

Middle East Technical University
Computer Engineering Department

CENG 491 – INITIAL DESIGN REPORT

DIGITAL POSTER with INTERACTIVE BLUETOOTH

By
redCat



1321785 Mehmet Kemikli İsmail
1408723 Ruslan Shavaliyev
1462704 Ebru Kulođlu
1395227 Dicle Berfin Kőse
1300862 Gőkhan Seyrankaya

TABLE OF CONTENTS

- TABLE OF CONTENTS..... i**
- 1. INTRODUCTION..... 1**
 - 1.1. Project Definition 1
- 2. HARDWARE DESIGN 2**
 - 2.1. General Hardware Diagram of the Project..... 2
 - 2.2. Required Hardwares 3
 - 2.2.1 XSA-3S1000 Board..... 3
 - 2.2.1.1 Included Components..... 5
 - 2.2.2 BR-EC30A Audio and Data Evaluation Board..... 6
 - 2.2.2.1 Included Devices 7
 - 2.2.2.2 Features 7
 - 2.2.3 Serial Cable Adapter (DB9 to DB25) 8
 - 2.2.4 Serial to Parallel Converter 8
 - 2.2.4.1 Features 9
 - 2.2.4.2 Specifications 9
 - 2.2.5 Bluetooth USB Adapter 9
 - 2.3 FPGA Circuits..... 10
 - 2.3.1 Designed Circuits 10
 - 2.4 VGA and DAC..... 10
 - 2.4.1 VGA Generation Operations..... 12
 - 2.4.1.1 I/O Ports 12
 - 2.4.1.2 PSEUDO CODE of VGA DISPLAY..... 15
 - 2.5 SDRAM PIN CONNECTIONS 16
 - 2.6 FLASH RAM 17
 - 2.7 PARALLEL PORT 19
- 3. SOFTWARE DESIGN..... 20**
 - 3.1. Login Panel 20
 - 3.2. Main Panel..... 21
 - 3.3. Main Panel with Image 22
 - 3.4. Features to be added..... 22
- 4. COMMUNICATION PART 23**
 - 4.1. Security..... 23
 - 4.2. File, Authentication Data and Configuration Data Transfer 23
 - 4.3. Broadcast..... 26
- 5. USE-CASE DIAGRAMS 28**
- 6. DATA FLOW DIAGRAMS 30**
 - 6.1. DFD0..... 30
 - 6.2. DFD1 31
 - 6.3. DFD2..... 32
 - 6.4. Data Dictionary 33
- 7. PROJECT SCHEDULING 35**
 - 7.1. Gannt Chart 35
- 8. CONCLUSION..... 36**
- 9. REFERENCES..... 37**

1. INTRODUCTION

1.1. Project Definition

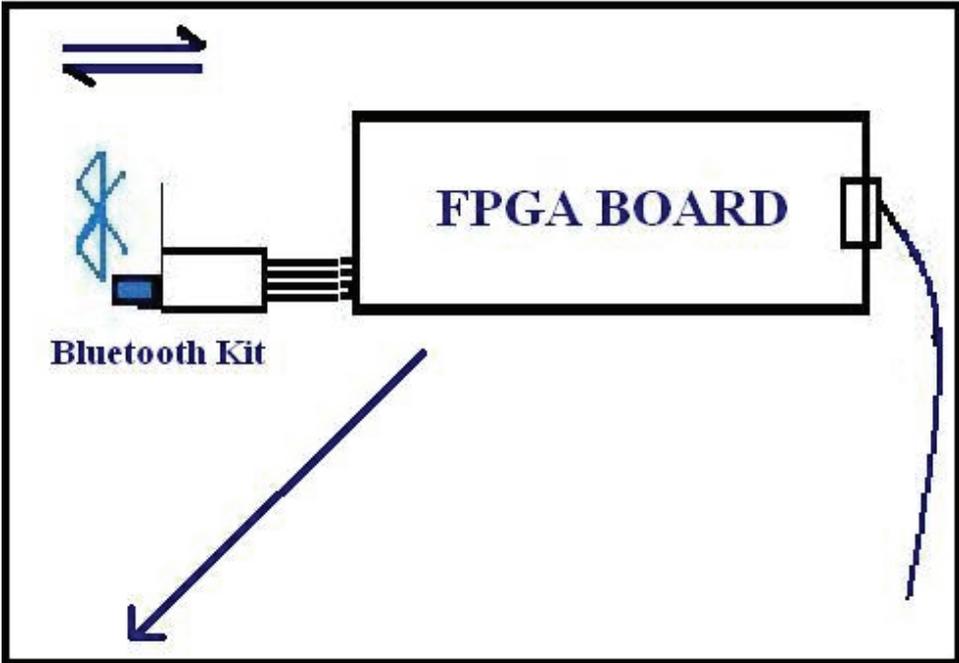
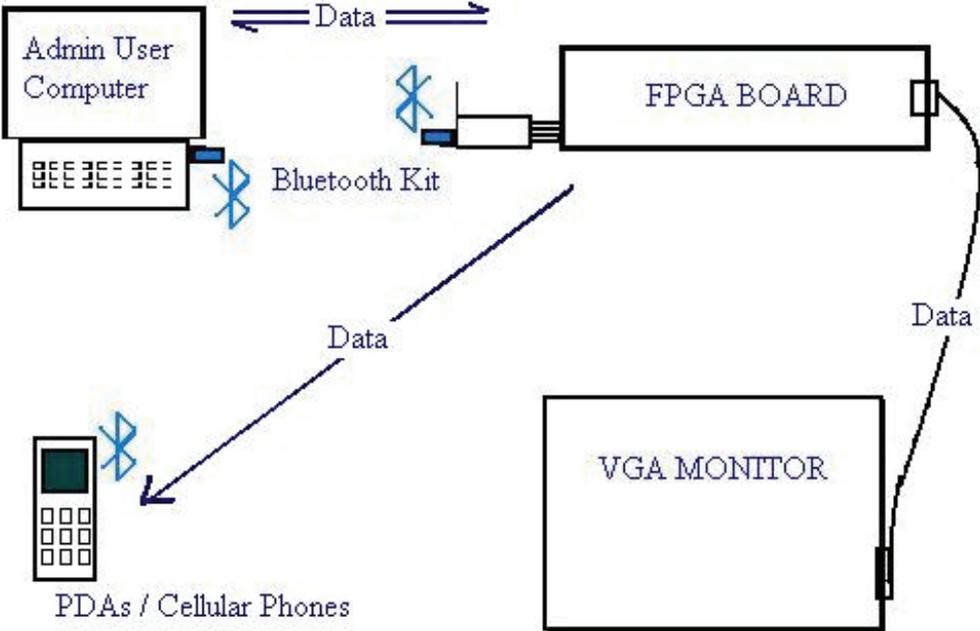
In this project, we will implement a digital poster displaying a color poster image to an VGA monitor and send the necessary poster information to cell phones or Personal Digital Assistants (PDAs) supporting bluetooth functionality. In this Project, we will be designing and programming necessary hardware required making the color poster image visible on the VGA monitors. Besides, users will be able to receive necessary poster event data via cell phones with this product. The poster we are going to implement can be used in subway stations, hotels, lifts, boardrooms, classrooms, shops, supermarkets and public areas or aboard various forms of transportation, like taxi, train. It will be used for advertisements; campaign, seminar, concert, competition, etc posters.

After giving a brief explanation about the project, it is continued with the aim of this document. This document is written in purpose of describing and reporting DIGIPOST project design process. The report mainly concentrate on our initial hardware design for FPGA board namely Xilinx XSA-3S1000 board, software design for user interaction and communication process between a device which has Bluetooth connectivity and the FPGA board,. It includes diagrams, their explanations and usage, which are determined during our design works.

2. HARDWARE DESIGN

2.1. General Hardware Diagram of the Project

The general Hardware diagram of the project is shown below:



As it can be seen from the diagram there are 2 main hardware parts in DIGIPOST project. Arroved lines are indicates the data flow.

First part is FPGA board. Its main functions can be ordered as below;

- ❖ Making and management of all communications part.
- ❖ Administration of security through connections.
- ❖ Sending/receiving and management of all kind of data such as image data and poster data, configuration data;
- ❖ Displaying poster image on VGA monitor via producing VGA signals.
- ❖ Serial to parallel or parallel to serial conversion of data.

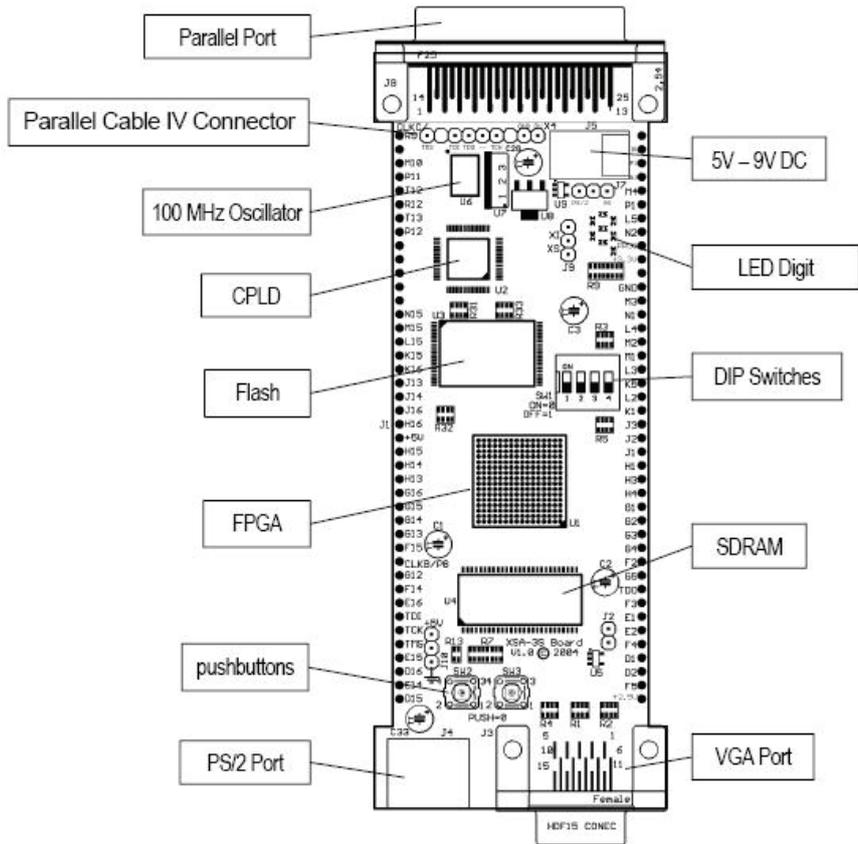
Further information is present in continuing parts of the report.

Second part is BR-EC30A that is used to communicate with the administrator computer and PDAs or Celleular Phones.through bluetooth. To explain generally, if this device gets a connection signal from user computer via bluetooth,once accepting secure connection it simply transfer the data to FPGA board through RS232 to parallel port. Then again, if it takes a signal from main circuit with the help of RS232 pin, data is send to admin computer or PDAs or cellular phones via bluetooth.

