FPGA-based Voice Visualization

Supervisor: Dr. Benjamin Carrion Schafer

Student: Gao Zhendong

Date :April 13, 2015

CONTENTS

Background and Objectives

Background:

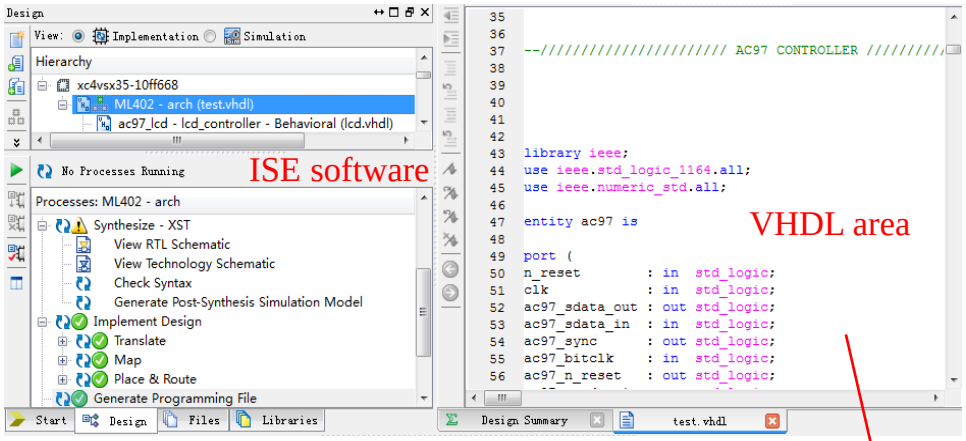
- Sound processing is used to many fields.
- FPGA are popular platforms for high-performance HW designs.
- The project can increase my practice ability and research capabilities which may broaden my future career.

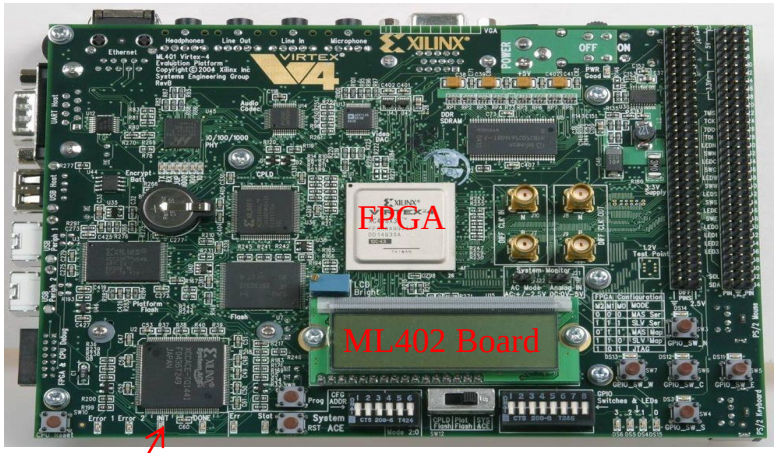
Goals of the project:

- Learn and master the use of VHDL for configuring FPGA with Xilinx ISE .
- Use FPGA to visualize in real-time the quantized speech waveform on the LCD.
- Use FPGA to visualize in real-time the quantized speech waveform on VGA monitor and changed visualization type using switches
  - Welcome message, Histogram, moving dots, sine waveform.

Tools and Platform Used

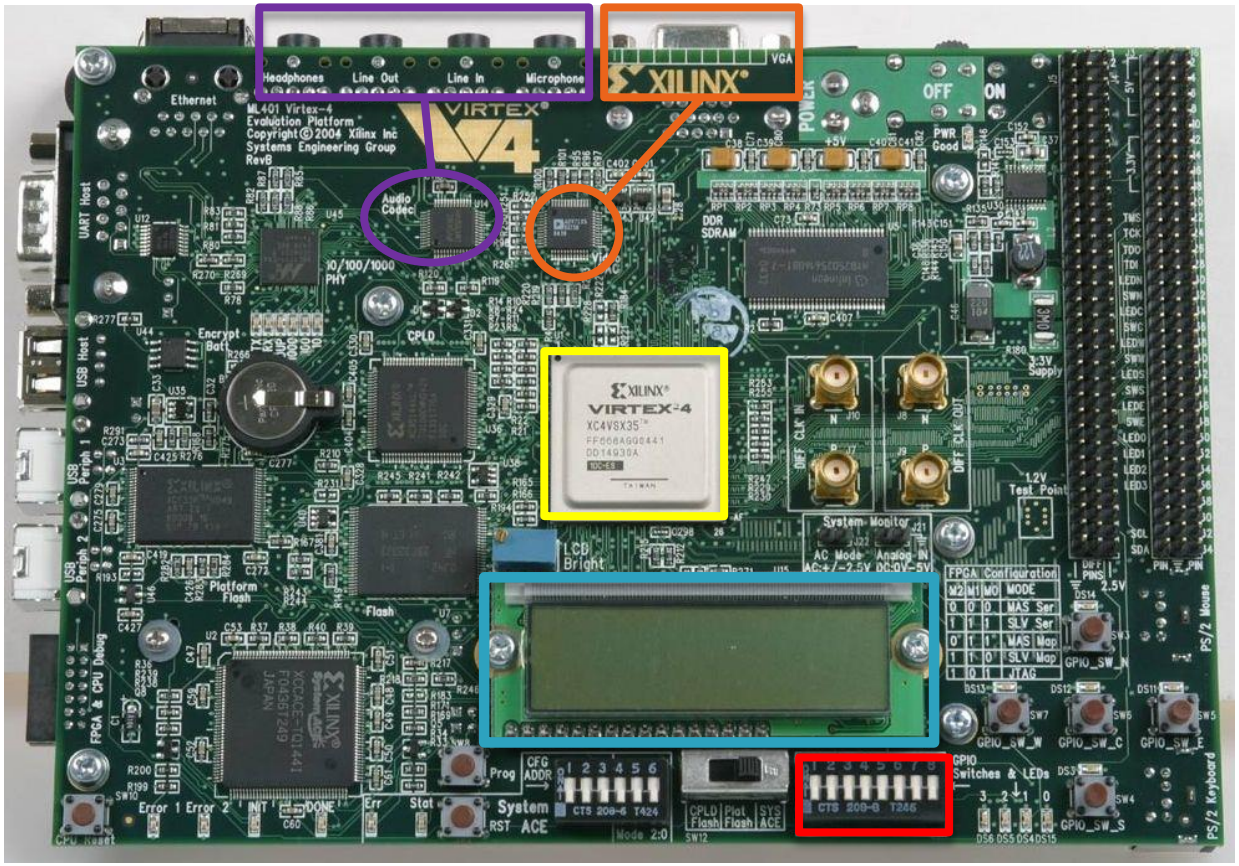
- Xilinx Virtex 4 FPGA—is an integrated circuit design that allows designers to quickly develop and prototype their HW designs.
- Xilinx ML402 board—contains single FPGA and peripherals
- ISE software—is a platform for VHDL synthesis and simulation.
- VHDL—is a popular Hardware description language





