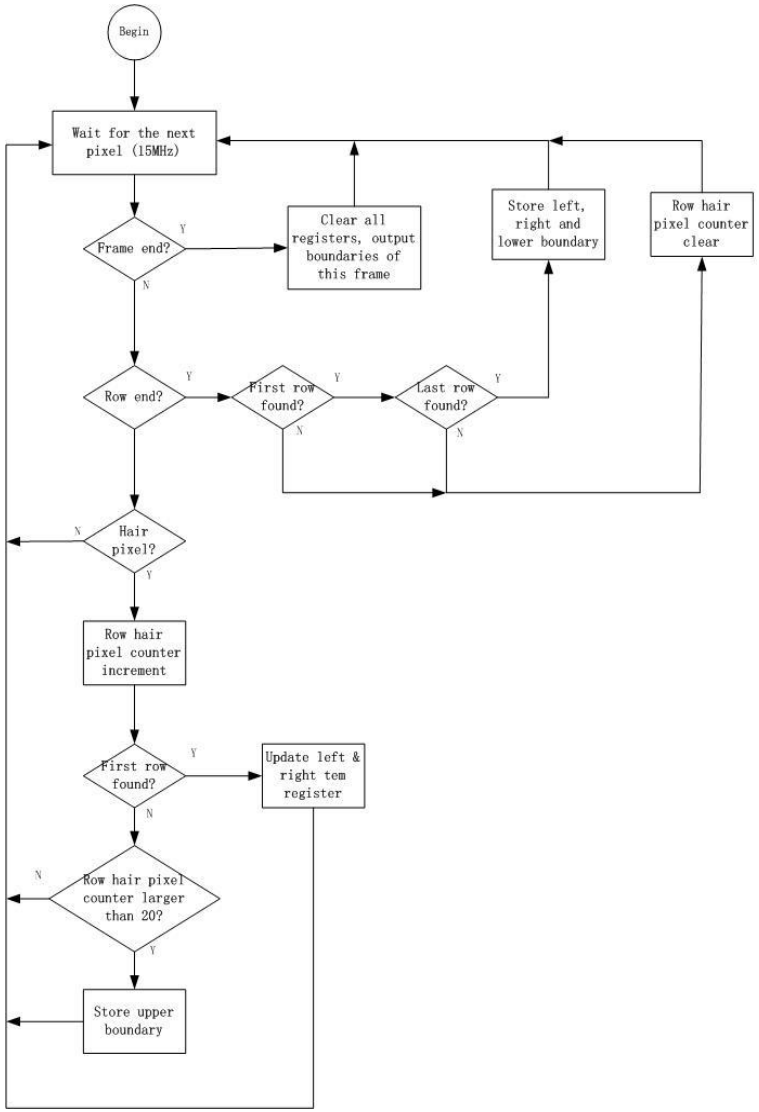
$$|R - G| <= 1 \,\&\, |R - B| <= 1 \,\&\, |B - G| <= 1;$$