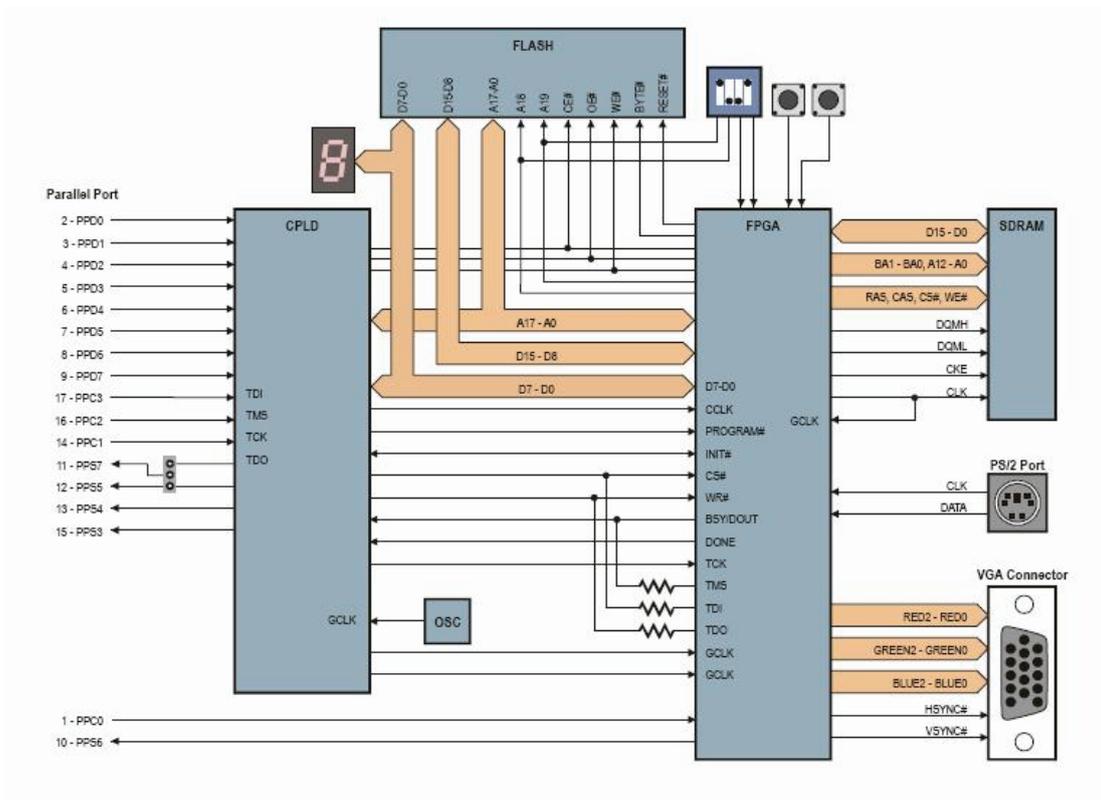
2.2. Required Hardwares

2.2.1 XSA-3S1000 Board

The XSA-3S1000 Board has the logic density to 1,000,000 gates with a Spartan-3 XC3S1000 FPGA.The FPGA is combined with a 32 MByte synchronous DRAM and 2 MByte Flash.Up to four unique bitstreams can be stored in the Flash and you can set the switches to select which bitstream configures the FPGA when power is applied.In addition to the larger FPGA, SDRAM and Flash chips, you also get a VGA port that produces vivid graphics in 512 colors. And the prototyping header gives you 65 general-purpose I/O pins that are completely free for building interfaces to external devices.



Arrangement of components on the XSA-3S1000 Board.



XSA-3S1000 Board programmer's model Block Diagram.

2.2.1.1 Included Components

- **XC3S1000 FPGA**
- **XC9572XL CPLD**
- **32 MByte SDRAM**
- **2 MByte Flash**
- **100 MHz oscillator**
- **Parallel port**
- **Keyboard/mouse PS/2 port**
- **512-color VGA port**
- **7-segment LED**
- **2 pushbuttons**
- **4 DIP switches**
- **84-pin prototyping interface (65 free I/O pins)**

FPGA: This is the main repository of programmable logic on the Board.

CPLD: This manages the interface between the PC parallel port and the rest of the Board. It can also configure the FPGA with a bitstream from Flash.

Oscillator: A fixed-frequency oscillator generates the master clock for the Board.

SDRAM: A 256 Mbit SDRAM provides volatile data storage accessible by the FPGA.

Flash: A 16 Mbit Flash device provides non-volatile storage for data and FPGA configuration bitstreams.

LED: A seven-segment LED allows visible feedback as the XSA-3S1000 Board operates.

DIP switch: A four-position DIP switch passes settings to the Board and controls the upper address bits of the Flash device.

Pushbuttons: Two pushbuttons send momentary contact information to the FPGA.

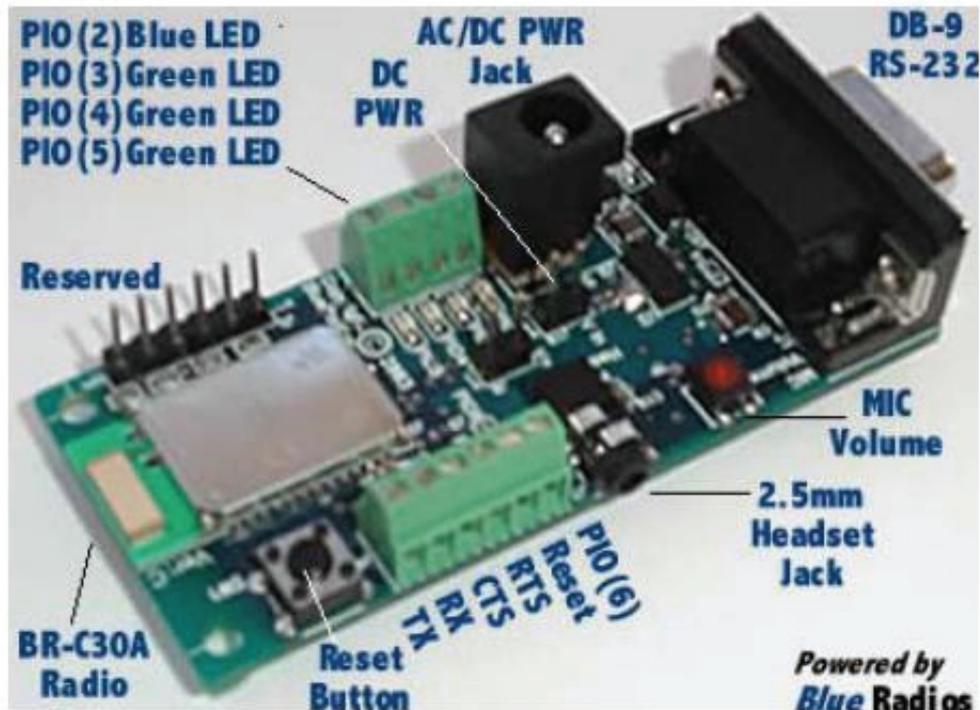
PS/2 Port: A keyboard or mouse can interface to the Board through this port.

VGA Port: The Board can send signals to display 512-color graphics on a VGA monitor through this port.

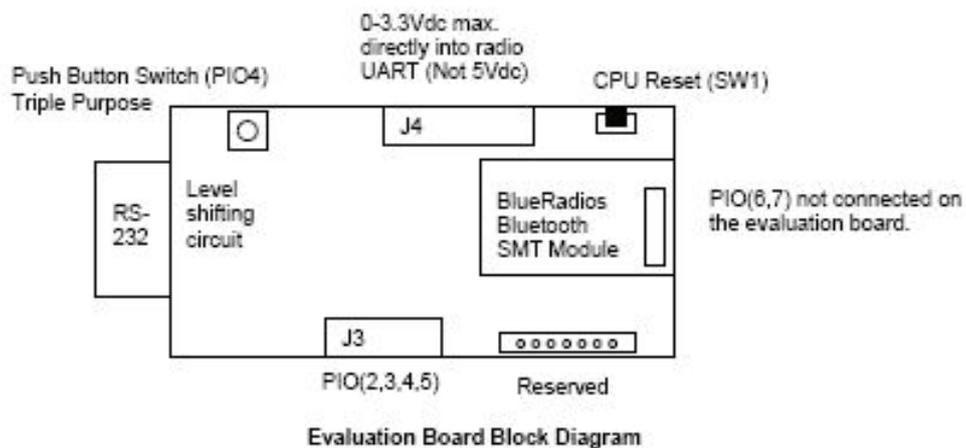
Parallel Port: This is the main interface for passing configuration bitstreams and data to and from the Board.

2.2.2 BR-EC30A Audio and Data Evaluation Board

This is a serial radio modem which is configured, commanded, and controlled through simple ASCII strings over the *Bluetooth* RF link or directly through the hardware serial UART.



RS-232 Audio and Data Evaluation PCB (BR-EC30A)



2.2.2.1 Included Devices

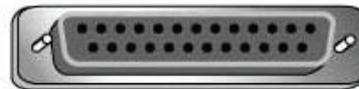
- ❖ Wireless data and voice communications module certified to Bluetooth® v1.2
- ❖ Audio CODEC, head jack, head phones, MIC volume control.
- ❖ FCC, CE, Industry Canada, and Bluetooth® certified ISM 2.4GHz band module.
- ❖ RS-232 (DB-9), and 0-3.3Vdc logic levels
- ❖ Includes integrated software stack, profiles, and AT modem like commands.
- ❖ Embedded Bluetooth Stack Profiles Included(requires no host MCU stack):
- ❖ SPP, DUN, LAN, PAN, Headset, Audio Gateway, FTP, GAP, SDP, RFCOMM, and L2CAP protocols.
- ❖ Evaluation Board Accommodates both BR-C30A and BR-C29A radio modules

2.2.2.2 Features

- ❖ Dedicated PCM voice audio channel
- ❖ UART baud rate data speeds: 1200bps up to 921.6Kbps, and customized
- ❖ +100 meter (330 feet) distance
- ❖ Software adjustable transmitter power from short to long range applications
- ❖ Includes AC/DC power supply
- ❖ 13 bit linear mono CODEC
- ❖ Programmable Input Output (PIO's)
- ❖ Reset push button
- ❖ LED status: Power, *Bluetooth* Connection, Slave status, etc.
- ❖ 2.5mm audio jack
- ❖ Low power consumption radio only
- ❖ RS-232 and 3.3Vdc TTL inputs
- ❖ Self-discovery and network equipped multi-points
- ❖ Operating temperature range: -40~+70°C.
- ❖ Secure and robust communication link
 - FHSS (Frequency Hopping Spread Spectrum)
 - Encryption, and 16 alphanumeric Personal Identification Number (PIN)
 - Error correction schemes for guaranteed packet delivery

2.2.3 Serial Cable Adapter (DB9 to DB25)

This Serial Cable Adapter transforms a DB25 male serial port into a DB9 male port



2.2.4 Serial to Parallel Converter

This device will provide us with a total switch from serial data to parallel data. Either the adapter above with a converter circuit [2.3.1] will be used on CPLD or this cable will be obtained. The diagrams base on the usage of a converter cable.