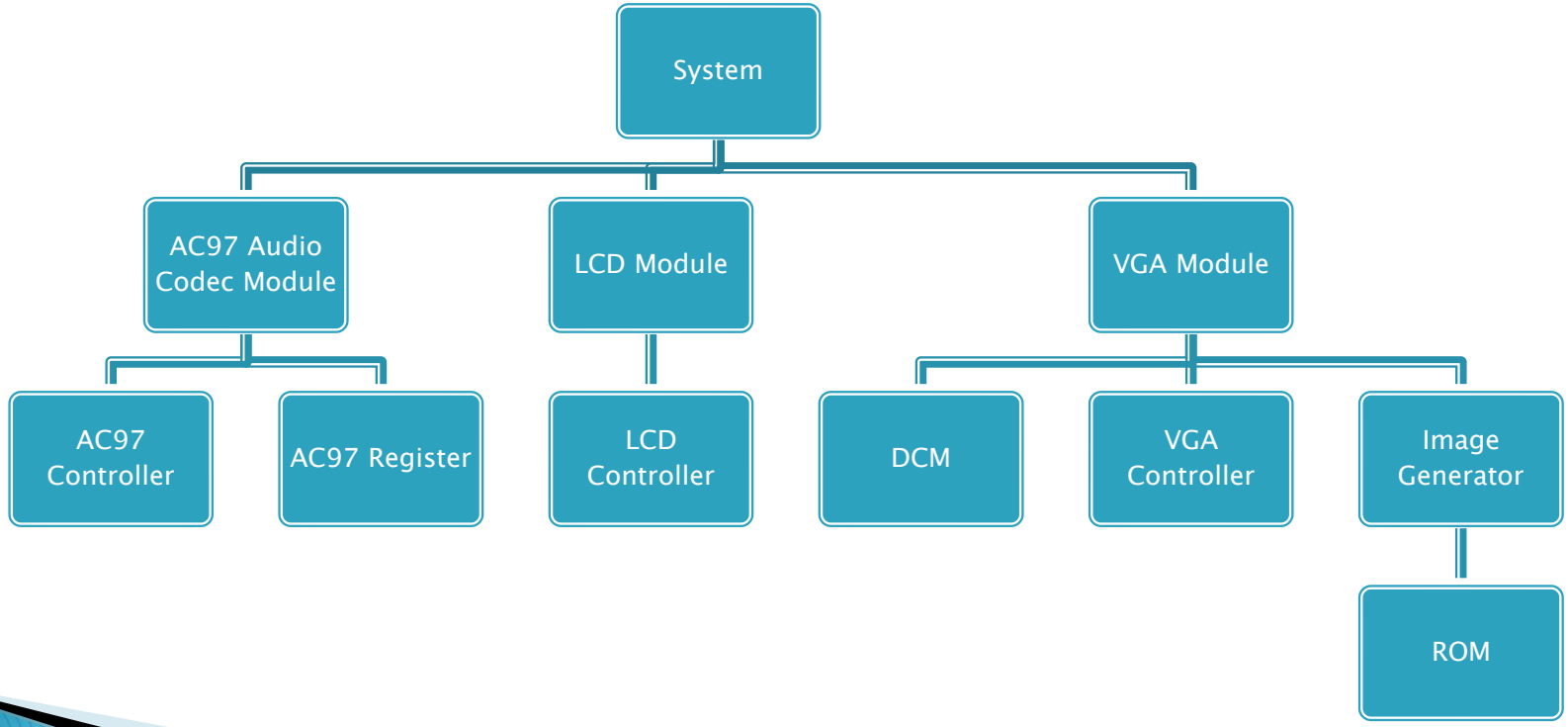
Tools and Platform Used

- AC97 Audio codec
- Video DAC
- VGA Interface
- LCD
- Switches
- FPGA



VHDL Program Structure

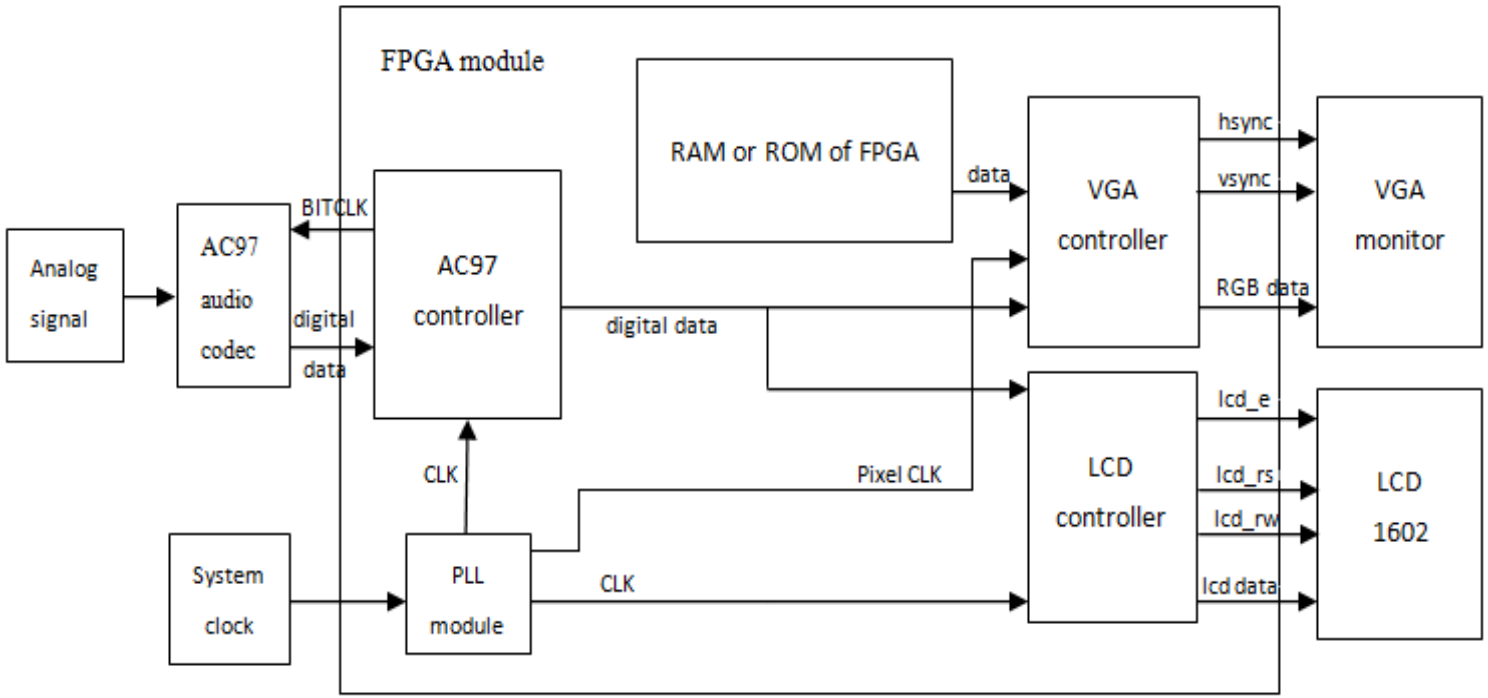
- Hierarchal design
- Top-down approach
- 9 VHDL files (include Top Module)
- 16 process
- ~6546 lines of VHDL Code



System Block Diagram

The complete system:

- Signal acquisition chip (AC97 audio codec)
- FPGA module (including PLL module)
- LCD module, ROM module and VGA controller module.
- DCM, VGA controller, image generator and ROM are categorized into VGA module.



AC97 Audio Codec Module

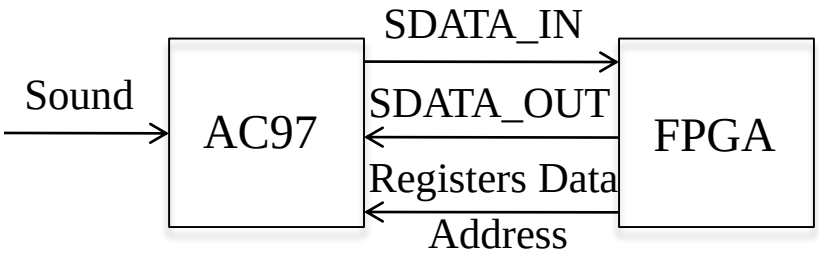
The project uses two parts (AC97 controller and AC97 cmd) to achieve AC97's work.

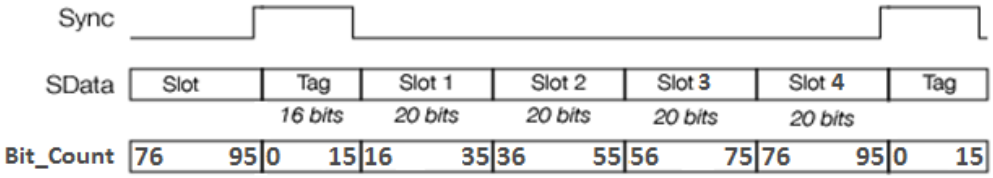
AC97 controller:

- Define slot 0 to slot 4 of AC97 link output frame and slot 3 and 4 of AC97 link input frame.
- Bit_count is used to calculate the length of each slot and assign the value in different slot.

AC97 cmd:

- Uses state machine to configure the registers of AC97.
- 12 states for configuring the registers.
- Refer to register map on AC97 data sheet.





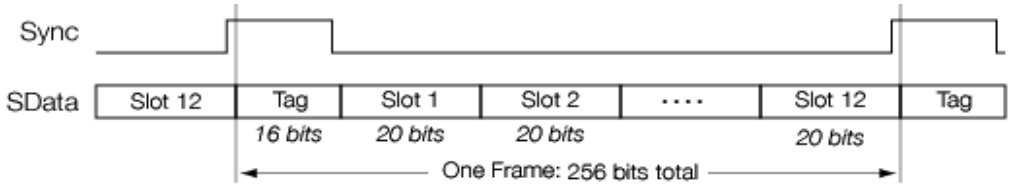
AC97 Audio Codec Module – Frame

AC97 Frames:

- It is a serial interface. Data is transmitted to and from the codec one bit at a time
- Frames are divided into twelve slots of 20 bits each, plus a 16-bit tag field.
- Slots 1 and 2 are used to read and write to configuration registers in the codec.
- Slot3 is used to send PCM data to the left channel DAC or ADC.
- Slot4 is used to send PCM data to the right channel DAC or ADC.
- Do not need to use other slots in this project.

Tag bits:

- Bit 15 is a valid flag for entire frame
- Bit 14-3 are valid flags for the individual slots in the frame.
- Bit 2-0 are zero in this project.



LCD Module

- Each character need to split into the upper 4-bits and lower 4-bits in order to be displayed.
- Users can display characters according to different ASCLL.
- There are some important interfaces of LCD needed to be mentioned briefly.

