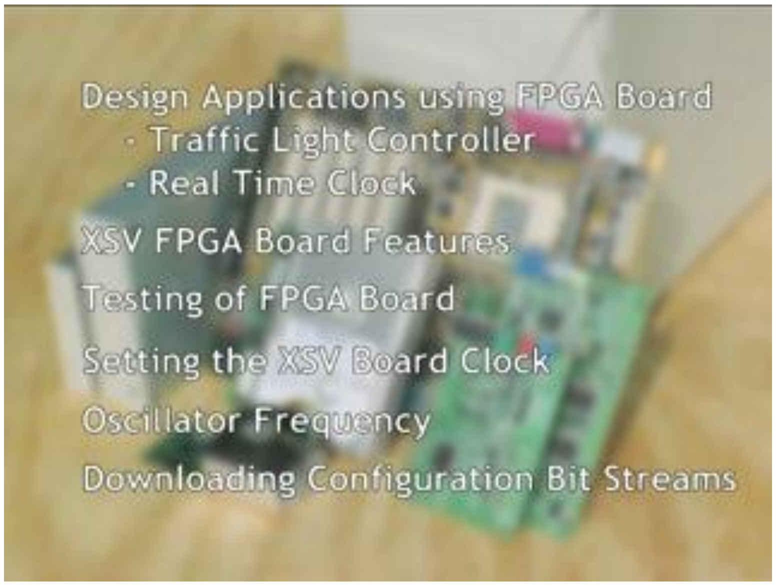
Digital VLSI System Design

Dr. S. Ramachandran
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture No. 48
System Design Examples using FPGA Board

(Refer Slide Time: 02:05)

Contents of Lecture 48



Design Applications using FPGA Board
- Traffic Light Controller
- Real Time Clock

XSV FPGA Board Features

Testing of FPGA Board

Setting the XSV Board Clock

Oscillator Frequency

Downloading Configuration Bit Streams

(Refer Slide Time: 02:39)

DESIGN APPLICATIONS USING FPGA BOARD

We have been looking at design applications in the previous classes, some of which are PCI arbiter. We have also seen an example of a video compression system and how to configure them based on PCI bus. For that, we needed a controller. That is what we developed on that occasion. Subsequently, we went for another application called a traffic controller. We have also seen a video compression algorithm, how to develop the algorithm and then finally code it in Verilog so as to make a video encoder. We have used the discrete cosine transform and quantization in order to effect the compression.

Thereafter, we need to gain experience on actual hardware and for that, we need to have FPGA boards. So far, we have only seen the CAD tools namely the ModelSim tool, to do the simulation, to get the timing diagram and analyze the circuit and then based on that, study. Thereafter, we also used the Synplify tool in order to do the logic synthesis. It not only maps the device but you also get an RTL view, etc. in that synthesis tool. It also generates an EDF file, which we had taken for Xilinx place and route – we mapped the actual device and done place and route.

We have also seen back annotation so as to reflect the actual gate delays and then plough it back into the ModelSim simulator so that we could see the actual gate delays incorporated. Frequency operation reported by the place and route is what is to be actually taken into account when we design the system. Once we run it through place and route, in this case Xilinx place and route that we have already seen, then the frequency of operation reported by that is be more or less going to be operating on the FPGA board, which we are going to see presently.

This is what is going to be done in the next few lectures. Prior to this, we need to become familiar with some of these boards. There are many vendors as far as the boards are concerned, namely XESS and Avnet and even Xilinx has some boards available. Plenty of other manufacturers are also available for making FPGA-based boards. They have many features, which you can readily use in many of the applications that you need for your own development.

(Refer Slide Time: 06:20)

XESS Corporation

XSV Board V1.1 Manual

How to install and use your new XSV Board

Let us first have a look at XESS board. This is from XESS Corp. If you go into the site www.xess.com (note the spelling – x, e, double s), you can download all the manuals. When you purchase these boards, you will also get a copy on the CD along with the software that goes with this. This is what is called XSV board and what we have is version 1.1. This has XCV-800 as the device, which we will be seeing shortly.

(Refer Slide Time: 07:07)