Serial↔Parallel Converter V

Easy conversion—no other devices needed.



Product Highlights

- Makes the DB9 serial port on your PC compatible with parallel devices. Connect your PC to a printer—even with something plugged into its parallel port.
 - Converts serial to parallel or parallel to serial.
 - Automatic serial/parallel mode detection.
- [more...](#)

2.2.4.1 Features

- Built-in 6-foot (1.8-m) cable eliminates the need for extension cables. It includes a DB9 female socket for serial ports and either a Centronics® makes or a DB25 male plug on the other end.
- Interface powered—no AC outlets needed.
- Runs at speeds up to 38.4 kbps.

2.2.4.2 Specifications

Controls — External DIP switches: Speed, flow control, parity, word length

Data Format — Even, odd, or no parity; 7 or 8 data bits, 1 stop bit

Flow Control — Hardware or X-ON/X-OFF

Speed — 300 bps to 38.4 kbps, switch selectable

CE Approval — Yes

Interface — Serial: Asynchronous;

Parallel: Centronics®

Connectors — Both: (1) DB9 F, DCE;

PI045A: (1) 36-pin Centronics M;

PI046A: (1) DB25 M

Indicators — (1) LED

Power — From the RS-232 interface

Size — 2.7“H x 1.3“W x 0.8“D (6.9 x 3.3 x 2 cm)

Weight — 0.2 lb. (0.1 kg)

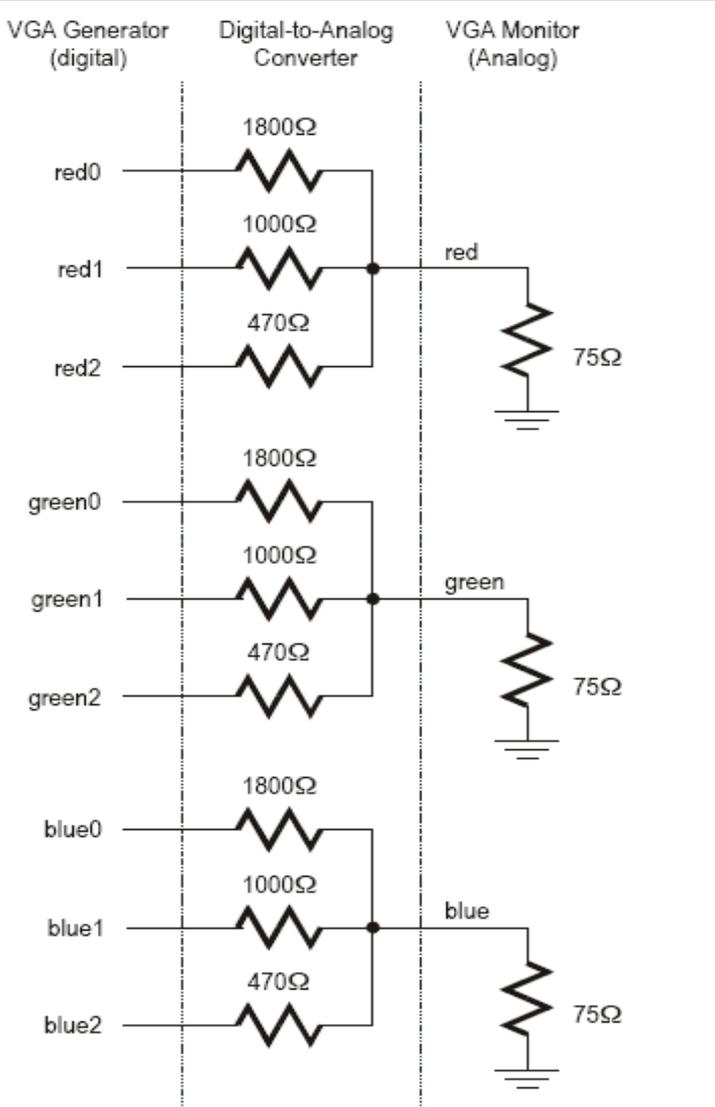
2.2.5 Bluetooth USB Adapter

Another device which is provided to us is INCA bluetooth usb adaptor which is supposed to create communication band between evaluation board and computer. This Serial Cable



The below circuit gets 3 3 bits of information from an digital input source [2 to 0] and converts the input into 3 analog signals which are the RGB values. The converter can generate 512(8 x 8 x 8) different colors and each pixel of the monitor will display one of these colors.

In our Project, we need this circuit in order to convert the digital inputs into analog signals and send the data to the VGA.



2.4.1 VGA Generation Operations

The block diagram of the VGA generator circuit is as follows. We see that there are two Sync Generator circuits namely Vertical and Horizontal Sync Generator circuits. These circuits are identical save for the parameters that determine the pulse timing. The horizontal sync generator outputs a single-cycle gate signal coincident with the leading edge of the horizontal sync pulse. This gate signal connects to the clock-enable of the vertical sync generator so it only updates its timing counter once per line of pixels. The gate signal of the vertical sync generator is used as an end-of-frame indicator to the external source of pixel data. It also resets the pixel buffer and clears its contents so the VGA generator starts from a completely cleared state on every frame. The sync generators also output the horizontal and vertical blanking signals .

2.4.1.1 I/O Ports

Now, it is time to describe the main input and output ports of the circuits.

rst: This active-high, asynchronous input resets the internal circuitry of the sync generators.

clk: This is the main clock input. The clock from the external oscillator enters the FPGA through a global clock input pin and drives this input.

pixel_data_in: 16-bit data containing one or more pixels enters the pixel buffer through this bus.

eof: This active-high output indicates when the display of a video frame has been completed and pixels for the next frame can begin entering the buffer.

full: This active-high output indicates when the pixel buffer is full and no more space is currently available for more pixels.

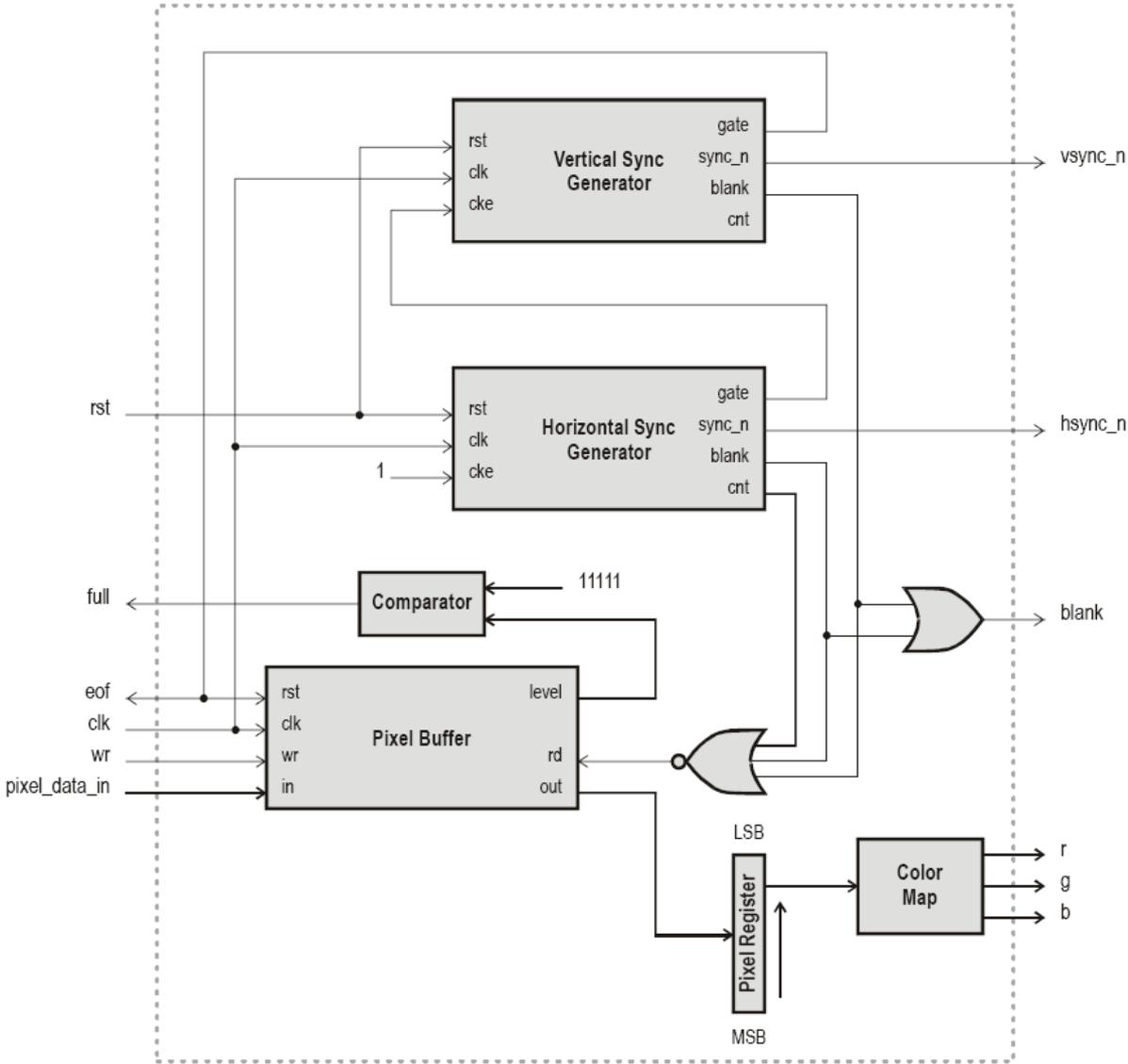
vsync_n: This active-low output drives the vertical sync input of a VGA monitor.

hsync_n: This active-low output drives the horizontal sync input of a VGA monitor.

blank: This active-high output signals when the red, green and blue video signals are blanked.

r: This bus carries the data bits for the red video component to a DAC whose analog output is delivered to the VGA monitor.