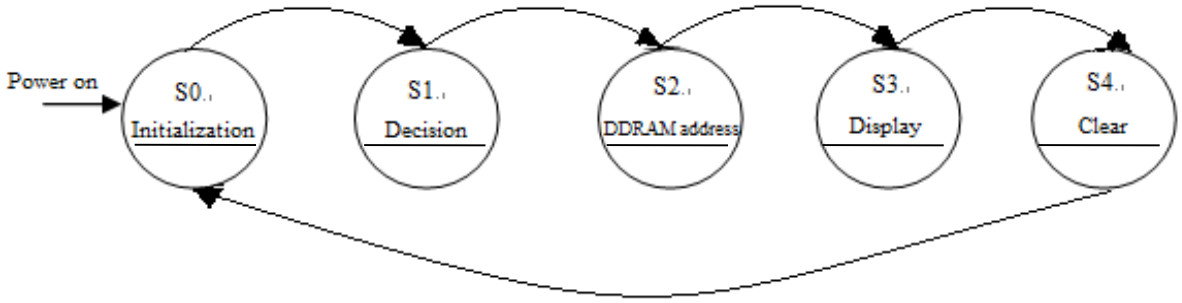
Interface Name	Mode	Description	Pin
rw	output	This interface is read/write select signal. LCD will read data during high level signal and write data during low level signal.	LCD Pin 5
rs	output	This interface is register select signal. High level signal means that data are being sent. Low level signal means that instructions are being sent.	LCD Pin 4
e	output	It is enable signal of LCD.	LCD Pin 6
lcd_data	BIDIR	It sends ASCLL values to LCD. Then LCD displays corresponding content on LCD screen.	LCD pins 7-14

LCD Module

- LCD used to receive the sampled data from AC97 and decide the range of value in order to display rough waveform on LCD screen.

A FSM is used to initialize and activate LCD and decide the range of temp:

- S0 is initialization step.
- S1 is decision of range and definition of mark signal.
- S2 is used to define the DDRAM address.
- S3 displays characters on LCD screen.
- S4 is used to clear the LCD screen and start from the initialization step (S0).



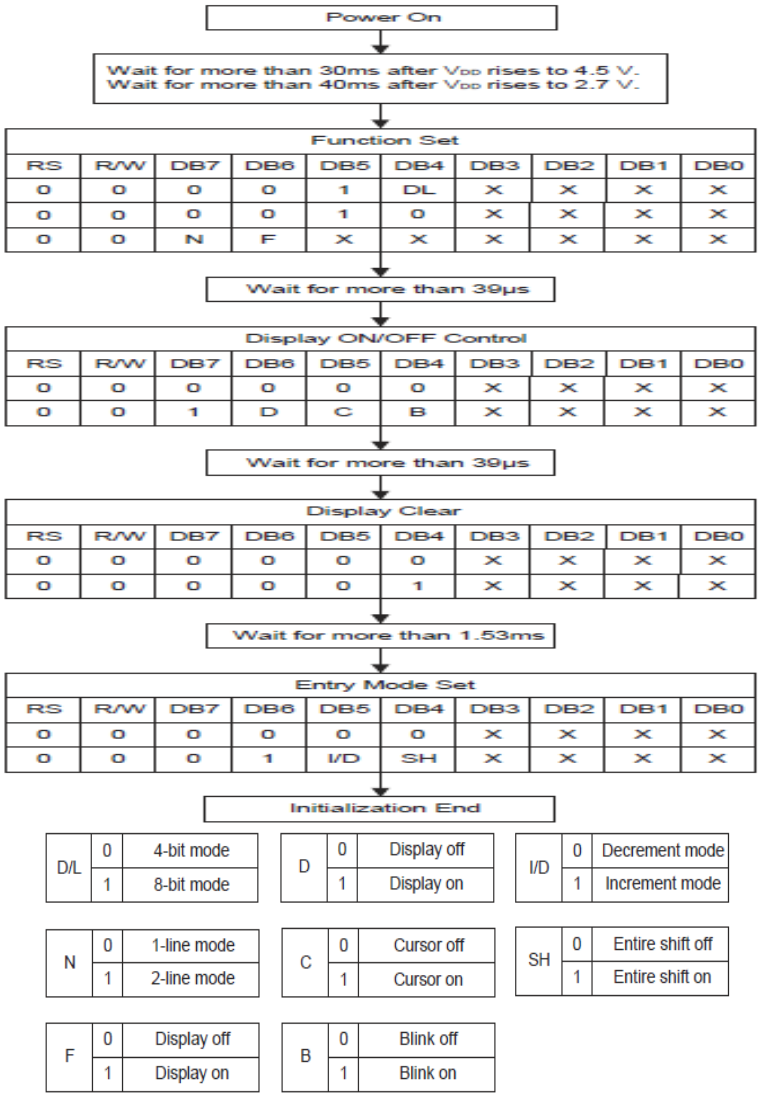
LCD Module – Initialization

Initialization processes have three steps:

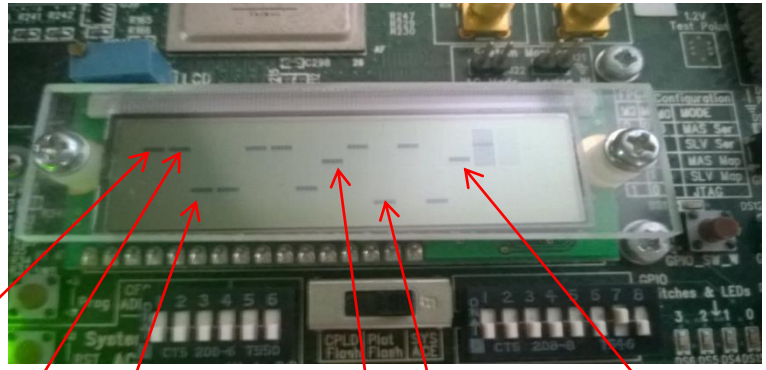
- Function set. (0010 0010 1100)
- Display ON/OFF. (0000 1111)
- Entry mode. (0000 0000)

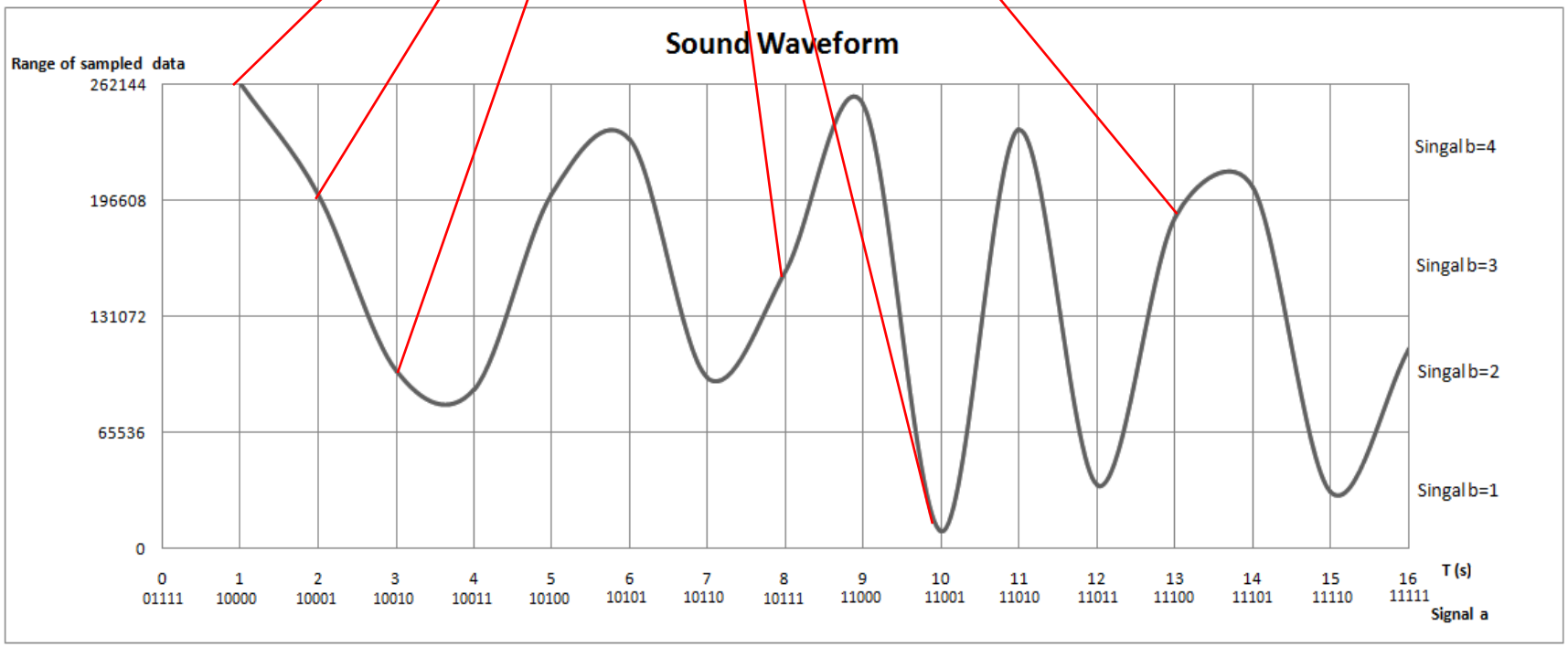
Display mode:

- LCD works in 4-bit mode.
- The whole content of LCD cannot be shifted and LCD displays 2-line with a flashing cursor on the screen.