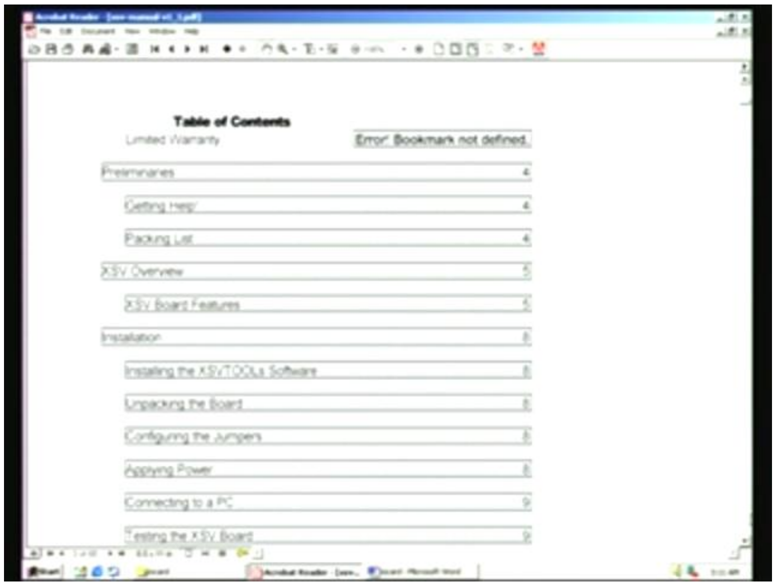
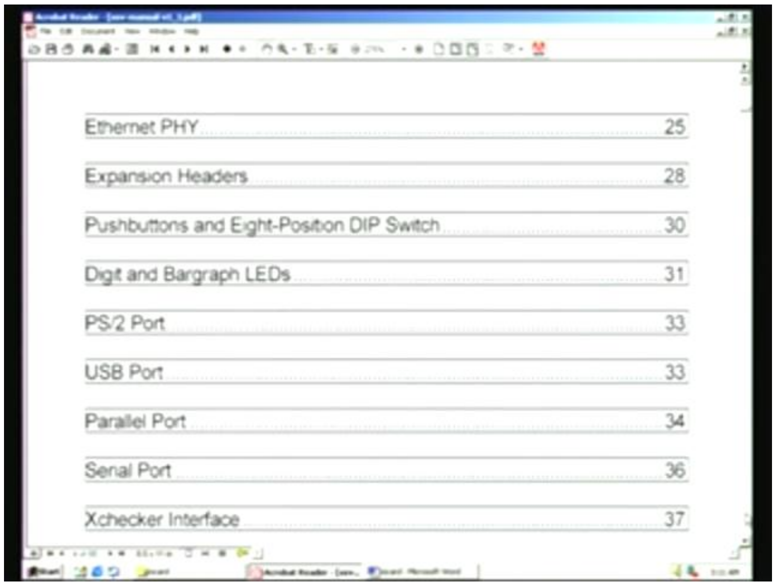
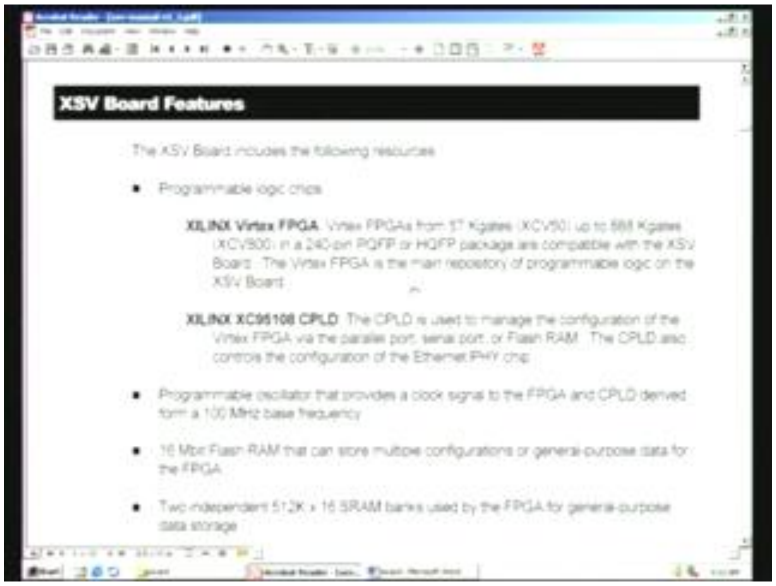


Table of Contents



This is user manual for the board from XESS Corp. Let us go to the relevant point so that we can see what the board contains.

(Refer Slide Time: 07:20)



XSV Board Features

The XSV Board includes the following resources:

- Programmable logic chips

  XILINX Virtex FPGA: Virtex FPGAs from 57 Kgates (XCV50) up to 888 Kgates (XCV800) in a 240-pin PQFP or HQFP package are compatible with the XSV Board. The Virtex FPGA is the main repository of programmable logic on the XSV Board.

  XILINX XC95108 CPLD: The CPLD is used to manage the configuration of the Virtex FPGA via the parallel port, serial port, or Flash RAM. The CPLD also controls the configuration of the Ethernet PHY chip.

- Programmable oscillator that provides a clock signal to the FPGA and CPLD derived from a 100 MHz base frequency.
- 16 Mbit Flash RAM that can store multiple configurations or general-purpose data for the FPGA.
- Two independent 512K x 16 SRAM banks used by the FPGA for general-purpose data storage.

These are all the features of the XESS board. This contains programmable logic chips. There are two chips available. One is for your own application, wherein you have to download the bit stream that we have got in place and route. This is the place where you have to download your bit streams. That is based on Xilinx Virtex FPGA and the device is XCV800. On many occasions, I think we have used some such device in our design earlier. It is a 240-pin PQFP package. This is what is housed on the board. We will see the board pictorially as well as the actual hardware.

We will also see the whole thing configured for one of the applications. We may be looking into some of the applications that we have designed or are going to design. This particular FPGA is not socketed. Some of the FPGA boards are socketed. I do not think this present board is socketed, which we can check later. Otherwise, you may have to, if you want to change it to different FPGAs of different capacities. For example, XCV50 is for 50,000 gates capacity. Like that, they have many FPGAs available on different cards.

We are going to demonstrate based on this XCV800. This is at the highest end of their product range. When new devices come, they may incorporate that and bring about a different board for that. The capacity 800 stands for 800,000 – it actually goes for 8888 kilogates. This is on this XSV board. This is the main FPGA in which we will be downloading our application programs. For example, if you want to put a traffic controller, which we have designed before, we need to download that bit stream to this FPGA here.

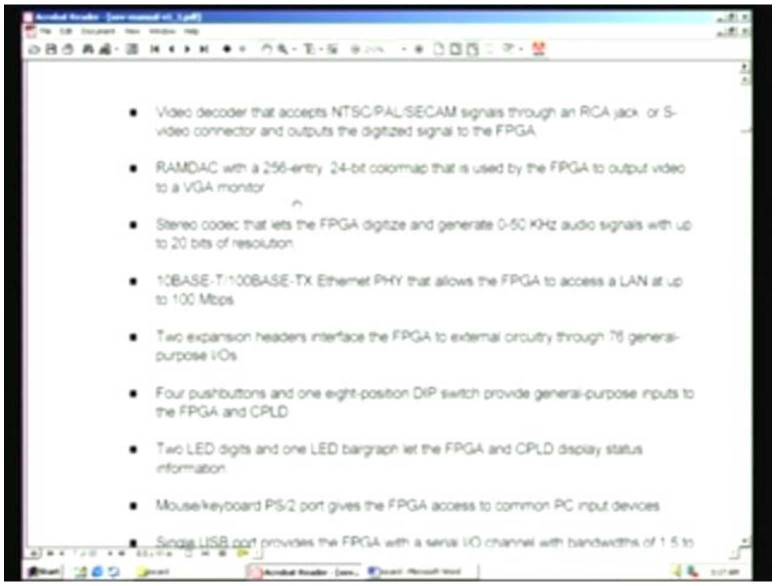
There is also another FPGA equivalent. This is called complex programmable logic device abbreviated as CPLD. This will not contain as many gates as XCV800 – it may go only for a few thousand gates. This CPLD is used to manage the configuration of the Virtex FPGA. Virtex FPGA is what we are having here. It will take in the bit stream for your application. Whatever application you download will be on this FPGA and this circuit is configured on this FPGA.