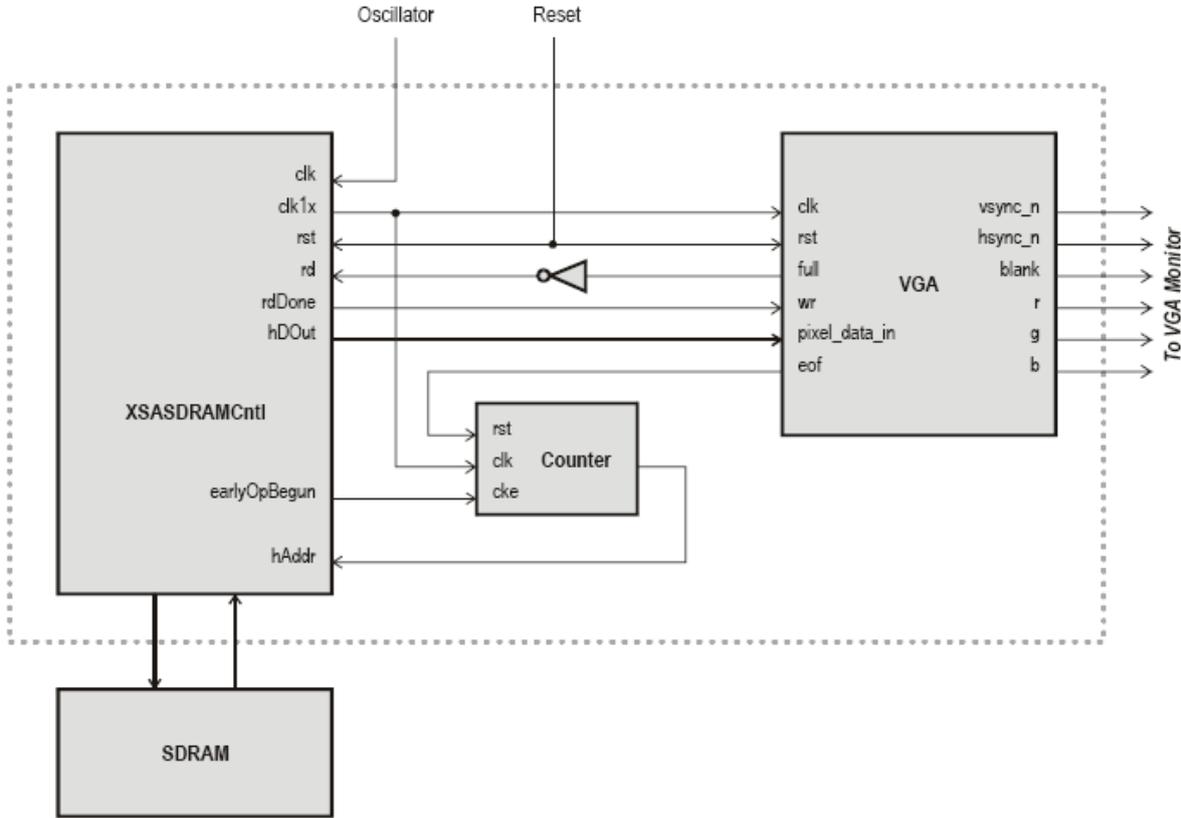
g: This bus carries the data bits for the green video component to a DAC whose analog output is delivered to the VGA monitor.



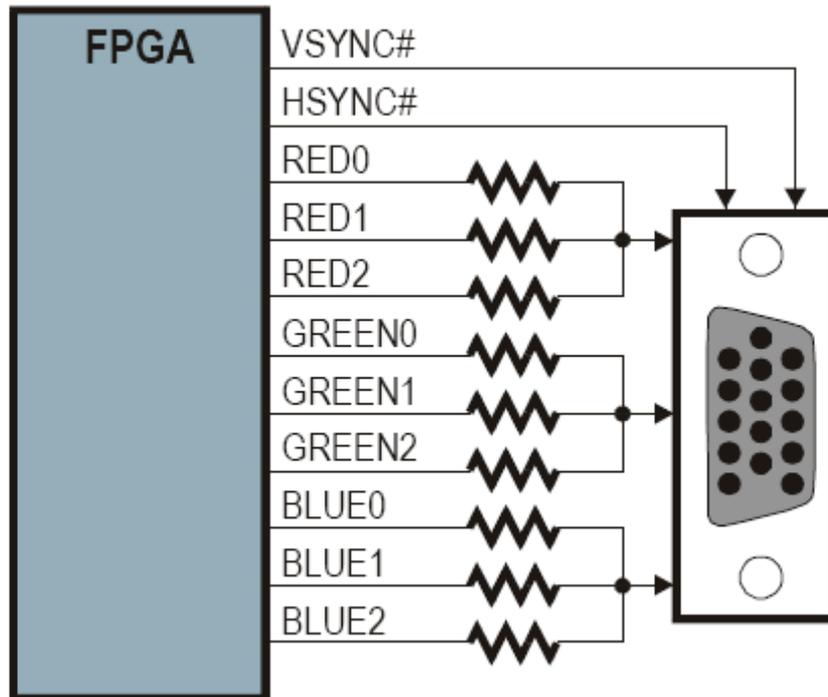
In this Project, we will get use of the functionality of this in order to read image data from the SDRAM of the XSA-3S1000 board and send the data to VGA monitor.

Below, the connections between the SDRAM and the VGA generator circuit is presented.

The circuit counter will be holding the value of the next SDRAM location to be read when the pixel buffer is empty. Once the read operation starts, the an signal will be high, namely earlyOpBegun. Once the read operation is complete and the data is available, the rdDone signal enables write operation to the pixel buffer. Once the a complete video frame is sent succesfully to the Vga monitor, a eof signal becomes high and resets the counter and the entire process starts again.



As, it is displayed below more clearly, the generated outputs from the FPGA are VSYNC, HSYNC, RED0, RED1, RED2, GREEN0, GREEN1, GREEN1, BLUE0, BLUE1, BLUE2. These are sent to a simple DAC. The outputs of the DAC are sent to the RGB inputs of VGA monitor.



2.4.1.2 PSEUDO CODE of VGA DISPLAY

```

/* send L lines of video to the monitor */
for line_cnt=1 to L
    /* send P pixels for each line */
    for pixel_cnt=1 to P
        /* get pixel data from the RAM */
        data = RAM(address) address = address + 1
        /* RAM data byte contains 4 pixels */
        for d=1 to 4
            /* mask off pixel in the lower two bits */
            pixel = data & 00000011
            /* shift next pixel into lower two bits */
            data = data>>2
            /* get the color for the two-bit pixel */
            color = COLOR_MAP(pixel)
            send color to monitor
            d = d + 1
        /* increment by four pixels */
    
```

```

pixel_cnt = pixel_cnt + 4
/* blank the monitor for H pixels */
for horiz_blank_cnt=1 to H
    color = BLANK
    send color to monitor
    /* pulse the horizontal sync at the right time */
    if horiz_blank_cnt>HB0 and horiz_blank_cnt<HB1
        hsync = 0
    else hsync = 1
    horiz_blank_cnt = horiz_blank_cnt + 1
line_cnt = line_cnt + 1
/* blank the monitor for V lines and insert vertical sync */
for vert_blank_cnt=1 to V
    color = BLANK send color to monitor
    /* pulse the vertical sync at the right time */
    if vert_blank_cnt>VB0 and vert_blank_cnt<VB1
        vsync = 0
    else vsync = 1
    vert_blank_cnt = vert_blank_cnt + 1

/* go back to start of picture in RAM */
address = 0

```

2.5 SDRAM PIN CONNECTIONS

The XSA-3S1000 board 's SDRAM is connected directly to FPGA as shown below. The SDRAM does not share any FPGA pins with any other device. Therefore, any application can use SDRAM regardless of the other components that are to be used. Therefore, to write to SDRAM or read from SDRAM is very easy to implement. It is done briefly as follows.

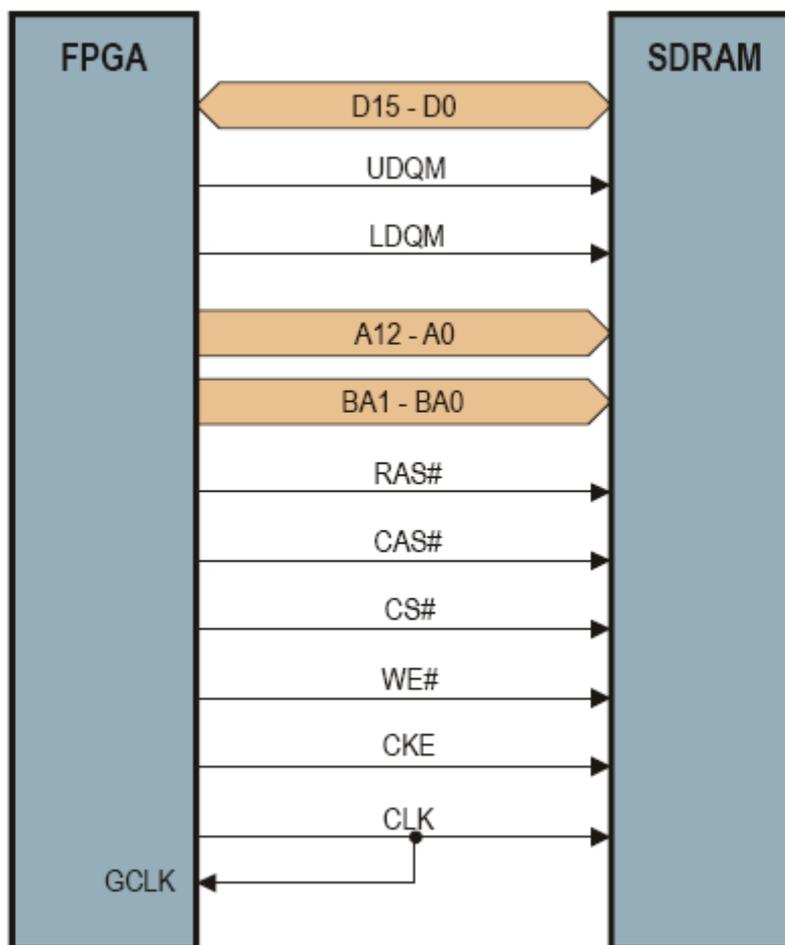
READ:

- Set WE# pin to 0
- Set the address to be read with the pins A12-A0
- Read the data from pins D15-D0

WRITE

- Set WE# pin to 1
- Set the address to be read with the pins A12-A0
- Write the data to pins D15-D0

Note that XSA-3S1000 allows synchronization with the FPGA's internal operations with the SDRAM operations via CLK pin.