Face Boundary Obtain



Face Boundary Obtain

- Flow chart of hair boundaries obtain module



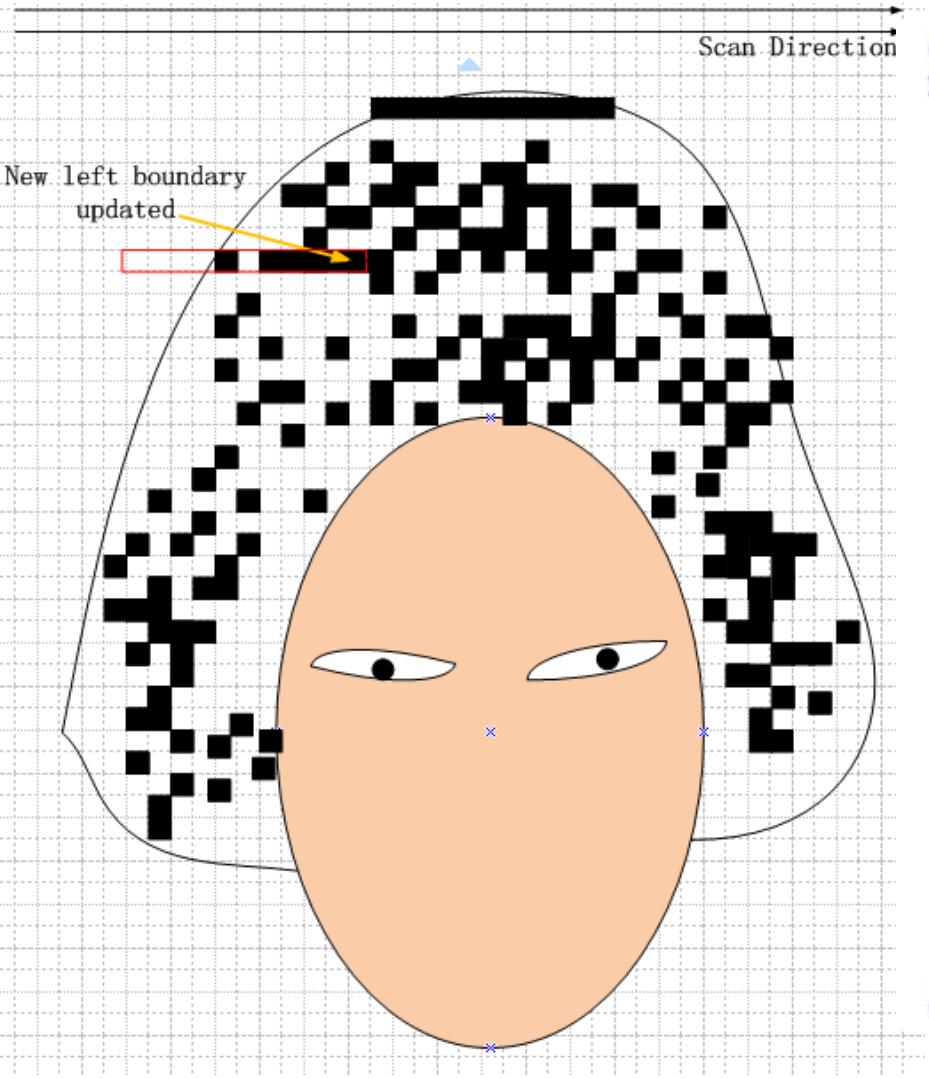
Eliminate Noise

log: 2015/04/13 20:01:56 #0

click to insert time bar

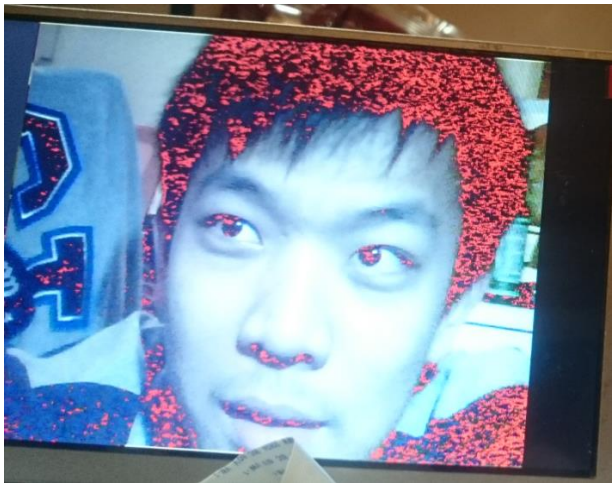
Type	Alias	Name	-511	-510	-509	-508	-507	-506	-505	-504	-503
		out_left					26				

Eliminate Noise




```
else if (dataen)//when frame and row effective
begin
   if(red < 9 && green < 9 && blue < 9 && (red - green < 2 || g
   begin
      cont <= cont + 10'd1; //if yes, cont++
      shift_cout <= {shift_cout[8:0], 1'b1};
      if (state == 2'd0)//when find it is the first time in a f
      begin
         if (cont >= contnum)
         begin
            boundary_up <= cont_row_num;
            state <= 2'd1;
         end
      end
      else
         if (cont_col_num < left && shift_cout_sum >= 5) //get
            left <= cont_col_num;
         else if (cont_col_num > right && shift_cout_sum >= 5)
            right <= cont_col_num;
   end
   else
   begin
      shift_cout <= {shift_cout[8:0], 1'b0};
   end
   shift_cout_sum <= {3'd0, shift_cout[0]} + {3'd0, shift_cout[
end
```