In order to manage it, you need this CPLD as well. It manages the configuration of the Virtex FPGA via the parallel port. There is also a parallel port on the board, which can be connected to the parallel port of the standard computer, say a personal computer such as Pentium and LPT1 or LPT2. There is a provision for selecting any of these that you want. It also has a serial port through which RS-232 type of communication can be undertaken. You can use this for any of your applications or you can even download the bit streams through the serial port but it will be time consuming. It is preferable that you take the parallel port and this is what we are going to adopt.

It also has flash RAM. Flash RAM is a non-volatile RAM – it will retain its contents even with the power switched off. It is a convenient media in which you can put your bit stream, that is, the application circuit that you have configured can be put into the flash RAM. Even with the power off, the code is not disturbed and the bit stream is not disturbed. When you switch on the power once again, it can immediately start working on your code or rather the circuit that you have configured. The CPLD also controls the configuration of the Ethernet chip. You have an Ethernet chip also available right on the card. It is also capable of having different clock speeds for FPGA operation and it goes right up to 100 Megahertz. You can also have lower values. We will see shortly what the different ranges are and how to program – we will see all that.

It also has 16 megabits of flash RAM. We have already seen flash RAM. It can store multiple configurations, which means that different bit streams can be stored at different locations and retrieved at any point of time for operation, or it can have general-purpose data. For example, you may be working on video compression, for which you may require huge storage. In this case, you can use this flash RAM. Since it is non-volatile, it will be very handy to use. It also has two independent 512K into 16 static RAM banks, organized as 8 bits. That is the byte-oriented thing. In fact, there are two chips in order to give you 1 megabyte of storage. You have two sets like this so you get a total of 2 megabytes of RAM, which is connected to the I/O pins of the FPGA. That FPGA is on Xilinx FPGA and part of it could be on CPLD or wholly CPLD, which we will be seeing when we see the actual pin configuration.

(Refer Slide Time: 13:42)



- Video decoder that accepts NTSC/PAL/SECAM signals through an RCA jack or S-video connector and outputs the digitized signal to the FPGA
- RAMDAC with a 256-entry 24-bit colormap that is used by the FPGA to output video to a VGA monitor
- Stereo codec that lets the FPGA digitize and generate 0-50 KHz audio signals with up to 20 bits of resolution
- 10BASE-T/100BASE-TX Ethernet PHY that allows the FPGA to access a LAN at up to 100 Mbps
- Two expansion headers interface the FPGA to external circuitry through 76 general-purpose I/Os
- Four pushbuttons and one eight-position DIP switch provide general-purpose inputs to the FPGA and CPLD
- Two LED digits and one LED bargraph let the FPGA and CPLD display status information
- Mouse/keyboard PS/2 port gives the FPGA access to common PC input devices
- Single USB port provides the FPGA with a serial I/O channel with bandwidths of 1.5 to

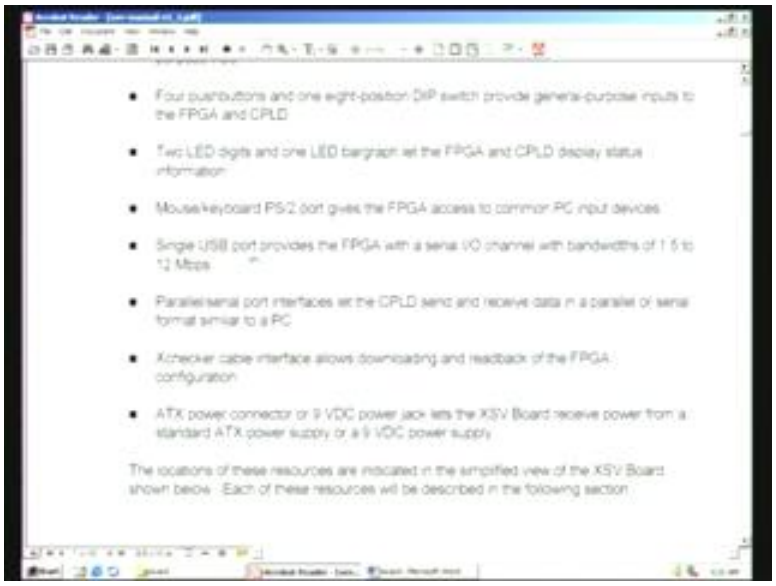
Next, you have very good features. For example, a video decoder is available in order to cater to NTSC/PAL/SECAM signals. This is the [13:51] signal that you will be getting and you can collect this and convert that analog signal into digital signal. This is what we refer to as digitization. It is through RCA jack or S-video connector. There is a video decoder chip on the board and that brings about this digitization. It outputs to the FPGA for further processing such as video compression, which we have seen earlier. That being involved, we may not cover this on the board – we may cover simpler applications such as traffic light controller and so on.

It also has a RAM plus DAC. DAC stands for the digital to analog converter and this can be used for converting color. For example, you can display the NTSC you have received on the computer monitor that you have right here if you want to – that is called the VGA interface or VGA monitor. It can be connected through this RAMDAC chip. It has 256 entries available – that much storage is available on this. You can configure for true color – you can configure for, say, a 24-bit color map. That is used by the FPGA to output video to a VGA monitor. In addition to this, it also has a decoder, a stereo codec on the board. It also does the digitization of the actual audio analog signal. Although the audio range is only from 20 Hertz to 20 Kilohertz, this stereo codec is capable of processing right from 0 to 50 Kilohertz. Once digitization is involved, you speak of resolution – number of bits. It can handle up to 20 bits.