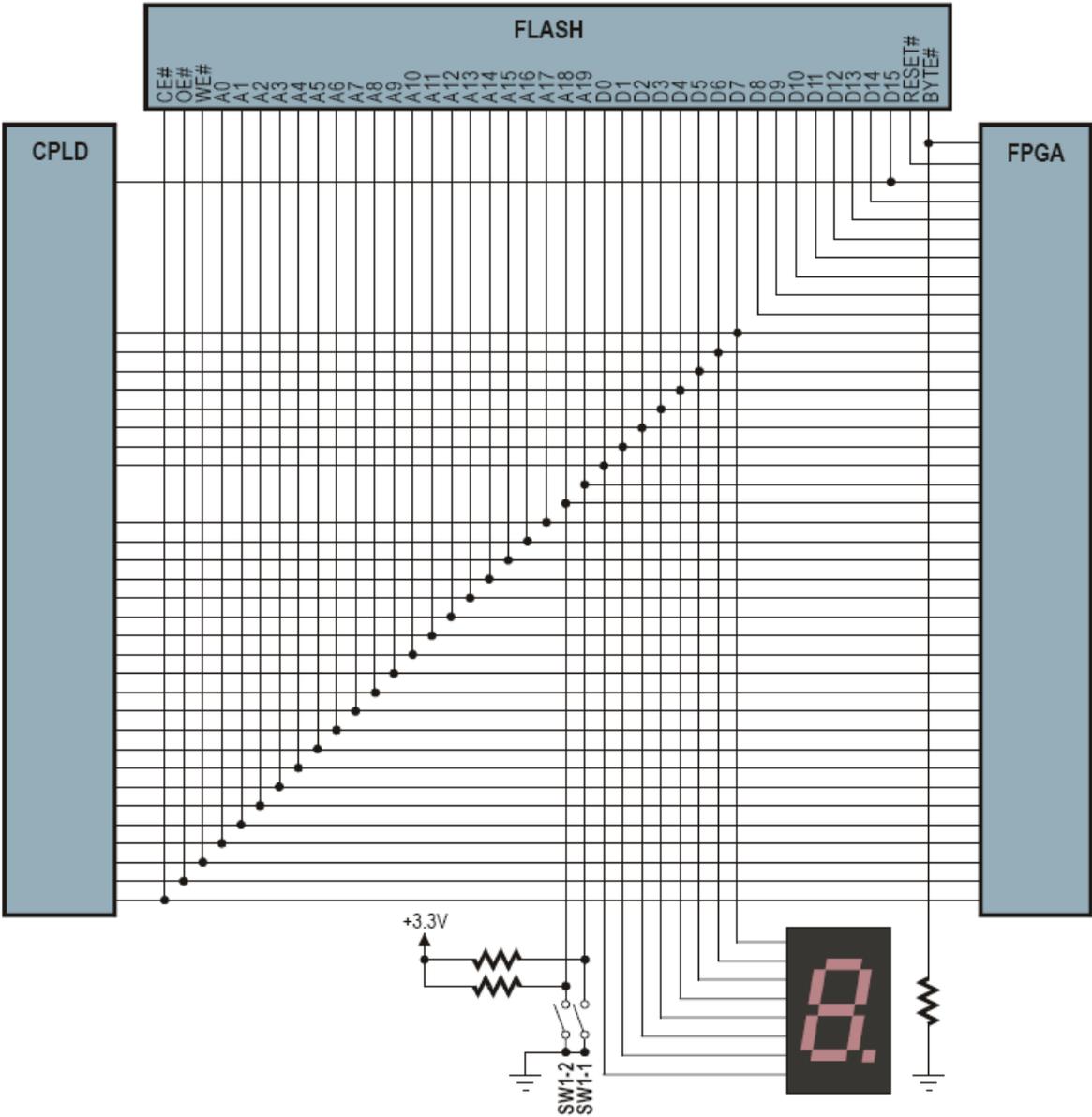


2.6 FLASH RAM

In this Project, we need to connect to FLASH RAM with the help of FPGA and need to write or read non-volatile data such as configuration data of the circuit. As it can be seen from the

pin connections of SDRAM and FPGA, the FPGA connects to the entire 16-bit data bus and can select either byte mode (2M x 8) or word mode (1M x 16) using the BYTE# control line.

After power-up, any application circuit loaded into the FPGA can read and/or write the Flash. Enabling WE# pin triggers write operation, whereas, disabling it causes a read operation. As it is obvious, the data bus is for carrying data and address bus is for supplying write or read address of FLASH RAM. To avoid contention, the CPLD is programmed to release control of all Flash address/data/control lines whenever the Flash CE# line. When the Flash is disabled by raising CE#, the I/O lines connected to the Flash are available for general-purpose communication between the FPGA and the CPLD.

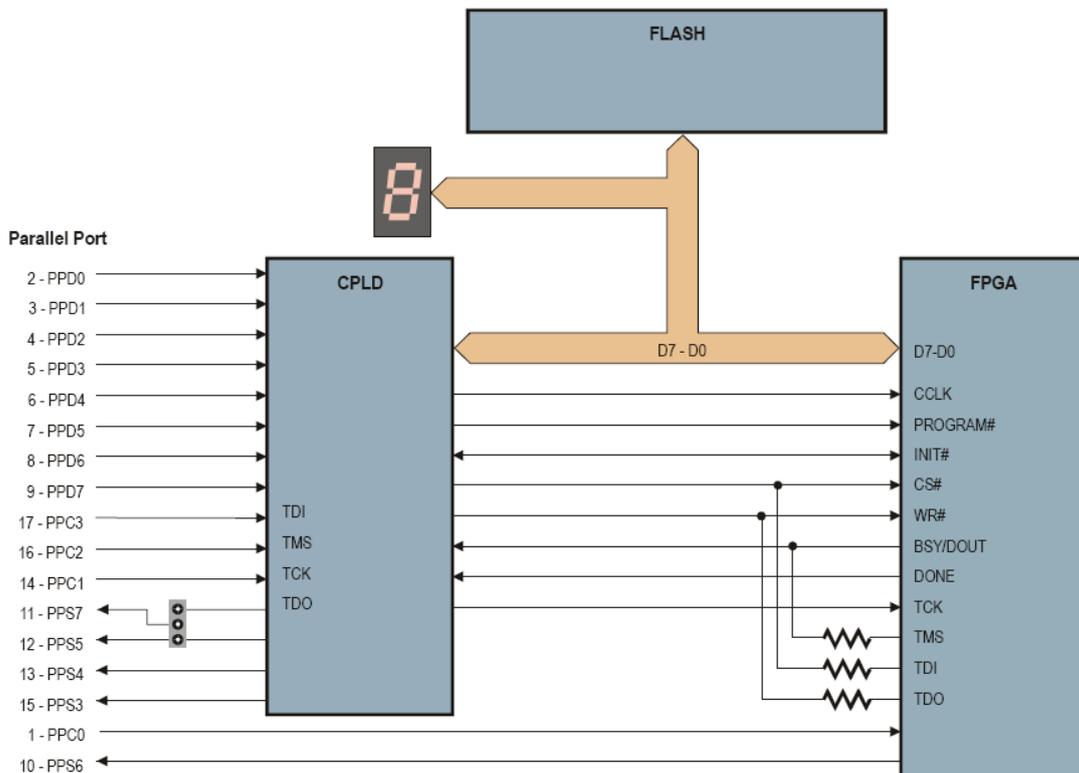


2.7 PARALLEL PORT

The parallel port is the main interface for communicating between the XSA-3S1000 Board and a PC. Control line C0 and status line S6 connect directly to the FPGA and can be used for bidirectional communication between the FPGA and PC. The CPLD handles the fifteen remaining active lines of the parallel port as follows.

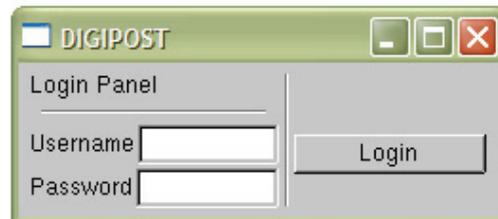
The eight data lines, D0–D7, and the three status lines, S3–S5, connect to general-purpose pins of the CPLD. The CPLD can be programmed to act as an interface between the FPGA and the parallel port. The CPLD connects to the FPGA configuration pins so it can pass bitstreams from the parallel port to the FPGA. The actual configuration data is presented to the FPGA on the same 8-bit bus that also connects to the Flash and seven-segment LED. The CPLD also drives the configuration pins (CCLK, PROGRAM#, CS#, and WR#) that sequence the loading of a bitstream into the FPGA. The CPLD can monitor the status of the bitstream download through the INIT#, DONE, and BSY/DOUT pins.

After the FPGA is configured with a bitstream and the DONE pin goes high, the CPLD switches into a mode that connects the parallel port data and status pins to the FPGA. This lets the PC pass data to the FPGA over the parallel port data lines while receiving data from the FPGA over the status lines. The active connections between the FPGA,



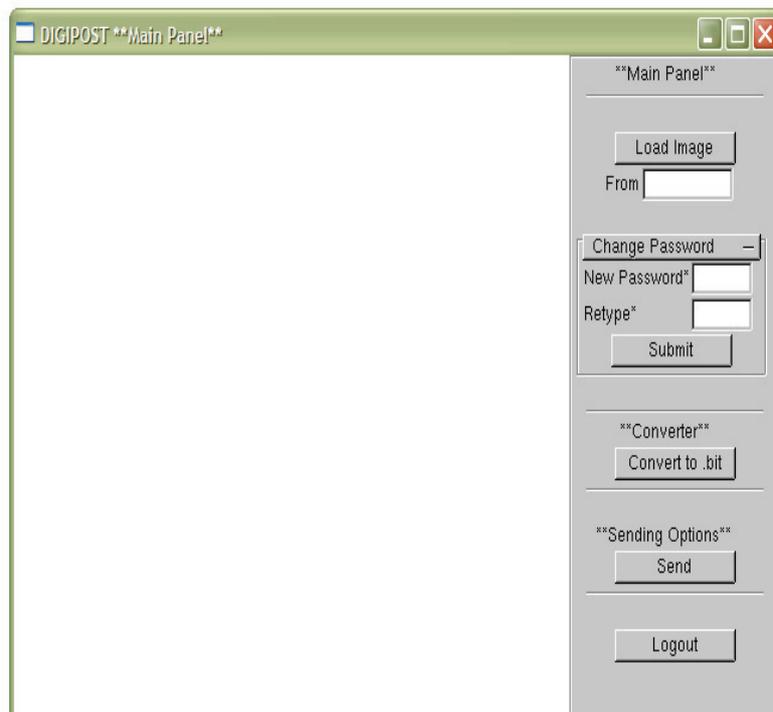
3. SOFTWARE DESIGN

3.1. Login Panel



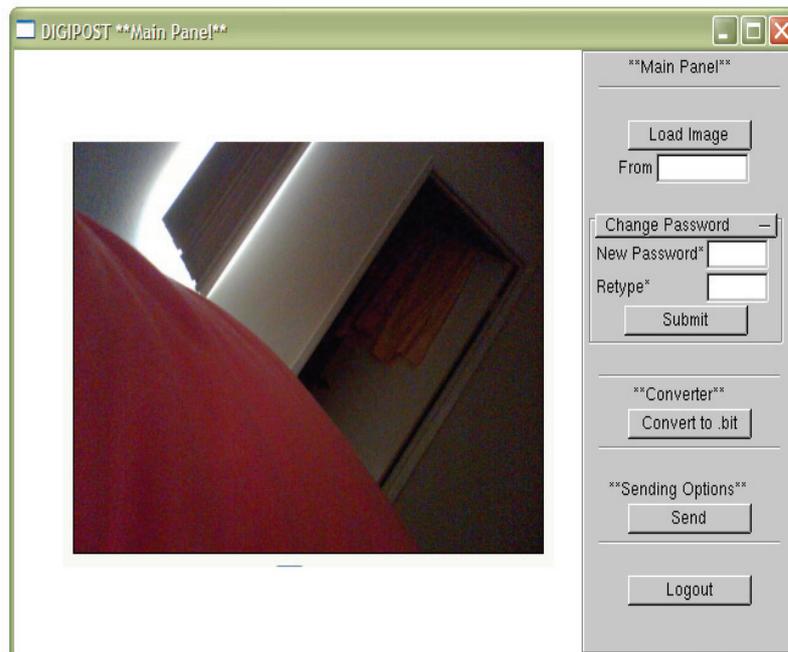
First of all, in order to use this software, user should authorize him/her by providing correct Username and Password; user should fill Username and Password fields correctly. After user fills the necessary fields, the provided data by him/her will be sent to Board, and checked for validity. If the validity signal received by the software from the Board, user will be redirected to "Main Panel". Otherwise, if the data provided by user wasn't accepted by the Board and the invalidity signal is sent to software, user will be asked to provide the valid Username and Password again.