Eliminate Noise



Type	Alias	Name	-510	-509	-508	-507	-506	-505	-504
		out_left				212			

Type	Alias	Name	-510	-509	-508	-507	-506	-505	-504	-503
		out_right				639				

Type	Alias	Name	-510	-509	-508	-507	-506	-505	-504	-503
		out_up				0				

Type	Alias	Name	-510	-509	-508	-507	-506	-505	-504	-503
		out_down				257				

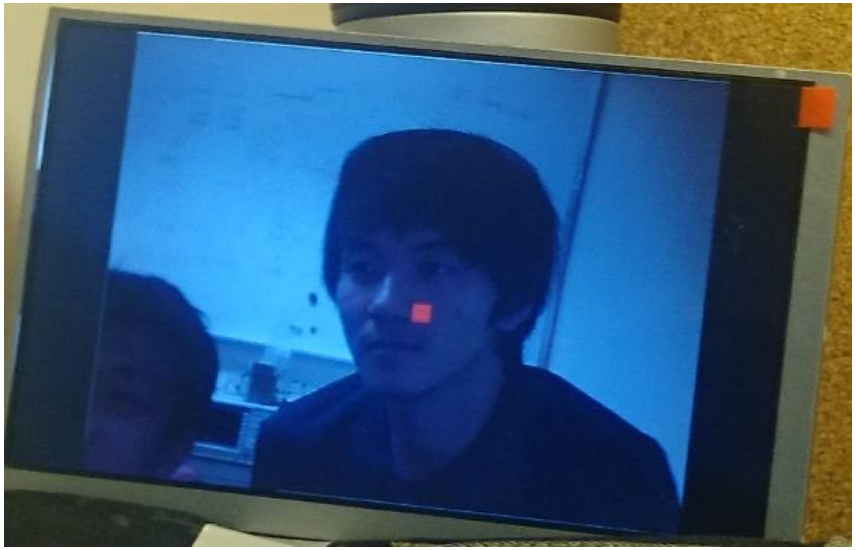
Find Center Point of Face

```
else
begin
   if (frame_done_fall)//what to do when one frame ends: row++ clear column
   begin
      num <= 26'd0;
      whole_col <= 26'd0;
      whole_row <= 26'd0;
   end
   else if (dataen)//when frame and row effective
   begin
      if(red <= 10'd17 && red > 10'd5 && green < 10'd17 && green > 10'd5 && blue < 10'd17 && blue > 10'd5 && red - b
      begin
         whole_col <= whole_col + cont_col_num;
         whole_row <= whole_row + cont_row_num;
         num <= num + 1;
      end
   end
end
```

```
if ((column_num - centre_column < 10 || centre_column - column_num < 10) && (row_num - centre_row < 10 || centre_row - row_num < 10) )
//if ((column_num < 320 && column_num > 300) && (row_num < 240 && row_num > 200) )
begin
   CMOS_oDATA[15:0] <= 16'b11111_000000_00000; //show the centre point of face and mark it with red color
end
```