We have already seen that you have an Ethernet chip for connecting it to a LAN and up to 100 Mbps link is possible with this board. You also have some expansion headers, which we will use for some of the applications that we are going to consider. Two expansion headers are available right on the board and they are all connected to the FPGA I/O pins. The total number of such I/Os is 76, putting together both the expansion headers.

Right on the card, you also have four push buttons, which you can configure for any of your applications. One eight-bit DIP switch is also available and it provides general-purpose inputs to FPGA as well as the CPLD. In addition to this, you have two seven-segment LEDs and one bar graph LED. I think you can display 10 bars. It lets the FPGA as well as CPLD display status information. You can display any information that you want here. Any parameters or engineering units that you are processing in your application can also be displayed using some of these displays.

(Refer Slide Time: 17:17)



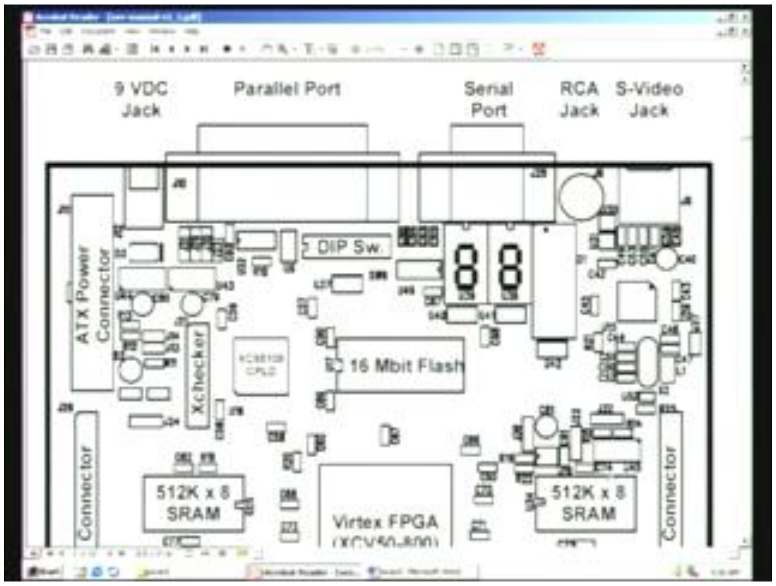
- Four pushbuttons and one eight-position DIP switch provide general-purpose inputs to the FPGA and CPLD
- Two LED digits and one LED bargraph let the FPGA and CPLD display status information
- Mouse/keyboard PS/2 port gives the FPGA access to common PC input devices
- Single USB port provides the FPGA with a serial I/O channel with bandwidths of 1.5 to 12 Mbps
- Parallel/serial port interfaces let the CPLD send and receive data in a parallel or serial format similar to a PC
- Xchecker cable interface allows downloading and readback of the FPGA configuration
- ATX power connector or 9 VDC power jack lets the XSV Board receive power from a standard ATX power supply or a 9 VDC power supply

The locations of these resources are indicated in the simplified view of the XSV Board shown below. Each of these resources will be described in the following section.

It also has a mouse–keyboard interconnection through a port called PS/2 and this is connected to the FPGAs. It offers the FPGA access to common PC input devices. You have an USB port. It is a serial link and it is connected to the FPGA through a special chip right on the chip. It has a bandwidth of 1.5 Mbps to 12 Mbps – this is the rate at which you can communicate serially. You also have parallel as well as serial ports available on this. We will use a parallel port in order to download the bit stream that we have for our applications. Parallel/serial port interfaces let the CPLD send and receive data in a parallel or serial format similar to a PC. This is precisely the same as a PC LPT printer interface, which is a parallel port that you are already familiar with.

It also has another way of communicating the bit stream in addition to the parallel or serial fashion and that is by a special cable called XChecker cable. Of course, we will not be using this here. This also helps in downloading your bit stream. You can also read back what you have already configured on the FPGA. The board will have to be connected to an external power supply. You can connect either 9 Volts power jack or a specially connected ATX power connector right on the board. Both of them are on the board. It lets the XSV board receive power from a standard ATX power supply. This is [19:07] supply, which we do not have. In view of this, we have just a simple power supply – 9 Volts supply. If you have an adopter for this, you can use it but current rating will have to be at least 1.5 Amps. That is what you have to take into account. Normally, the board demands 5 Volts and 3.3 Volts or perhaps a 2.5 Volts DC is also required. All are derived right from 9 Volts, which we can supply externally or you can supply from this ATX power supply. Having seen all these features of the board, we can see pictorially how the board looks.

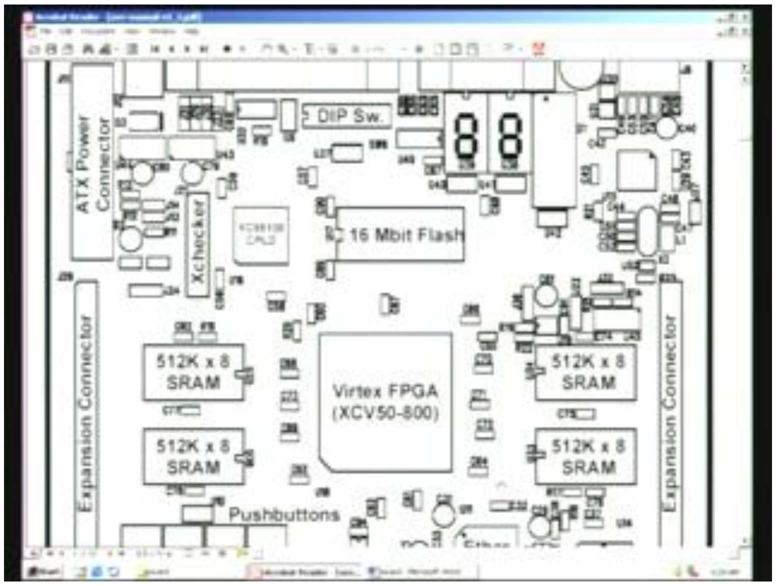
(Refer Slide Time: 20:00)



This is how the board looks. You can see there is a jack right on the top. You can connect the conventional 9 Volts jack and supply power. The power requirement is that you must have at least 1.5 Amps. We also need the parallel port interconnection. Associated with this will be a parallel port cable. This will go into the LPT1 or LPT2, which you can configure on the host Pentium processor. It is through this cable that you are going to download into the Virtex FPGA, which is here. Before we arrive at this, we will see all other features in the same order.