LCD Module – Data Representation

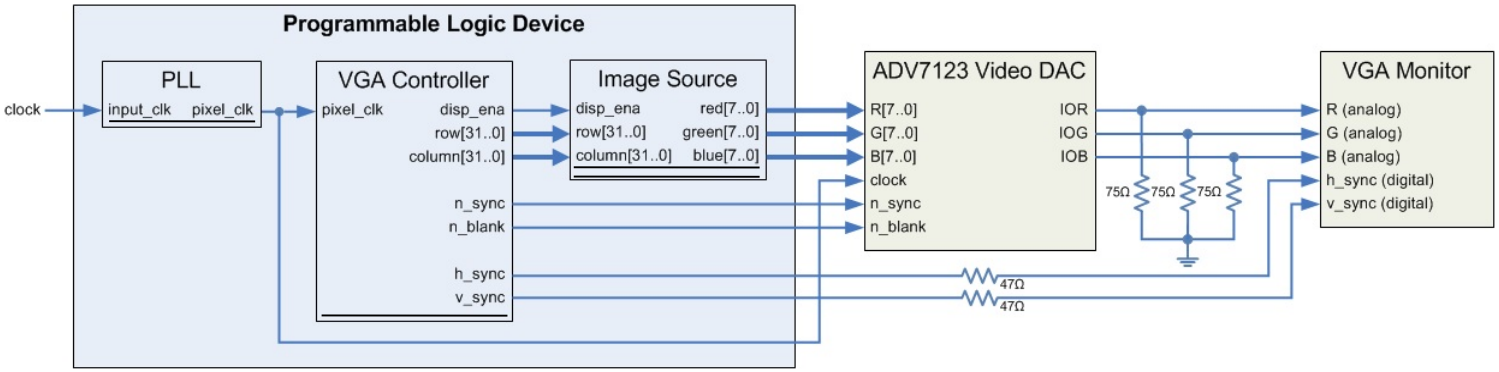




VGA Module – Block Diagram

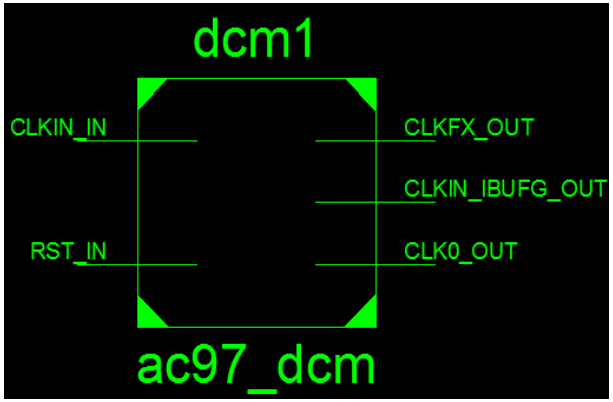
VGA module has four main parts:

- PLL (DCM) part.
- VGA Controller part.
- Image Generator part.
  - Switch is 000.
  - Switch is 001.
  - Switch is 010.
  - Switch is 011.



VGA Module – DCM Part

- Change the system clock to the desired pixel clock
- frequency depends on the monitor resolution and fps rate.
- The resolution is 1440x900 60 Hz, so the pixel clock is 106.47MHz.
- Core Generator define the DCM module as an IP using Xilinx's Core Generator.



Inputs for Jitter Calculations

Input Clock Frequency: 100 MHz

Use output frequency

MHz ns

Use Multiply (M) and Divide (D) values

M 16 D 15

Calculate

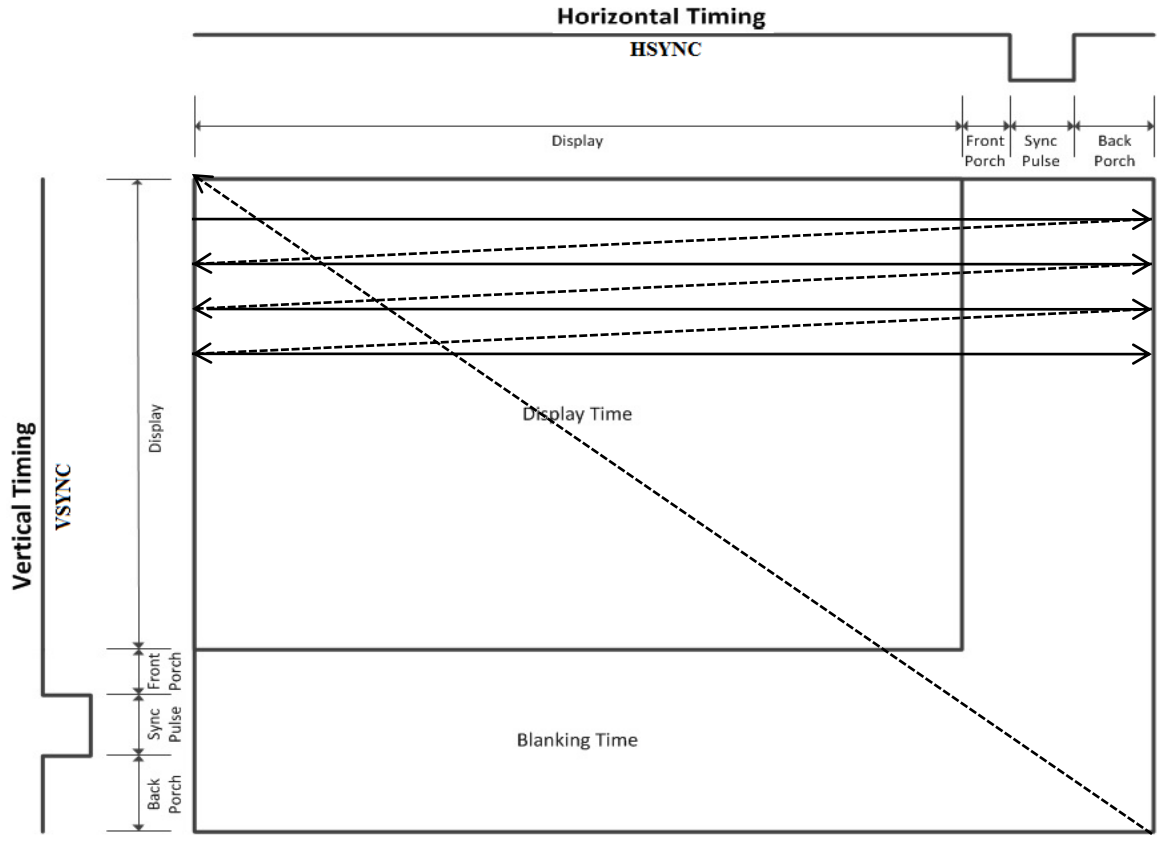
Generated Output

M	D	Output Freq (MHz)	Period Jitter (unit interval)	Period Jitter (pk-to-pk ns)
16	15	106.667	0.030	0.278