Result

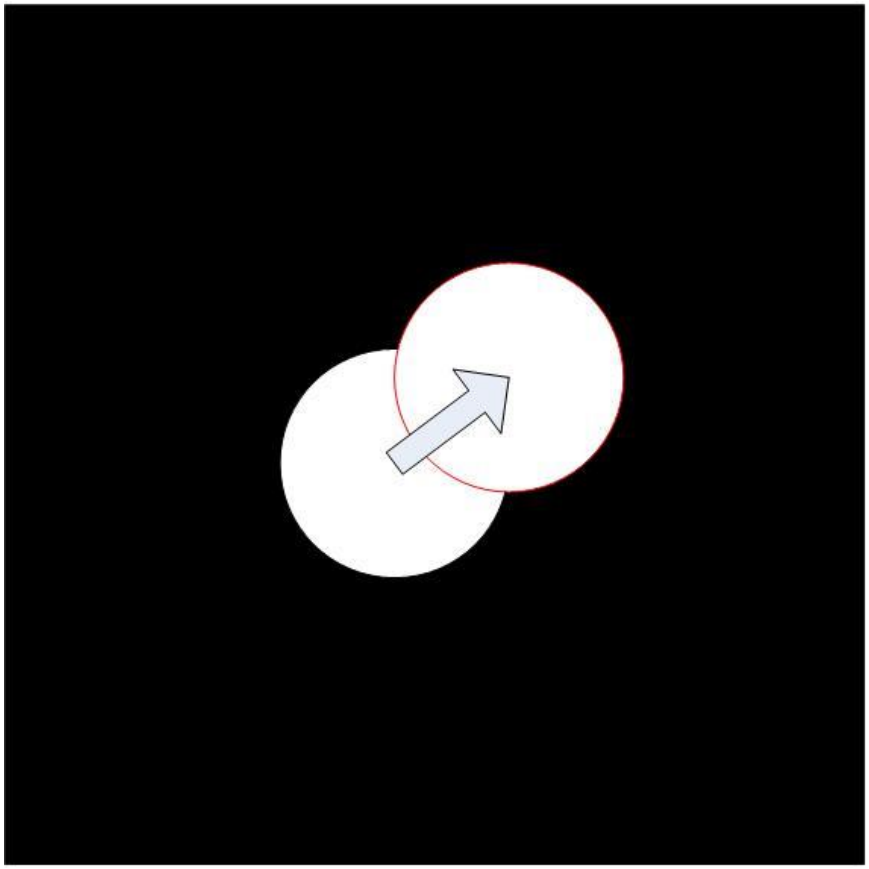
Result

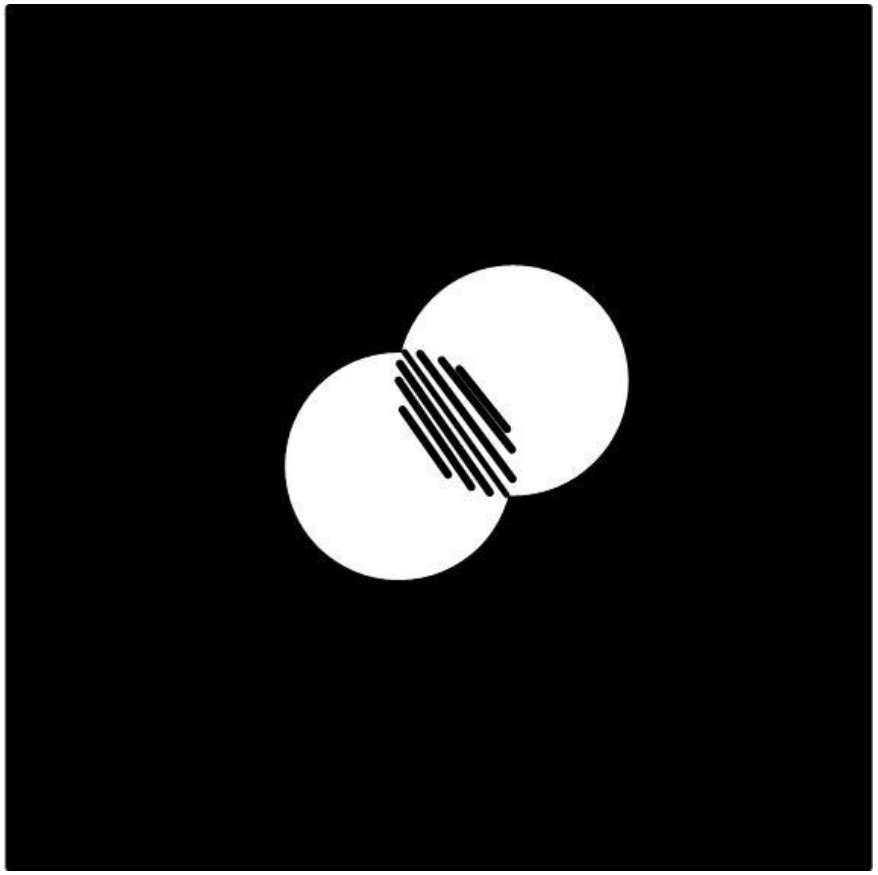


2.4 Higher Accuracy with the Help of Motion Detection

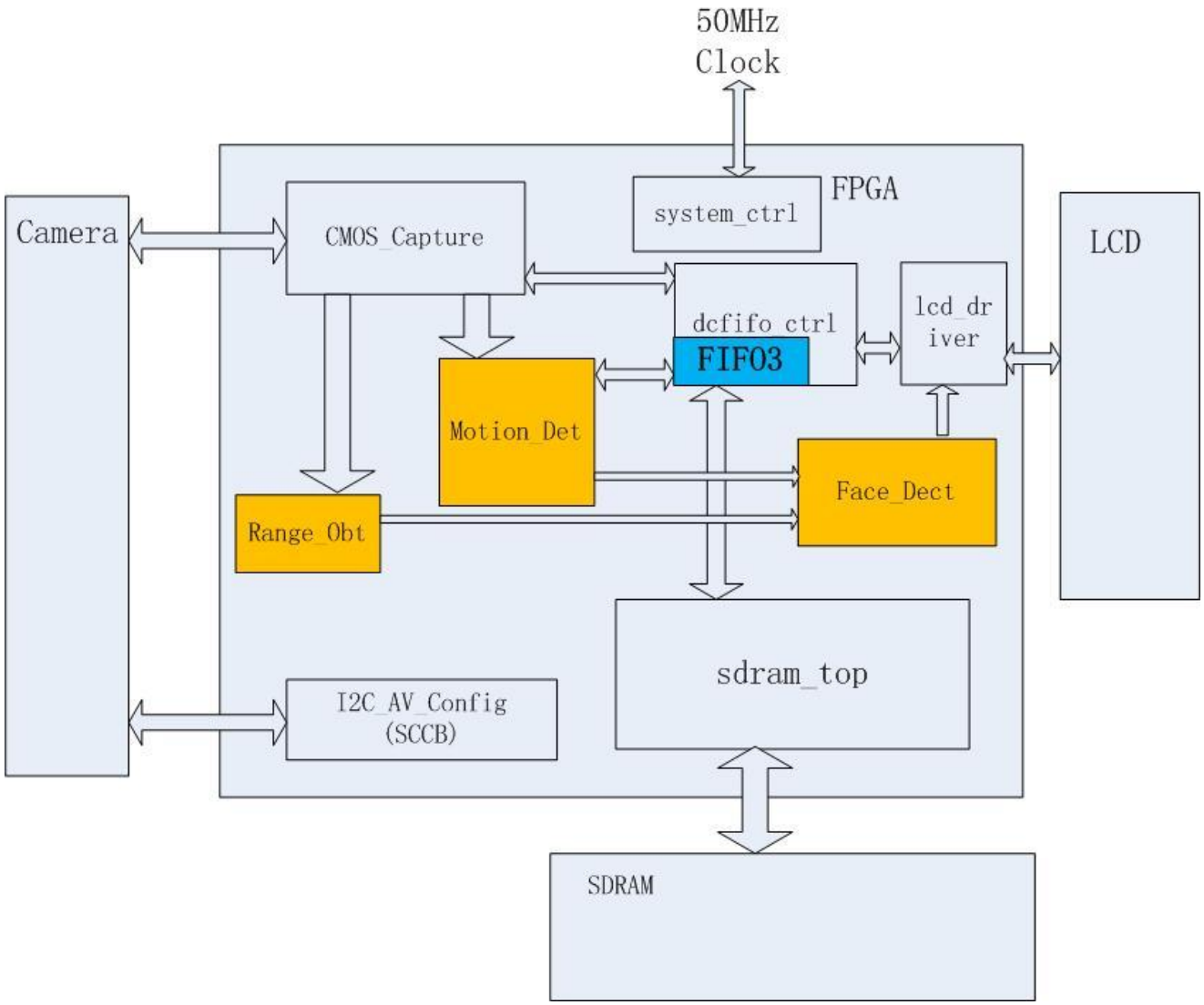
Function

- Remove Unchanged Background



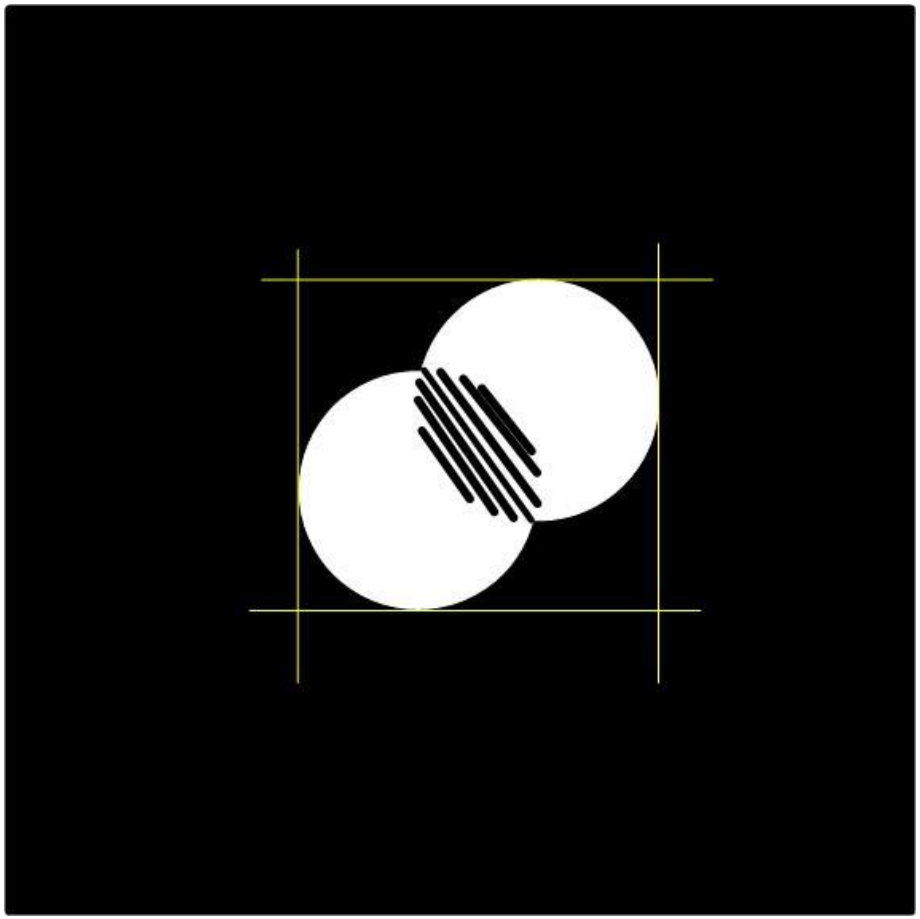


Block Diagram with Motion Detection Module



Construction of Motion Detection Module

- Similar to Range_Obt module



3. Reflection and Outlook

FPGA Report

Modify algorithm to reduce resource consumption

Flow Status	Successful - Tue Apr 14 12:04:35 2015
Quartus II 64-Bit Version	12.0 Build 178 05/31/2012 SJ Full Version
Revision Name	sdram_ov7670_vga
Top-level Entity Name	sdram_ov7670_vga
Family	Cyclone IV E
Device	EP4CE15F17C8
Timing Models	Final
Total logic elements	2,992 / 15,408 (19 %)
Total combinational functions	2,848 / 15,408 (18 %)