In addition to this parallel port, we also have a serial port and you can use it for any of the applications. For example, if you want to have an RS-232 type of serial link between your board and PC (personal computer), you can have a serial cable connected. Whatever application you have configured, you can download your data information through the serial port to the host system and vice versa and thereby, you can configure a total system for catering to a particular application. For example, if you want to do a data [21:30] system, you can send any data based on real time. You can send data to and fro between this FPGA board and also the personal computer. For that, you can use a serial port. We have already seen that one RCA jack and S-video jack are available here. There is also a chip, which is the video decoder. This takes in the NTSC or PAL/SECAM [22:02] signal and routes it to the Virtex FPGA pins. Via the I/O pins, you can transact whatever image you have collected via this signal, be it NTSC or PAL signal. We can route it to the FPGA and do further processing such as compression.

(Refer Slide Time: 22:35)

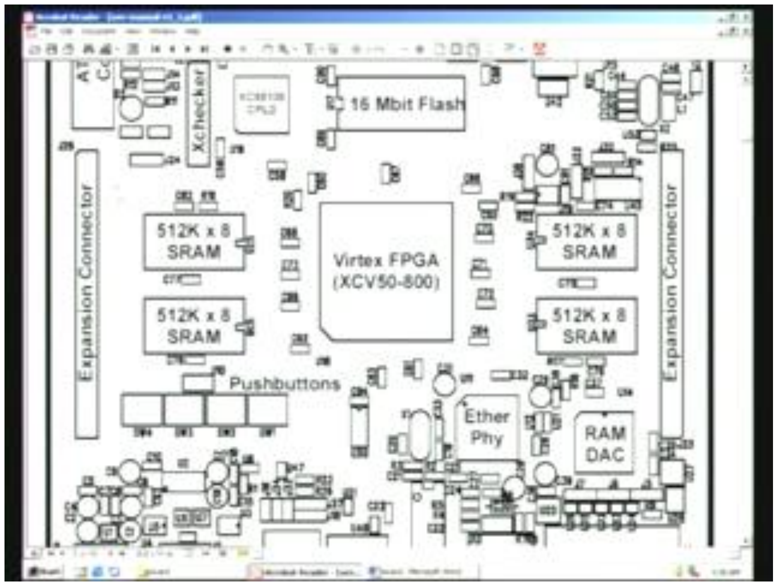


Next, we also have a power supply connector. That is what needs to be connected here. You connect either this or the 9 Volts. We are going to use only the 9 Volts connection, not this. We have also seen that an XChecker interconnection is there. We are not going to use this. Instead of this, we are going use the parallel interface in order to connect it to a PC LPT1.We have switches here. For example, an eight-bit DIP switch is available here. You can use it for any of your applications. For example, if you want to set a particular value (a timing value), suppose you are running a timer, you can set your timer, etc., right at the DIP switch. You can set any other parameter you wish to set through the DIP switch. This will be connected to the CPLD, which is housed here, as well as the Virtex FPGA.

What we have is on board XCV800 – it is capable of nearly 900,000 gates. We can put a huge circuitry inside this. Our application is going to reside here. It is going to come through this parallel port to this FPGA. We are going to download .bit. For example, if it is a traffic controller, we are going to generate a traffic_controller.bit by using place and route. We will download that particular bit stream to this. For this, you need software associated with this board. XESS also supplies software that we need to load onto the system and that will have to be run. Only then can you communicate with this board effectively. In fact, without that, it is almost impossible to communicate with this. You must have this.

In addition to this CPLD, we also have two digits seven-segment LEDs, which you have already seen. We will see some of these details a little later – how you go about connecting different segments. In addition to this, we mentioned that there is a 10-bar segment. That is what is here. In fact, similar segments will be placed here – one segment here, one below another and all ten will be available here. This is a bar display and you can use it for display of histograms, which you may design in some of your engineering units; you can also use this for such an application and so also the two LEDS. You also have a 16 megabit flash RAM available and that is housed here. In addition to this, we have 2 megabytes of static RAM storage. As I mentioned, they are all arranged as 512K bytes here. Two such devices would give you 512K into 16, which we have already seen in the features earlier. We have two sets like this here and so, in totality, you get 2 million bytes of RAM storage.

(Refer Slide Time: 26:00)

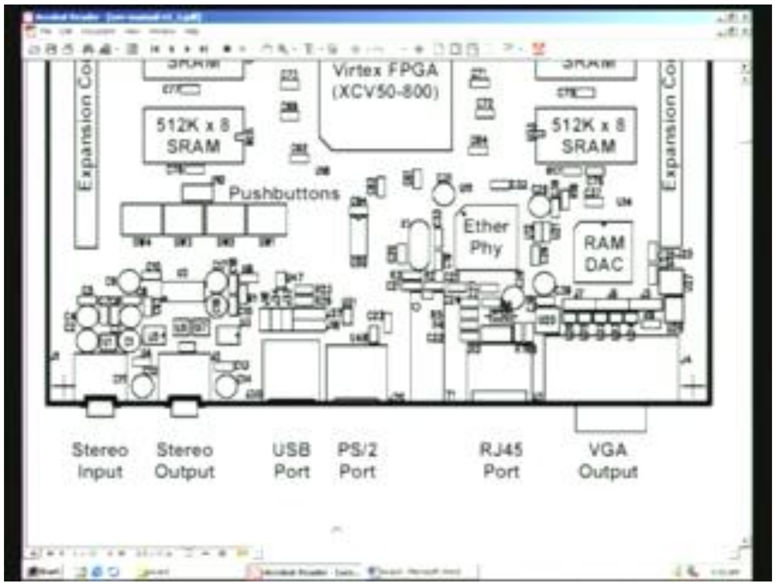


Further, you have this expansion connector. Suppose you are not satisfied with the storage available – the 2 million bytes storage, you can expand it by using this expansion connector or while expanding, you may have to disable this RAM. If you want to put much more storage or you do not want to use the internal storage, you may have to put an extra card outside and connect that card to these expansion connectors available on both sides.