VGA Module – Controller

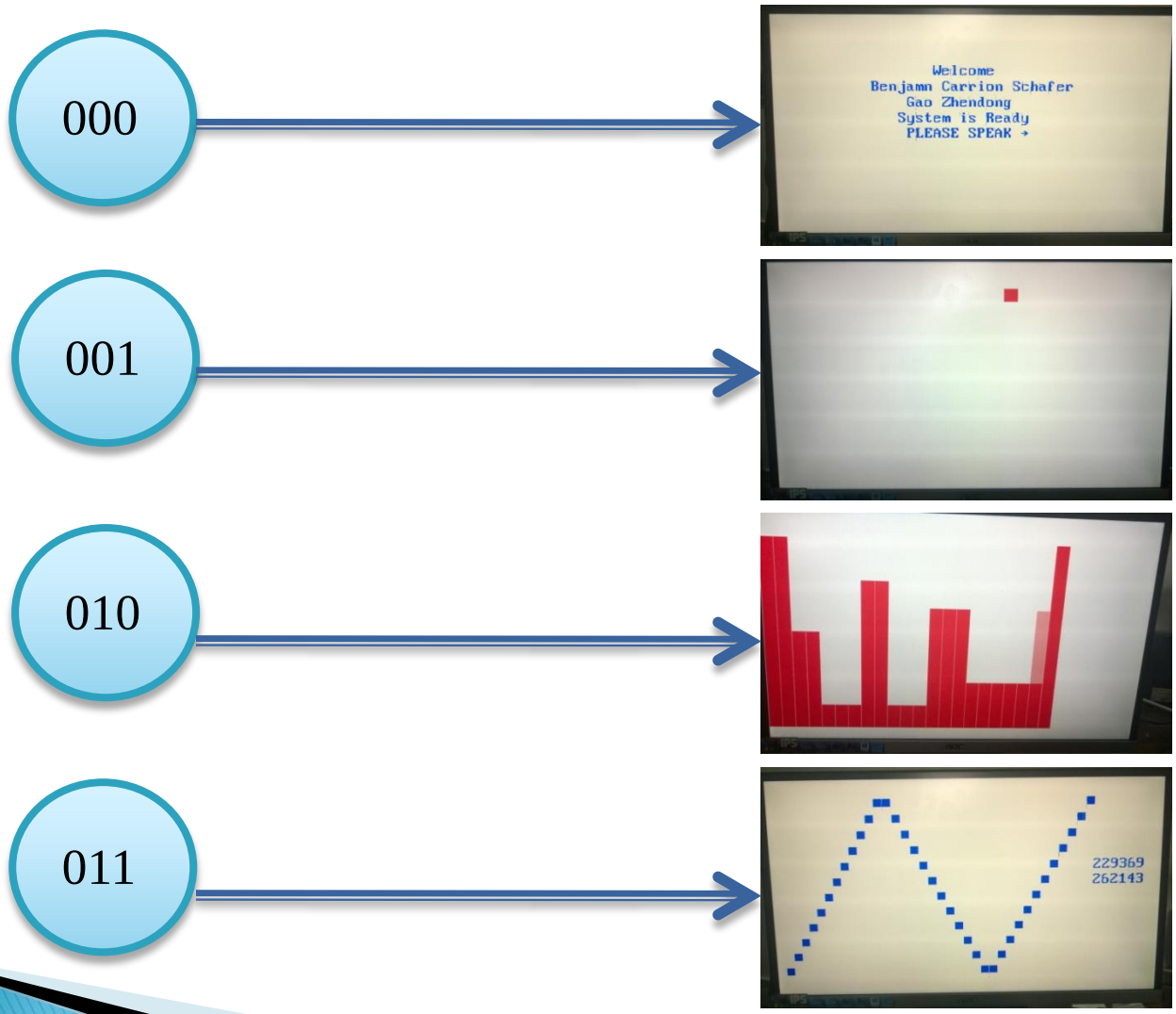
- Define the monitor resolution, fps rate, HSYNC and VSYNC signal.
- According to resolution, find the Front Porch, Sync Pulse and Back Porch.

Resolution (pixels)	Refresh Rate (Hz)	Pixel Clock (MHz)	Horizontal (pixel clocks)				Vertical (rows)			
			Display	Front Porch	Sync Pulse	Back Porch	Display	Front Porch	Sync Pulse	Back Porch
1440x900	60	106.47	1440	80	152	232	900	1	3	28
1600x1200	60	162	1600	64	192	304	1200	1	3	46



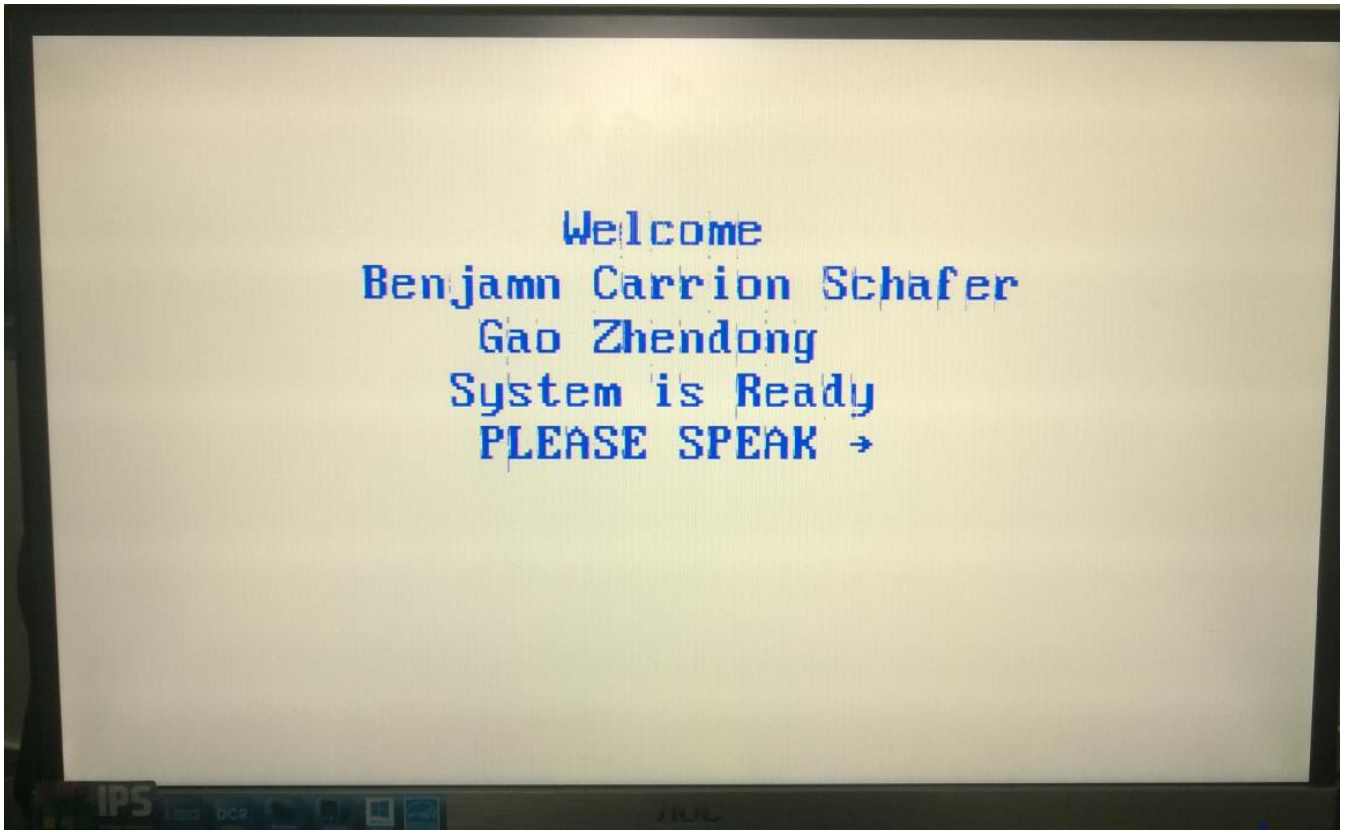
VGA Module – Imager Generator

Switches



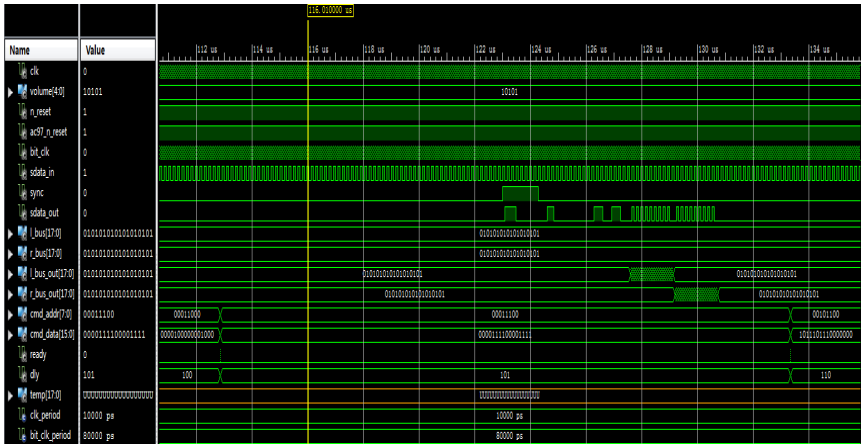
ROM

- Used to store monitor display messages.
- Send the stored display to imager generator part.