3.2. Main Panel



After user had been accepted by the system, in other words, the provided data by the user were valid, and the validity signal was received by the software, user will be redirected to the "Main Panel", which will provide user by necessary applications and abilities. First of all, user is strongly recommended to change his/her default password for some security reasons, by clicking to "Change Password" roll-out. In order to achieve password change, user is asked to provide new password, and retype it also for security reasons . After new password was typed twice, user has to click on "Submit" button, to update the system. Secondly, user will have the ability to load image and have a look at image. The last application is necessary, in order to be sure that the loaded image is exactly that one, which has to be sent. In other words, to prevent the user from double-checking the image.

3.3. Main Panel with Image



After the image is loaded, it is displayed in the working window. Thirdly, the user has the ability to convert loaded image to .bit format by clicking "Convert to .bit" button, in order to have the user be able to send image to Board. Image is converted to default.bit, and then if user wishes to send the image to Board, he/she can easily do it, by clicking on "Send" button. Finally, after all work is done, user can logout from the system, by clicking on "Logout" button, he/she will be redirected to initial Login Panel.

3.4. Features to be added

In initial design, we just want to provide the user with all common features, and then we will decide on which features can be added to software.

4. COMMUNICATION PART

4.1. Security

Although there seem to be quite some problems in the security aspects of Bluetooth, such as the encryption and the authentication algorithms, one could justify the weaknesses by the special characteristics of the network. Firstly one should note that a device's range normally is up to 10 meters, with a maximum of 100, thus the threats can be minimized in such a constraint environment by other means. Also the most common uses are for communication between mobile phones, that normally do not require tremendous security features, whilst they do require low power consumption.

As long as the correct password is typed in for the bluetooth kit, the connection between the admin pc and the kit is provided. Our bluetooth kit also has the feature of 56-128 bit encryption and authentication that comes Standard with the Bluetooth. Thusly enough security is provided via our hardware.

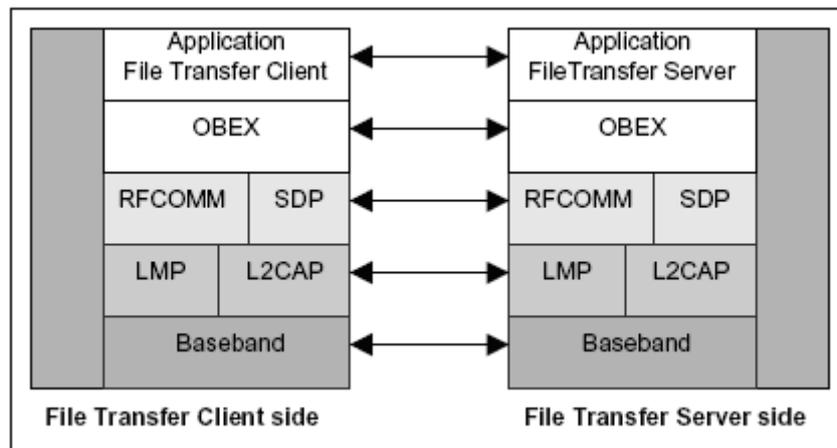
4.2. File, Authentication Data and Configuration Data Transfer

Bluetooth units communicate with other units through several profiles. There are 13 of these profiles and the ones that are used for data transfer between 2 Bluetooth units are K12-File Transfer Profile in conjunction with the K10-Generic Object Exchange Profile (GOEP). By these profiles, files, entire folders, directories and streaming media formats can be transferred. The model also offers the possibility of browsing the contents of the folders on a remote device.

GOEP defines the set of protocols and procedures to be used by applications handling object exchanges. Several usage models, are based on this profile, e.g. File Transfer and Synchronization. Typical Bluetooth units using this profile are notebook PCs, PDAs, mobile phones and smart phones.

Applications using the GOEP assume that links and channels are established, as defined by the GAP. The GOEP describes the procedure for pushing data from one Bluetooth unit to another. The profile also describes how to pull data between units. The GOEP is dependent on the Serial Port Profile.

K12-File Transfer Profile defines the requirements for the protocols and procedures that shall be used by the applications providing the File Transfer usage model. This profile uses the Generic Object Exchange profile (GOEP) as a base profile to define the interoperability requirements for the protocols needed by the applications.



Protocols and entities used in K12-File Transfer Profile

The following roles are defined for this profile:

Client – The Client device initiates the operation, which pushes and pulls objects to and from the *Server*. In addition to the interoperability requirements defined in this profile, the Client must also comply with the interoperability requirements for the Client of the GOEP if not defined in the contrary. The Client must be able to interpret the OBEX Folder Listing format and may display this information for the user.

Server – The Server device is the target remote Bluetooth device that provides an object exchange server and folder browsing capability using the OBEX Folder Listing format. In addition to the interoperability requirements defined in this profile, the Server must comply with the interoperability requirements for the Server of the GOEP if not defined in the contrary.

OBEX (OBject EXchange) is a communications protocol that facilitates the exchange of binary objects between devices.

Operation no.	OBEX Operation	Client	Server
1	Connect	M	M
2	Disconnect	M	M
3	Put	M	M
4	Get	M	M
5	Abort	M	M
6	SetPath	M	M

OBEX operations used for File Transfer Profile

Header no.	OBEX Headers	Client	Server
1	Count	O	O
2	Name	M	M
3	Type	M	M
4	Length	M	M
5	Time	O	O
6	Description	O	O
7	Target	M	M
8	HTTP	O	O
9	Body	M	M
10	End of Body	M	M
11	Who	M	M
12	Connection ID	M	M
13	Authenticate Challenge	M	M
14	Authenticate Response	M	M
15	Application Parameters	X	X
16	Object Class	X	X

OBEX headers used for File Transfer Profile

'M' for mandatory to support

'O' for optional to support

'X' for excluded

The design of transferring data- file, authentication data, and configuration data- is not complete yet; all will be designed in following weeks using these profiles and their operations. Then it will completely be clear.

4.3. Broadcast

The order, the events will occur is:

1. Scan other discoverable bluetooth devices
2. Connect
3. Sent current image's event data
4. Repeat the Bluetooth broadcast packet several times to increase reliability of broadcast

According to our vice inspection on our bluetooth kit handbook, we have figured that part 3.2 is well suited for broadcasting. Although the overall mechanism building is still on progress, the modules can be provided now.

ATUCL {Clear Unit}

Recommend executing this before performing an inquiry command, places the radio in idle mode.

ATDI, <number>, <cod>

Inquiry for all devices around, a 20 second timeout occur while searching <number> of devices . This command is used to discover all Bluetooth radios (within range) that match the Class of Device (COD). If the COD is not known it is best to use 00000000 which allows discovery of all devices. You can not be in the default slave mode and perform the inquiry command. Only a Master or Radios in idle mode can perform an inquiry. and Returns the following:

<bt_Address 12chars>,<cod 8chars>,<name up to 16chars>

*Bluetooth device's address is used while connecting, so a list of these addresses will help us while broadcasting our packet.

ATLAST

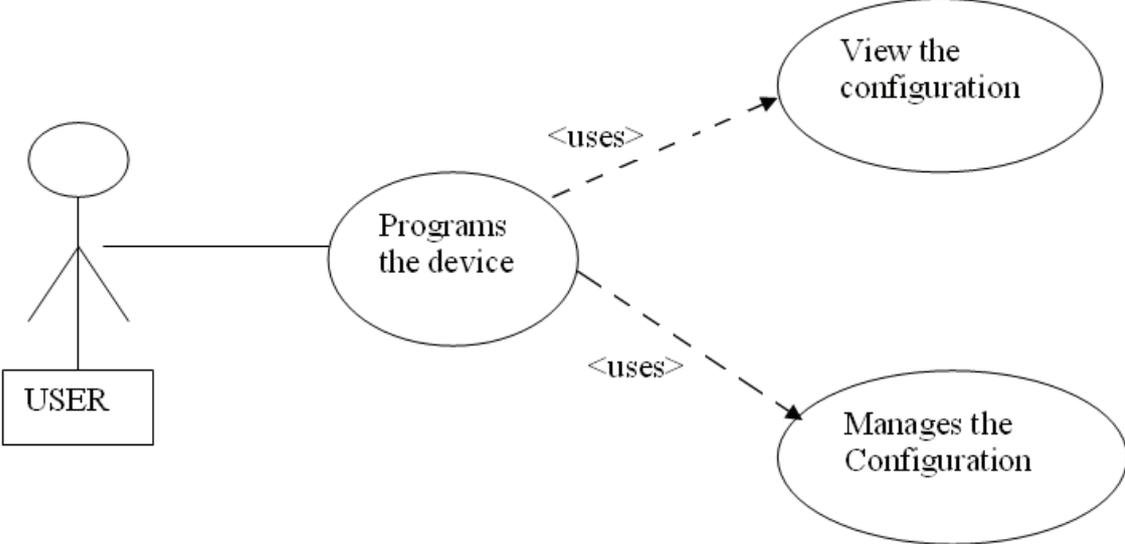
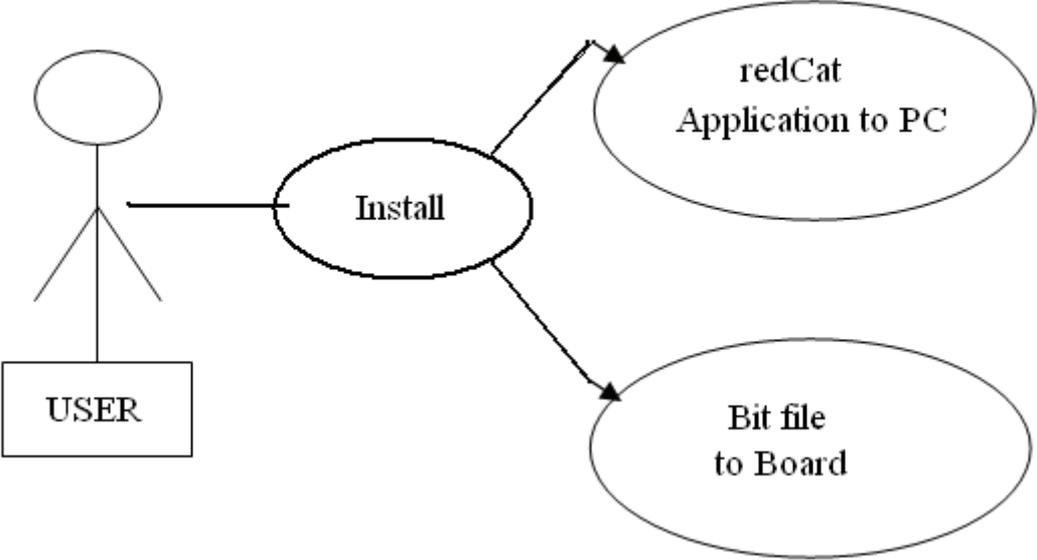
Last connected bluetooth devices address returned, can be used to double check if our connection list is finished