Dedicated logic registers	721 / 15,408 (5 %)
Total registers	721
Total pins	84 / 166 (51 %)
Total virtual pins	0
Total memory bits	16,384 / 516,096 (3 %)
Embedded Multiplier 9-bit elements	0 / 112 (0 %)
Total PLLs	1 / 4 (25 %)

Flow Status	Successful - Sat May 09 23:24:00 2015
Quartus II 64-Bit Version	12.0 Build 178 05/31/2012 SJ Full Version
Revision Name	sdram_ov7670_vga
Top-level Entity Name	sdram_ov7670_vga
Family	Cyclone IV E
Device	EP4CE15F17C8
Timing Models	Final
Total logic elements	3,519 / 15,408 (23 %)
Total combinational functions	3,215 / 15,408 (21 %)
Dedicated logic registers	1,181 / 15,408 (8 %)
Total registers	1181
Total pins	93 / 166 (56 %)
Total virtual pins	0
Total memory bits	53,248 / 516,096 (10 %)
Embedded Multiplier 9-bit elements	0 / 112 (0 %)
Total PLLs	1 / 4 (25 %)

Determine Color Space

- Use logic analyzer to transmit data to computer
- Find the typical color space of hair and face for this camera
- Find different sets of color space in different light conditions
- Find different sets of color space for people of different hair and skin color

Support More Faces

- More faces can be supported by calculating the density of hair/face pixels.

4. Conclusion

- 1. In this project, a face detection system is built on FPGA, utilizing its parallel processing nature.
- 2. Color detection approach is used in this project
- 3. Hair is used as boundaries of face, to filter the noise in the background.
- 4. Motion detection helps to filter the hair color noise in the background.
- 5. Several work are involved in future research: color space determination, supporting more faces and improve algorithms.

Q&A

Thank you!