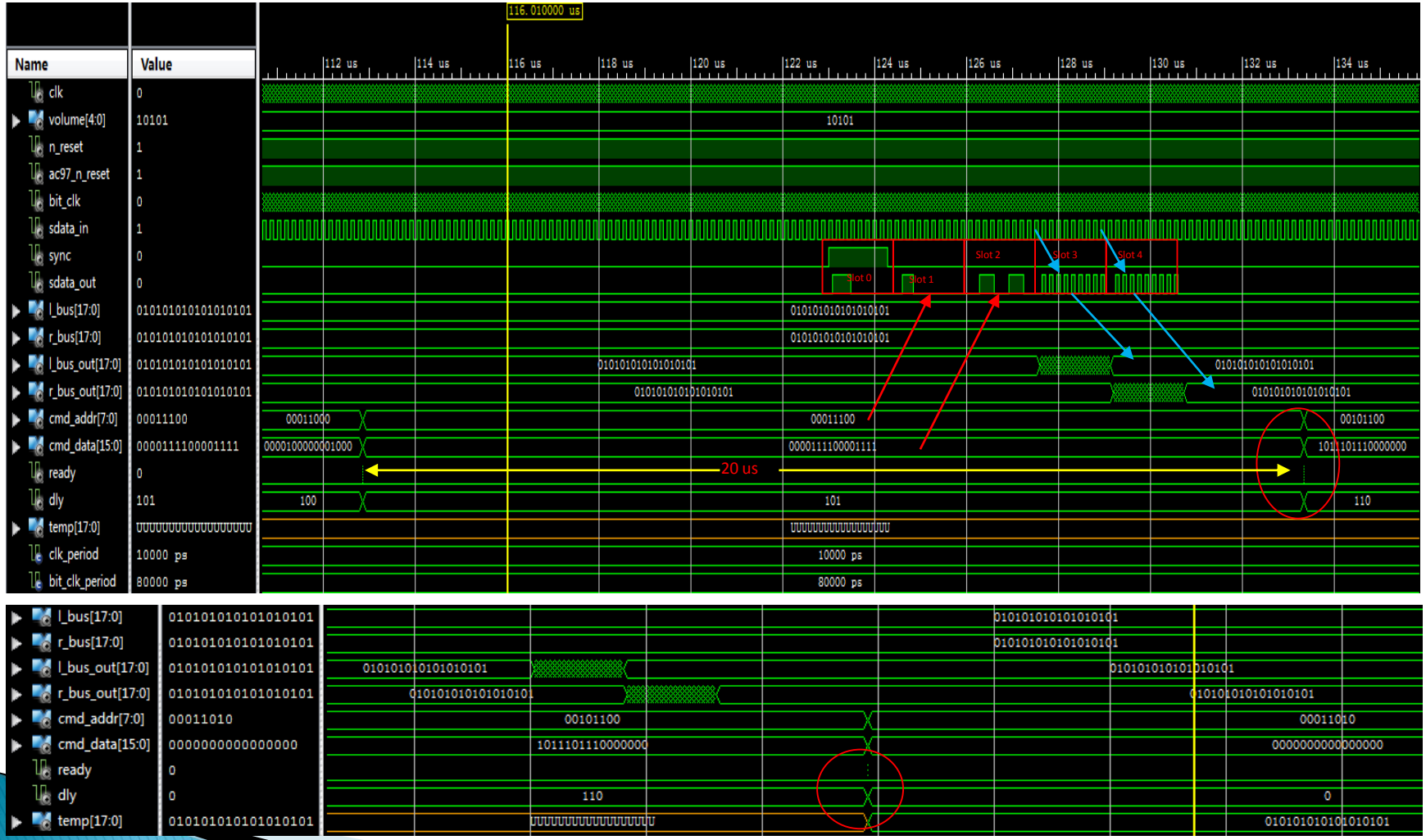
Verification and Testing

- Simulation based

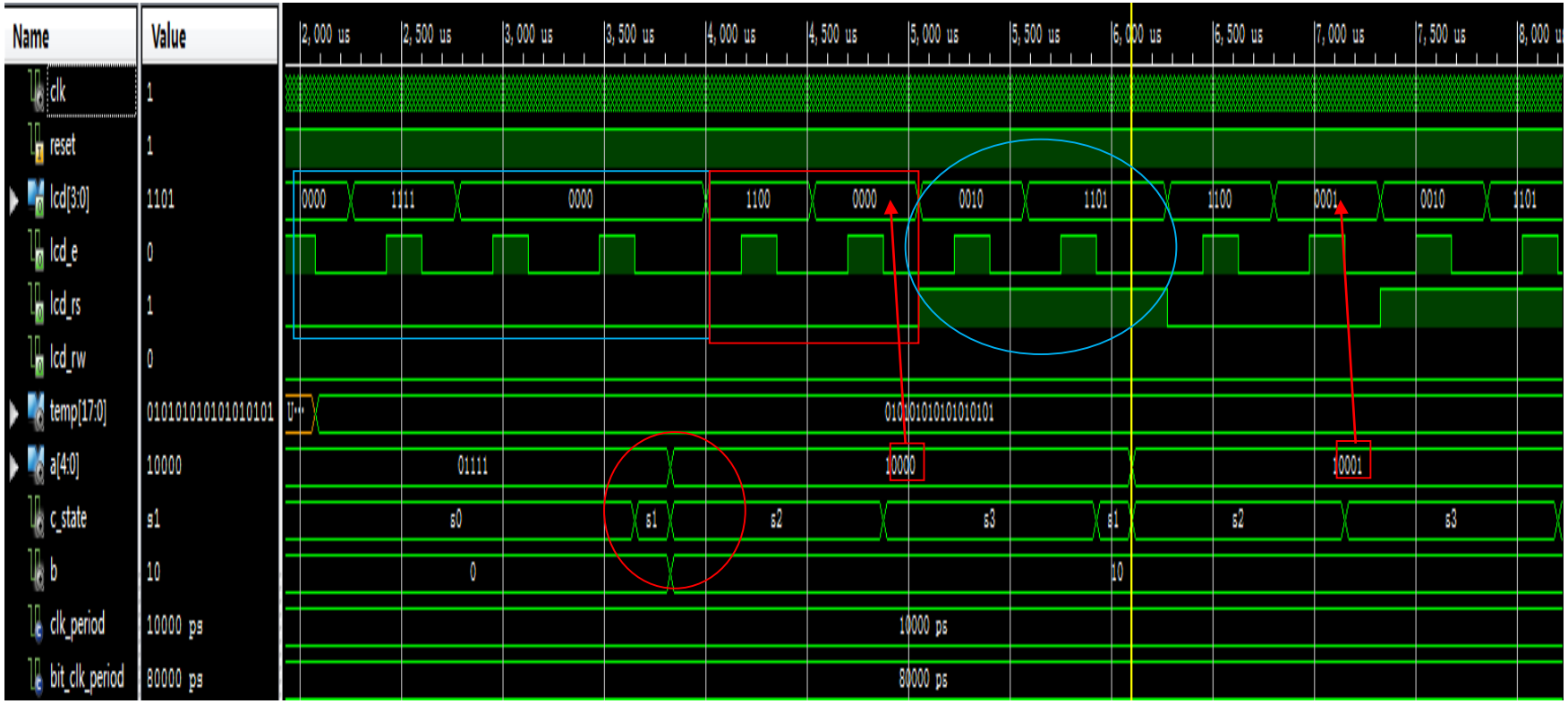


- Prototyping

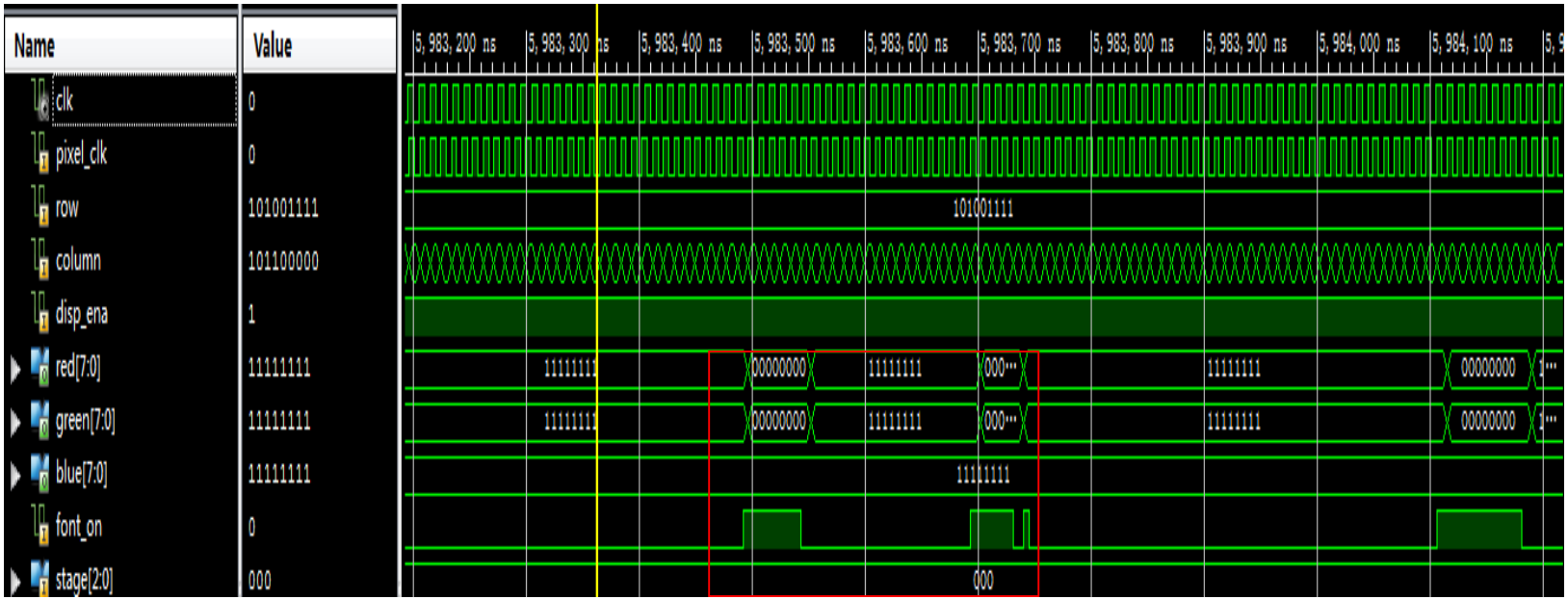
Simulation of AC97