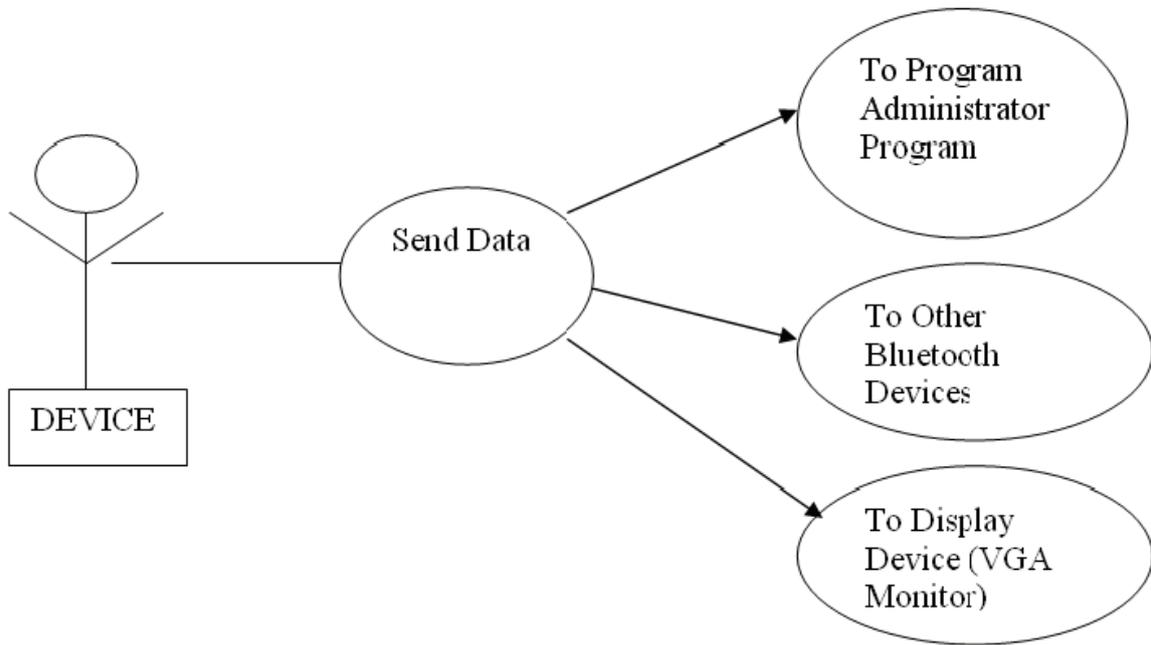
ATDM,< bt_Address >,<UUID>

This command gives the Slave address, and the type of profile that it will connect to/with.

For arranging the packet to be sent to the bluetooth enabled devices via our kit, the path followed would be from SDRAM to the user with this device. First event data is taken from the SDRAM with the help of the interface we will be building in between SDRAM and FPGA. Then the interface between the parallel port and the FPGA will help the information to be sent out of the board. With the circuit on the CLDP or with the help of the cable that switches serial data to parallel data, the parallel data will be sent to our bluetooth kit and be received as serial, here the kit is capable of converting this type of data to bluetooth and a packet is created here in order to be send to the slaves. For applications that require more than point-to-point devices communicating simultaneously – this is called a pico-net. These applications require one of the Bluetooth devices to manage all the network connections. Here, our master will be our bluetooth kit, and the broadcasting will be handled via keeping the list of the slaves on the board and sending the event data to the slaves in the list via this negotiated master, our bluetooth kit.