While doing so, you may have to disable this by an appropriate jumper connection, which is available on the board. All these details are available in the manual, part of which you are already having a look at right now. We are not going to put an external RAM. We are not even going to use the internal RAM for one of the applications to start with. We will go for the traffic light controller, which we are already familiar with. That will be the first application we will be taking.

For that particular application, we need this expansion connector and we do not need any other resources we have already seen, except for the power supply and the parallel port connection. We will see more details of the expansion connector later on. We have also seen that there are four push button switches and they are all placed in this order. SW1, 2, 3 and 4 are placed in this order. We have an Ethernet chip available here. We have also seen that RAMDAC is used for VDU.

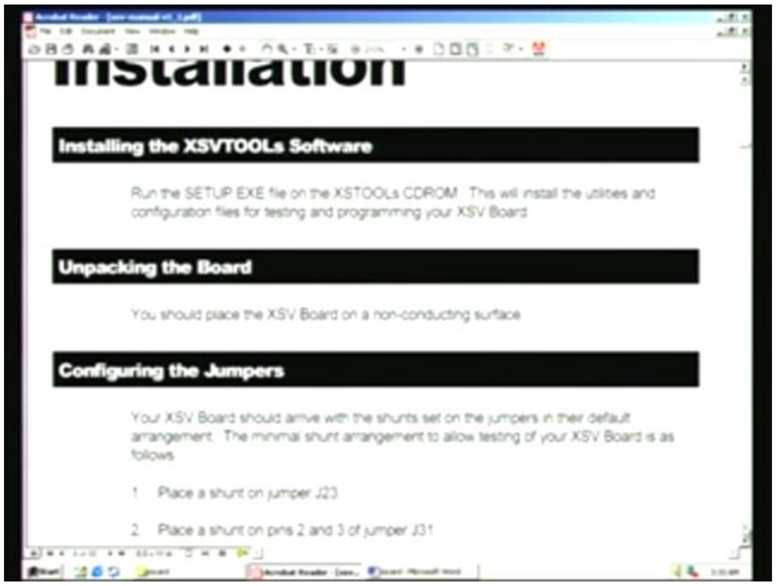
(Refer Slide Time: 27:55)



VGA monitor connection is here. VGA output is available through this connector. The processing chip is available here. We also have a port for mouse–keyboard connection and a USB port is available here. For all these, different chips are also used here, one of these here. We also have a stereo input. You can give the audio input here. For example, if you want to compress the audio, you can route this particular thing. It has a chip here that digitizes and gives a 20-bit resolution output, which can be connected to the FPGA and you can do the compression of sound. You can do mp3 for audio layer. If you want, you can do such a project using this. For that purpose, you can use this. You can use it for any other processing of audio signal.

Since only the digitized signals are available at the FPGA pins, you can do any processing that you wish. You can have a USB port, which we have already seen, and it is capable of up to 12 Mbps. You can establish a serial link in addition to this RS-232 serial link that we have already seen. That will only go for 19600 Bauds, which means bit per second. We have had a look at the board features. We will actually have a look at the board after a while.

(Refer Slide Time: 29:42)


Installation

Installing the XSVTOOLs Software

Run the SETUP.EXE file on the XSTOOLs CDROM. This will install the utilities and configuration files for testing and programming your XSV Board.

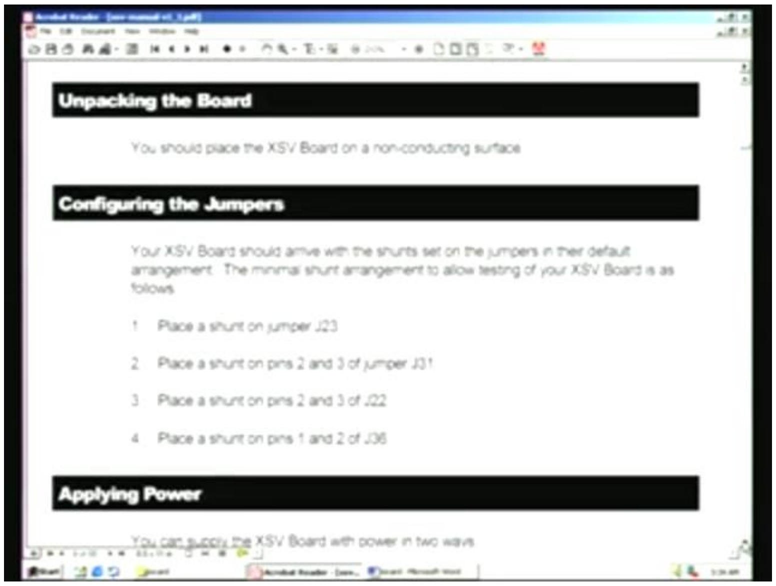
Unpacking the Board

You should place the XSV Board on a non-conducting surface.

Configuring the Jumpers

Your XSV Board should arrive with the shunts set on the jumpers in their default arrangement. The minimal shunt arrangement to allow testing of your XSV Board is as follows:

1. Place a shunt on jumper J23.
2. Place a shunt on pins 2 and 3 of jumper J31.


Unpacking the Board

You should place the XSV Board on a non-conducting surface.