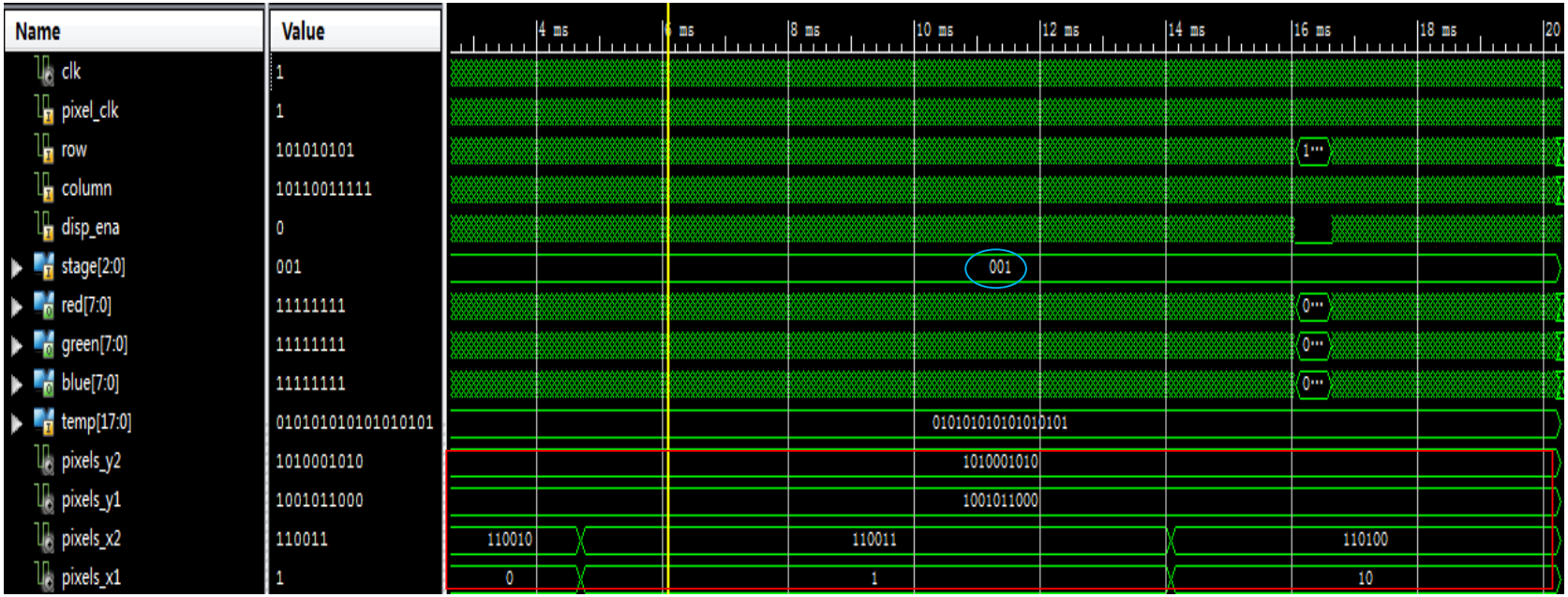
Simulation of LCD



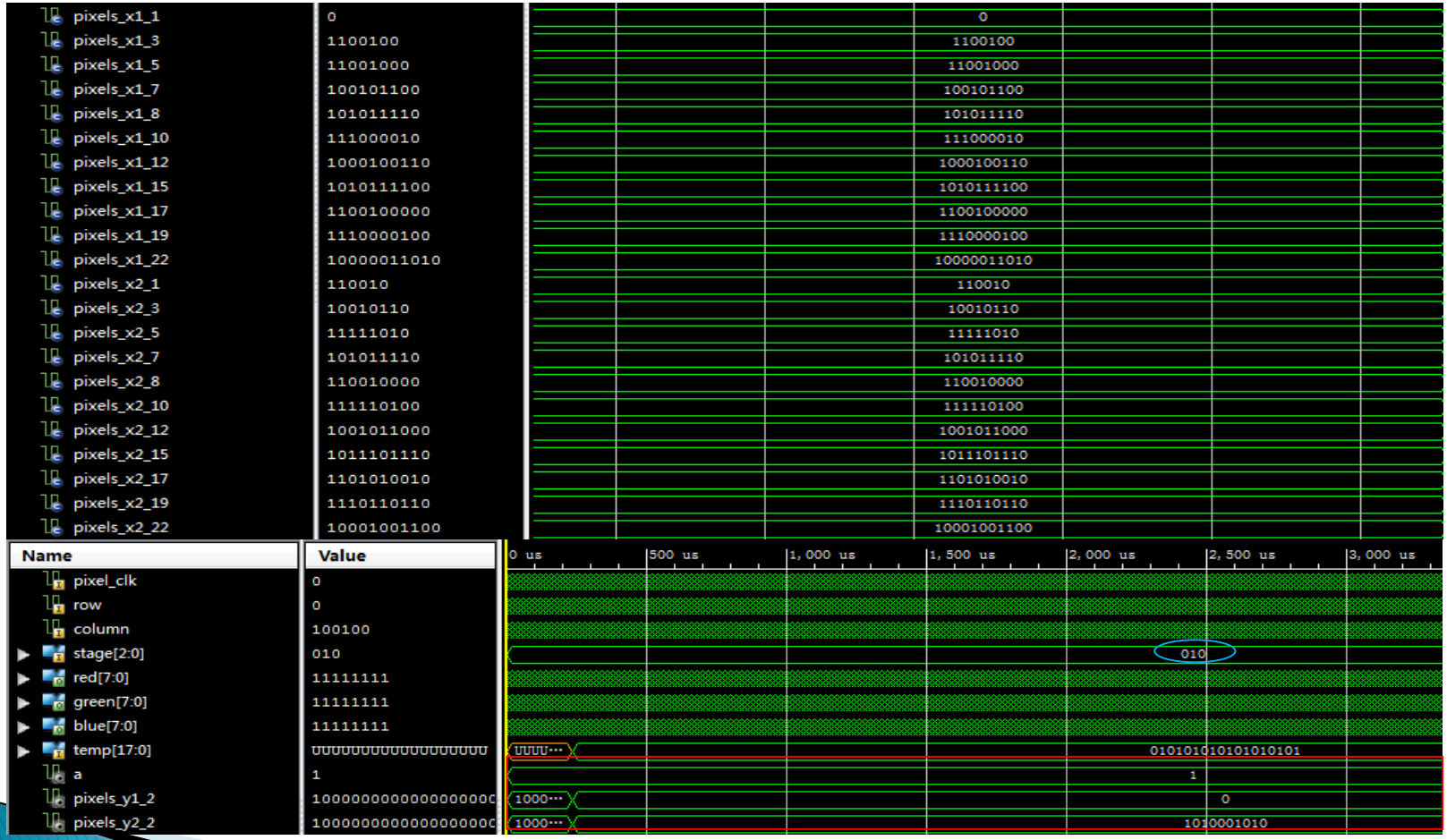
Simulation of Switch (000)



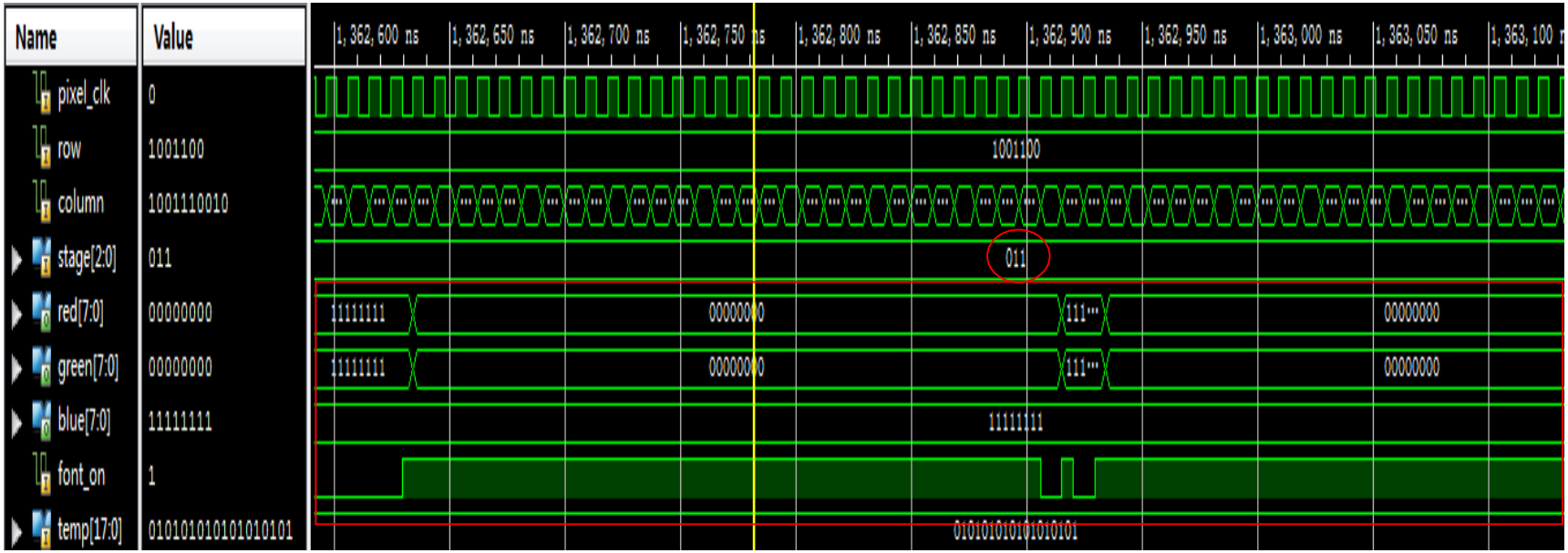
Simulation of Switch (001)