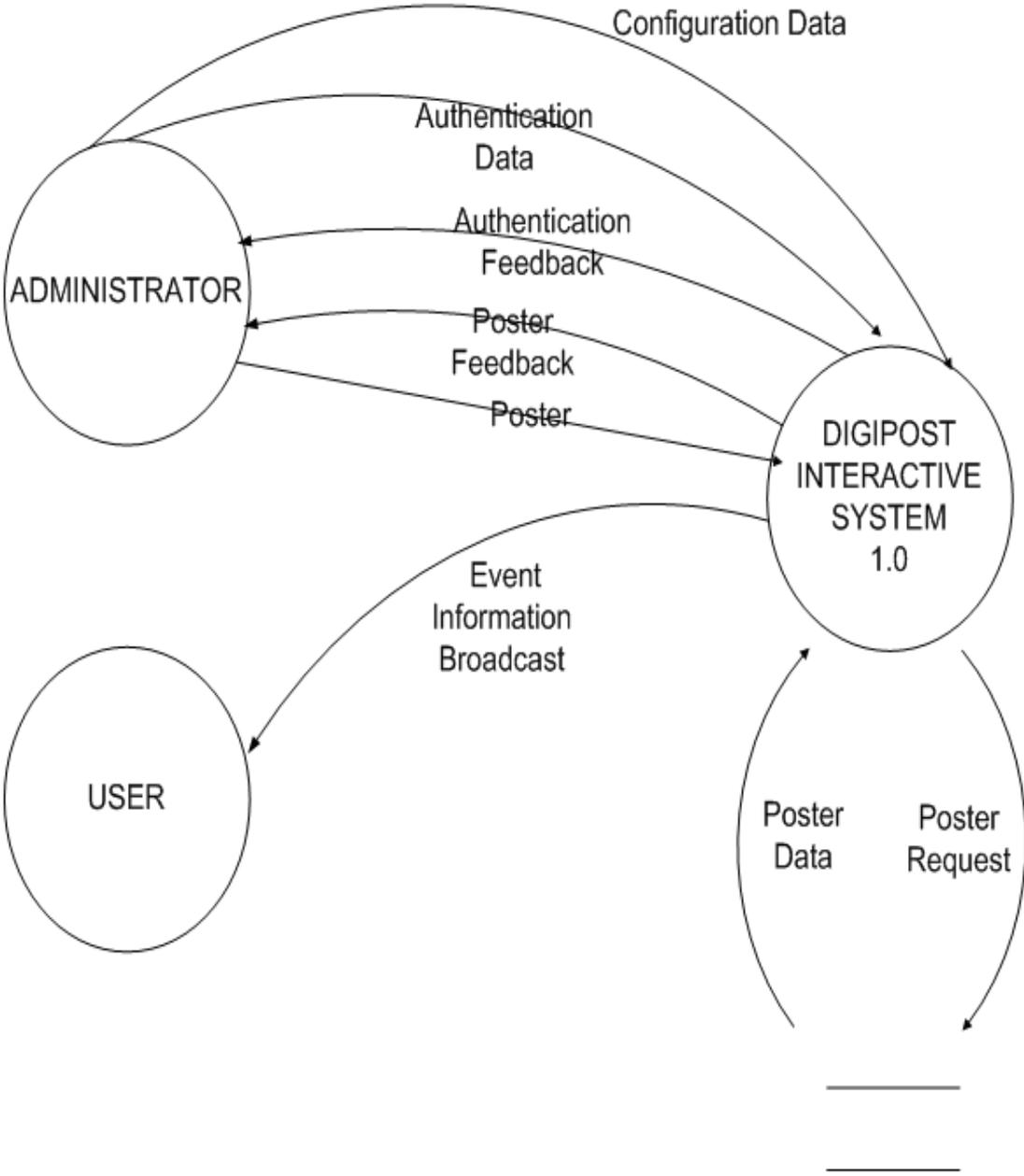
5. USE-CASE DIAGRAMS



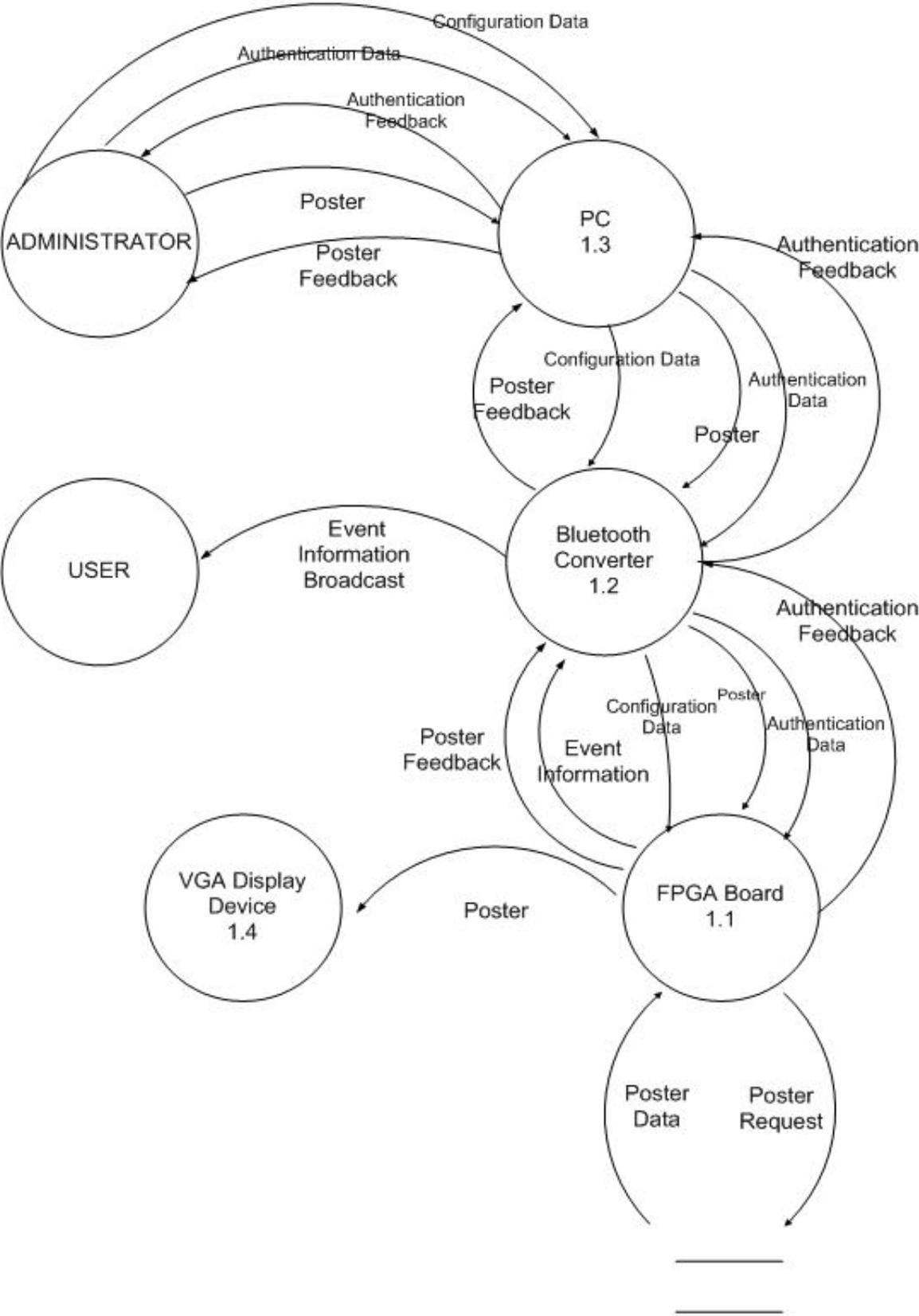


6. DATA FLOW DIAGRAMS

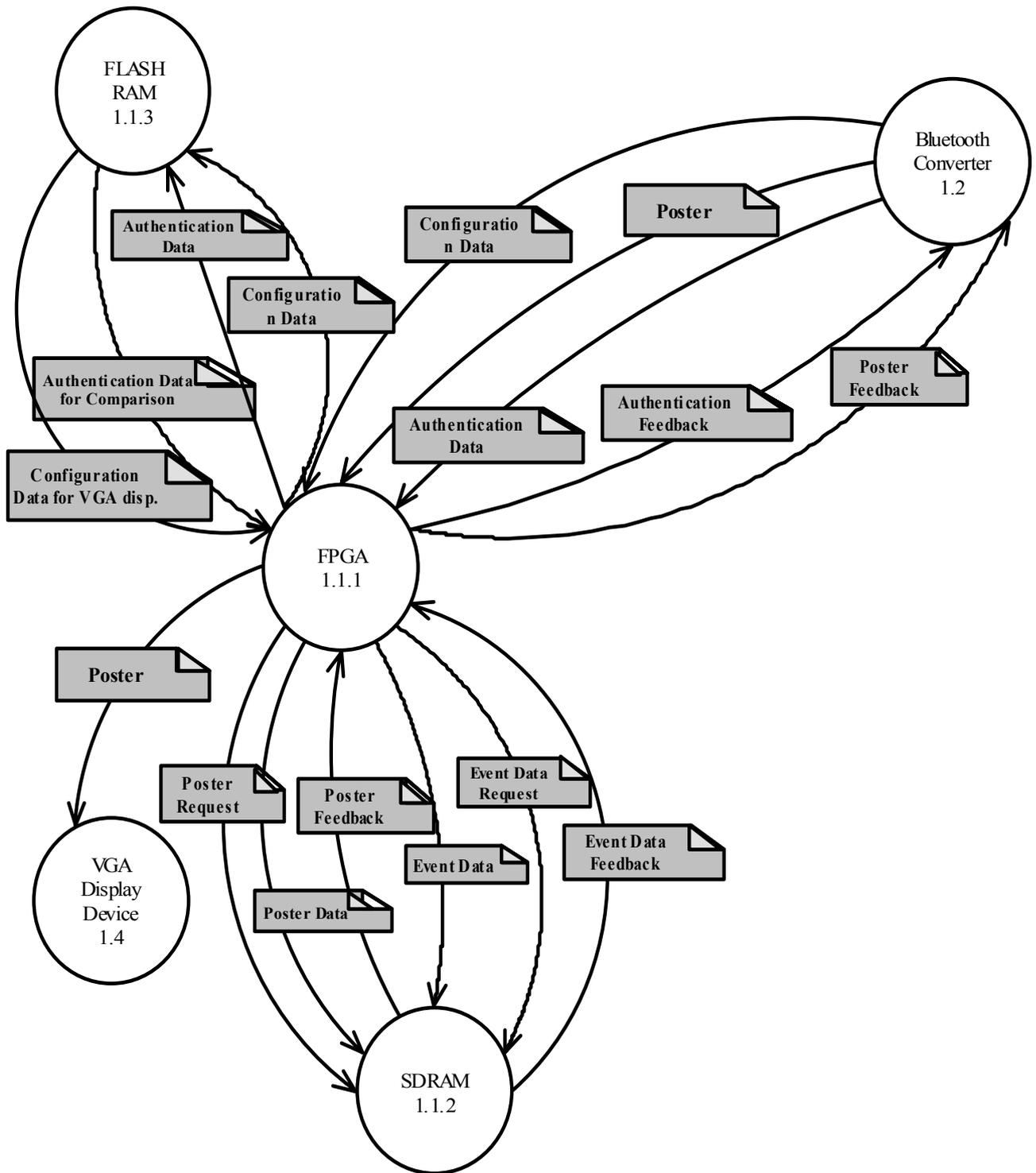
6.1. DFD0



6.2. DFD1



6.3. DFD2



Database in our DFD1 corresponds to SDRAM and FLASHRAM in our DFD2

Database

6.4. Data Dictionary

Name:	Poster, Poster Data
Aliases:	Image and event data
Where & how used	ADMINISTRATOR(output) PC(1.3)(input + output) Bluetooth Converter(1.2) (input + output) FPGA Board(1.1)(output) VGA Display Device(1.4)(input)
Description	Poster= image + event information 1. Admin loads poster to PC (input) 2. PC loads poster to FPGA Board through Bluetooth Converter (input) 3. FPGA Board keeps poster in memory (input) 4. FPGA Board loads poster image to VGA Display Device (input) 5. VGA Display Device displays image (output)

Name:	Poster Feedback
Aliases:	Done/ not accomplished
Where & how used	ADMINISTRATOR(input) PC(1.3)(input + output) Bluetooth Converter(1.2)(output)
Description	States whether uploading poster event has been accomplished or not

Name:	Event Information Broadcast
Aliases:	Event data of the poster
Where & how used	USER(input) Bluetooth Converter(1.2)(output)
Description	Event info=detailed information of an upcoming event in the format that is acceptable by the bluetooth communication

Name:	Event Information
Aliases:	Event data of the poster
Where & how used	FPGA Board(1.1)(output) Bluetooth Converter(1.2)(input)
Description	Event info=detailed information of an upcoming event

Name:	Poster Request
Aliases:	Selection query for a specific poster
Where & how used	Database(input) FPGA Board(1.1)(output)
Description	Poster is asked to be broadcasted to the bluetooth devices and poster is mainly saved in database, where database corresponds to FPGA Board's memory

Name:	Configuration Data
Aliases:	Amount of time for an image to be on the monitor
Where & how used	Administrator(output) PC(1.3)(input+output) Bluetooth Converter(1.2)(input+output) FPGA Board(1.1)(input)
Description	Administrator may want to change the config data on the board

Name:	Authentication Data
Aliases:	Username+password
Where & how used	Administrator(output) PC(1.3)(input+output) Bluetooth Converter(1.2)(input+output) FPGA Board(1.1)(input)
Description	For the admin, username and password should be typed in via the interface on the PC's software, and then this has to be compared on our FPGA board to see whether admin has the authentication

Name:	Authentication Feedback
Aliases:	0 1
Where & how used	Administrator(input) PC(1.3)(input+output) Bluetooth Converter(1.2)(input+output) FPGA Board(1.1)(output)
Description	Admin has failed or passed the authentication on the board

7. PROJECT SCHEDULING

7.1. Gantt Chart

ID	Task Name	Start	Finish	Duration	Eki 2006				Kas 2006				Ara 2006				Oca 2007						
					1.10	8.10	15.10	22.10	29.10	5.11	12.11	19.11	26.11	3.12	10.12	17.12	24.12	31.12	7.1	14.1	21.1		
1	Deciding on the company structure and the roles of the members in the team	03.10.2006	03.10.2006	1d																			
2	Gaining knowledge on the internet for a wider project description	03.10.2006	05.10.2006	3d	■																		
3	Learning the technical literature and the hardware to be used for the project	06.10.2006	08.10.2006	3d	■																		
4	Milestone1	08.10.2006	08.10.2006	0w	◆																		
5	Doing market research	16.10.2006	20.10.2006	5d	■																		
6	Task distribution among the members is achieved and Gantt Chart is formed	20.10.2006	21.10.2006	2d	■																		
7	Entity Relation diagram design& documentation is done	27.10.2006	29.10.2006	3d	■																		
8	Data Flow Diagrams are to be designed	30.10.2006	02.11.2006	4d	■																		
9	Behavioural Modelling is to be done	01.11.2006	03.11.2006	3d	■																		
10	Hardware and Software requirements are to be analyzed	01.11.2006	03.11.2006	3d	■																		
11	Scenario is to be written	02.11.2006	03.11.2006	2d	■																		
12	Final version of Req. Analysis Report	05.11.2006	05.11.2006	1d																			
13	Milestone2	05.11.2006	05.11.2006	0w	◆																		
14	Determining the design constraints	07.11.2006	07.11.2006	1d																			
15	Going over the modeling diagrams	08.11.2006	10.11.2006	3d	■																		
16	Examining the example hardware designs online	10.11.2006	13.11.2006	4d	■																		
17	Extensive search on bluetooth communication	13.11.2006	17.11.2006	5d	■																		
18	Search on serial to paralel conversion	18.11.2006	19.11.2006	2d	■																		
19	GUI implementation	19.11.2006	19.11.2006	1d																			
20	VGA display circuit design examination	19.11.2006	22.11.2006	4d	■																		
21	Communication examination	23.11.2006	26.11.2006	4d	■																		
22	Final version of Initial Design Report	19.11.2006	28.11.2006	1w 3d	■																		
23	Milestone3	03.12.2006	03.12.2006	0w	◆																		
24	Rehearsal of the presentation of our blueprint	04.12.2006	04.12.2006	1d																			
25	Presentations	05.12.2006	08.12.2006	4d	■																		
26	Reviewing initial design report	10.12.2006	14.12.2006	5d	■																		
27	Drawing the interface for admin and user	14.12.2006	18.12.2006	5d	■																		
28	Reviewing the model diagrams	18.12.2006	30.12.2006	1w 6d	■																		
29	Final version of the Final Design Report	30.12.2006	15.01.2007	2w 3d	■																		
30	Milestone4	15.01.2007	15.01.2007	0w	◆																		
31	Working on Prototype	20.11.2006	22.01.2007	9w 1d	■																		
32	Prototype demo	23.01.2007	23.01.2007	0w	◆																		

8. CONCLUSION

As stated in the Initial Design Report explanation of the ceng490 course, we have written a report ,including the formal specification of our system solution. We have described the system modules, data flow, data dictionaries, UML diagrams, and syntax specifications depending on our project and methodology. This report actually had the effect of keeping us moving on, since such a project needs great attention and research. Ready-to-use system modules has been searched to figure out how to make them collaborate, and be useful for our project. We still have some optional cases since this is a hardware project, and we can not risk ourselves by relying on a single hardware component for example; the serial to parallel converter.

9. REFERENCES

<http://www.blackbox.com/>

<http://www.xess.com/>

<http://www.blueradios.com/>

<http://whitepapers.zdnet.com/>

http://www.palowireless.com/infotooth/tutorial/k12_ftp.asp

<https://www.bluetooth.org/>

<http://www.swedetrack.com/usblue4.htm>