Configuring the Jumpers

Your XSV Board should arrive with the shunts set on the jumpers in their default arrangement. The minimal shunt arrangement to allow testing of your XSV Board is as follows:

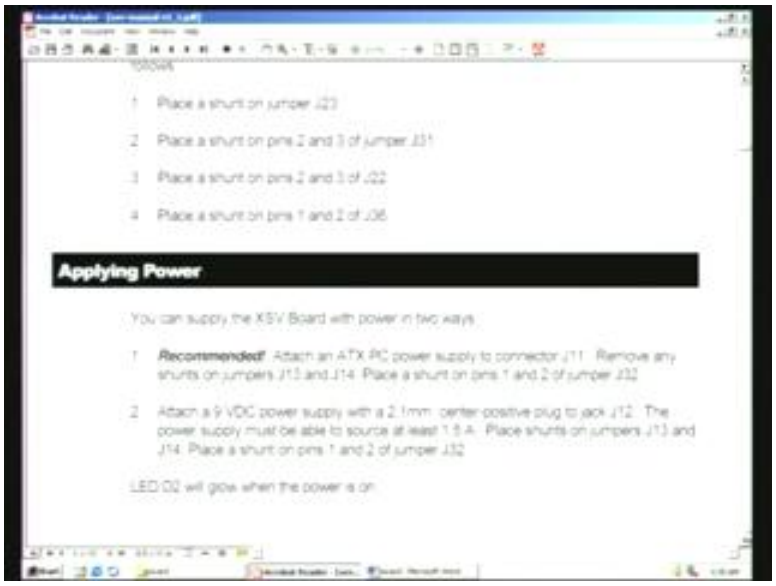
1. Place a shunt on jumper J23.
2. Place a shunt on pins 2 and 3 of jumper J31.
3. Place a shunt on pins 2 and 3 of J22.
4. Place a shunt on pins 1 and 2 of J36.

Applying Power

You can supply the XSV Board with power in two ways:

Before this, the manual gives lots of jumper settings, etc., which you will have to go through yourself. There is also a software associated with that, which we will have to install, as I mentioned before. We will skip this, because it is a matter of details pertaining to how to configure.

(Refer Slide Time: 30:04)



follows

1. Place a shunt on jumper J23
2. Place a shunt on pins 2 and 3 of jumper J31
3. Place a shunt on pins 2 and 3 of J22
4. Place a shunt on pins 1 and 2 of J36

Applying Power

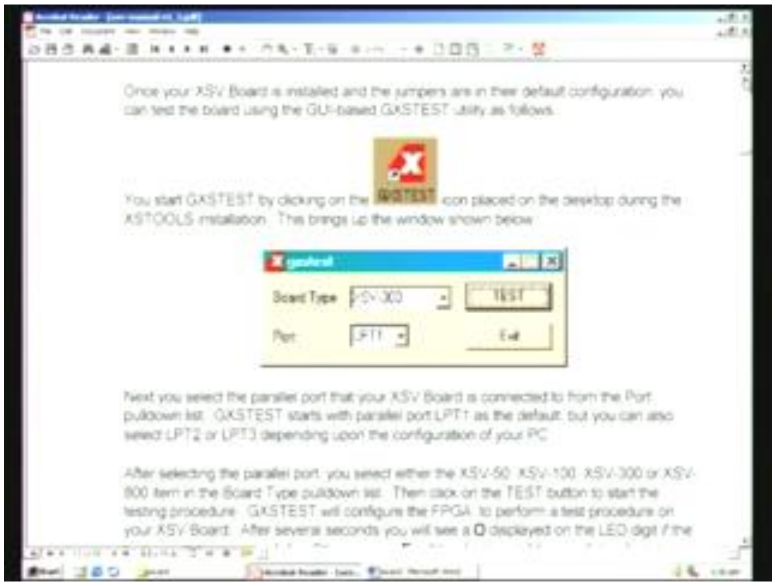
You can supply the XSV Board with power in two ways

1. ***Recommended!*** Attach an ATX PC power supply to connector J11. Remove any shunts on jumpers J13 and J14. Place a shunt on pins 1 and 2 of jumper J32
2. Attach a 9 VDC power supply with a 2.1mm center-positive plug to jack J12. The power supply must be able to source at least 1.5 A. Place shunts on jumpers J13 and J14. Place a shunt on pins 1 and 2 of jumper J32

LED D2 will glow when the power is on.

What is important is that we have already seen how to apply power. There is a jack and there are two ways of applying power: either you go for 9 Volts, which we are going to do through a separate jack, or ATX power supply supplied by XESS Corp., for which you need to connect to a different connector. Notice that 1.5 Amps is the actual requirement. In practice, it is taking less than 1 Amp but when you configure the clock, etc., it may touch 1.25 Amps or so. If you put 1 Amps supply, you will get into trouble. It will start malfunctioning. So be on your guard while selecting the power supply. Ensure that it is capable of supplying at least 1.5 Amps. When supply is applied, there is also an LED to indicate that the power is on.

(Refer Slide Time: 31:06)



Once your XSV Board is installed and the jumpers are in their default configuration, you can test the board using the GUI-based GXSTEST utility as follows.