Simulation of Switch (010)



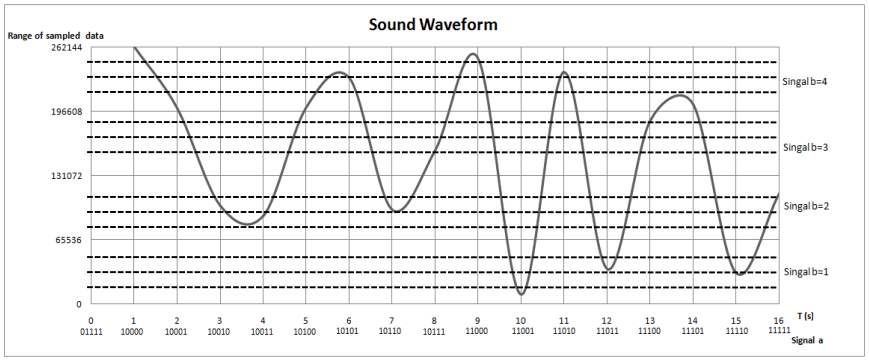
Simulation of Switch (011)



Prototyping (Video)

Summary and Conclusions

- FPGA based speech visualization
  - Top-down approach
  - High flexibility and reliability
- Visualization on:
  - LCD
  - VGA
- Learnt:
  - Mastered VHDL and Xilinx ISE
  - Interface of FPGA with different peripherals
    - Audio Codec
    - LCD
    - VGA
- Future work:
  - Divide more sampled data's range
  - Use RAM to store sampled data
  - Design an equalizer or oscilloscope



Acknowledgement

- I would like to extend my sincere gratitude to my supervisor, Dr. Benjamin Carrion Schafer, for his patience, instructive advice and useful suggestions on my thesis. I am deeply grateful of his help in the completion of this thesis.

- I am also deeply indebted to all the other students, tutors and teachers in Hong Kong Polytechnic University for their direct and indirect help to me.

- Thank you for listening to my presentation.

Thank you!

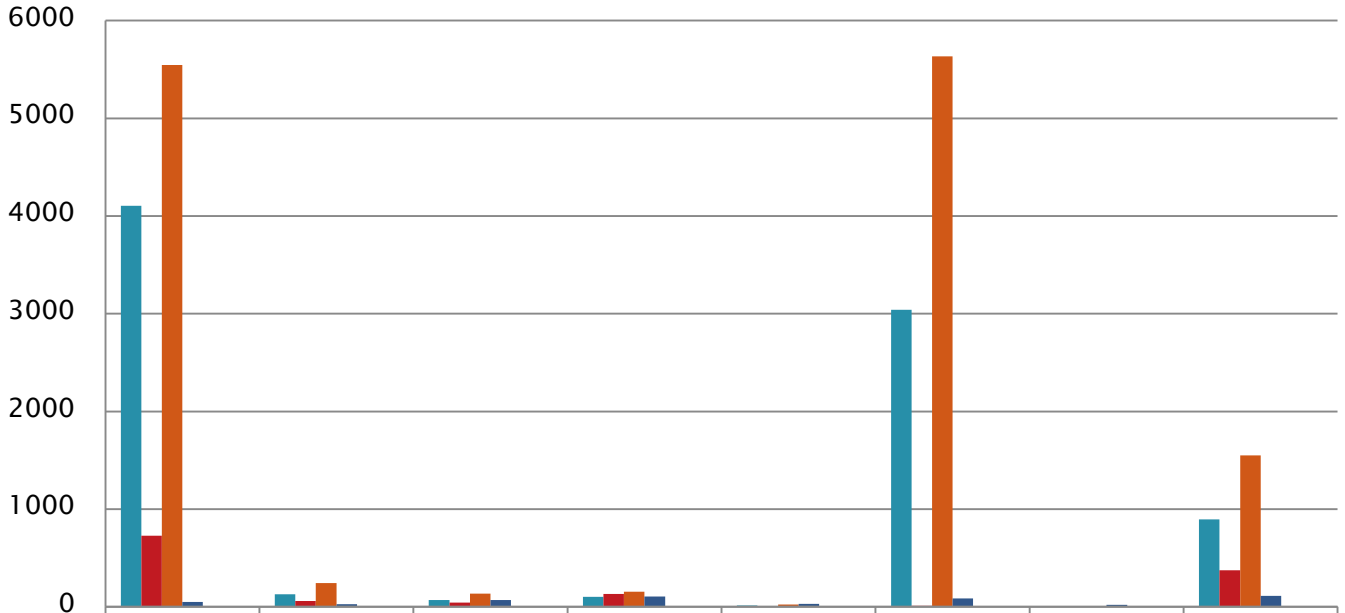
System Resources

Synthesis Report of Console System

Logic Utilization	Used	Available	Utilization
Number of Slices	4104	15360	26%
Number of Slice Flip Flops	728	30720	2%
Number of 4 input LUTs	5544	30720	18%
Number of IOs	51		
Number of bonded IOBs	51	448	11%
Number of FIFO16/RAMB16s	1	192	0%
Number used as RAMB16s	1		
Number of GCLKs	4	32	12%
Number of DCM_ADVs	1	8	12%

Return

Diagram of the Device Utilization



	Console System	LCD 1602 Module	VGA Controller Module	AC97 Controller Module	AC97 Register Module	Top Rom Module	Rom Module	Image Generator Module
Slices 15360	4104	129	71	103	13	3041	0	896
Slice Flip Flops 30720	728	60	45	133	9	11	0	375
4 input LUTs 30720	5544	242	136	154	23	5633	0	1551
bonded IOSs 448	51	27	71	105	32	87	20	112
GCLKs 32	4	1	1	2	1	1	1	2
DCM_ADVs 8	1	0	0	0	0	0	0	0

LCD Module Synthesis Report

Synthesis Report of LCD 1602

Logic Utilization	Used	Available	Utilization
Number of Slices	129	15360	0%
Number of Slice Flip Flops	60	30720	0%
Number of 4 input LUTs	242	30720	0%
Number of IOs	27		
Number of bonded IOBs	27	448	6%
Number of GCLKs	1	32	3%

ROM

Synthesis Report of TOP ROM

Logic Utilization	Used	Available	Utilization
Number of Slices	3041	15360	19%
Number of Slice Flip Flops	11	30720	0%
Number of 4 input LUTs	5633	30720	18%
Number of IOs	87		
Number of bonded IOBs	87	448	19%
IOB Flip Flops	1		
Number of FIFO16/RAMB16s	1	192	0%
Number used as RAMB16s	1		
Number of GCLKs	1	32	3%

Return

VGA Module Synthesis report

Synthesis Report of VGA Controller

Logic Utilization	Used	Available	Utilization
Number of Slices	71	15360	0%
Number of Slice Flip Flops	45	30720	0%
Number of 4 input LUTs	136	30720	0%
Number of IOs	71		
Number of bonded IOBs	71	448	15%
Number of GCLKs	1	32	3%

Synthesis Report of Image Generator

Logic Utilization	Used	Available	Utilization
Number of Slices	896	15360	5%
Number of Slice Flip Flops	375	30720	1%
Number of 4 input LUTs	1551	30720	2%
Number of IOs	112		
Number of bonded IOBs	112	448	25%
Number of GCLKs	1	32	6%

Return

AC'97 Audio Codec Module Synthesis Report

Synthesis Report of AC97 Controller

Logic Utilization	Used	Available	Utilization
Number of Slices	103	15360	0%
Number of Slice Flip Flops	133	30720	0%
Number of 4 input LUTs	154	30720	0%
Number of IOs	105		
Number of bonded IOBs	105	448	23%
Number of GCLKs	2	32	6%

Synthesis Report of AC97 Register

Logic Utilization	Used	Available	Utilization
Number of Slices	13	15360	0%
Number of Slice Flip Flops	9	30720	0%
Number of 4 input LUTs	21	30720	0%
Number of IOs	32		
Number of bonded IOBs	32	448	7%
Number of GCLKs	1	32	3%