You start GXSTEST by clicking on the GXSTEST icon placed on the desktop during the XSTOOLS installation. This brings up the window shown below.

gxstest

Board Type: XSV-300

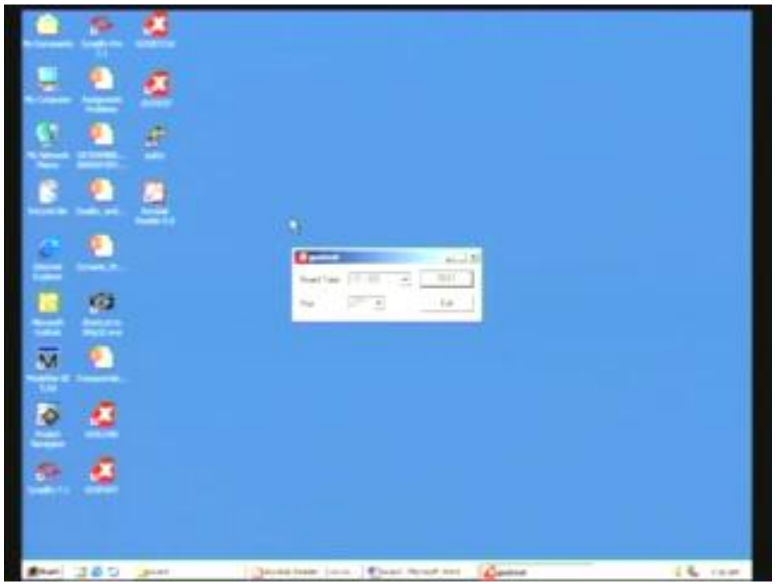
Port: LPT1

Next you select the parallel port that your XSV Board is connected to from the Port pulldown list. GXSTEST starts with parallel port LPT1 as the default, but you can also select LPT2 or LPT3 depending upon the configuration of your PC.

After selecting the parallel port, you select either the XSV-50, XSV-100, XSV-300 or XSV-800 item in the Board Type pulldown list. Then click on the TEST button to start the testing procedure. GXSTEST will configure the FPGA to perform a test procedure on your XSV Board. After several seconds you will see a O displayed on the LED digit if the

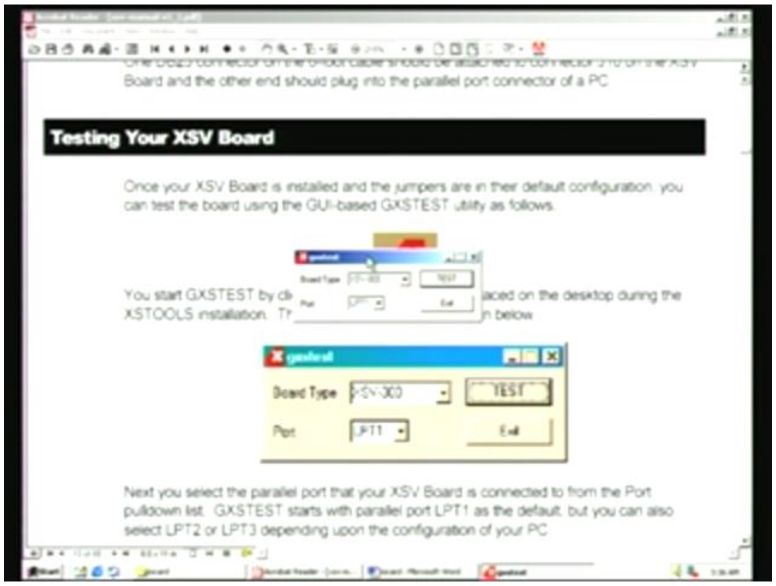
We have to follow certain procedures as far as this board is concerned in order to use it for your application. The board connection is quite simple. To the FPGA, the board is connected, cabled to the parallel port and connected straightaway to the LPT1 printer or LPT2 – any parallel port on your host system can be connected. In addition to the power supply, that is the only connection that you really require on the board. We will be seeing this later on. Once you have connected and supplies are on for the system as well as the host system, you have what is called a separate window. For example, this window says gxstest. I will first open this.

(Refer Slide Time: 32:05)



It will be very difficult for you to look at the screen but you can make out what it is. There is an icon for text. If I double click, it opens this window. It may be very difficult for you to view. Therefore, I will go back to the same thing, gxstest. This is a bloated up view. You notice that you can select different boards. For example, if it is XCV800, you have to click here.

(Refer Slide Time: 32:40)



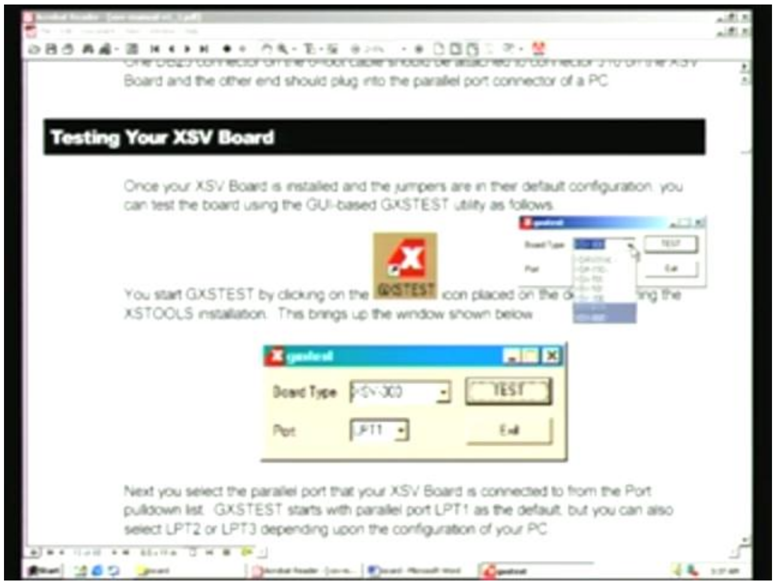
Board and the other end should plug into the parallel port connector of a PC

Testing Your XSV Board

Once your XSV Board is installed and the jumpers are in their default configuration, you can test the board using the GUI-based GXSTEST utility as follows.

You start GXSTEST by cl aced on the desktop during the XSTOOLS installation. Th n below

Next you select the parallel port that your XSV Board is connected to from the Port pulldown list. GXSTEST starts with parallel port LPT1 as the default, but you can also select LPT2 or LPT3 depending upon the configuration of your PC.



Board and the other end should plug into the parallel port connector of a PC.

Testing Your XSV Board

Once your XSV Board is installed and the jumpers are in their default configuration, you can test the board using the GUI-based GXSTEST utility as follows.

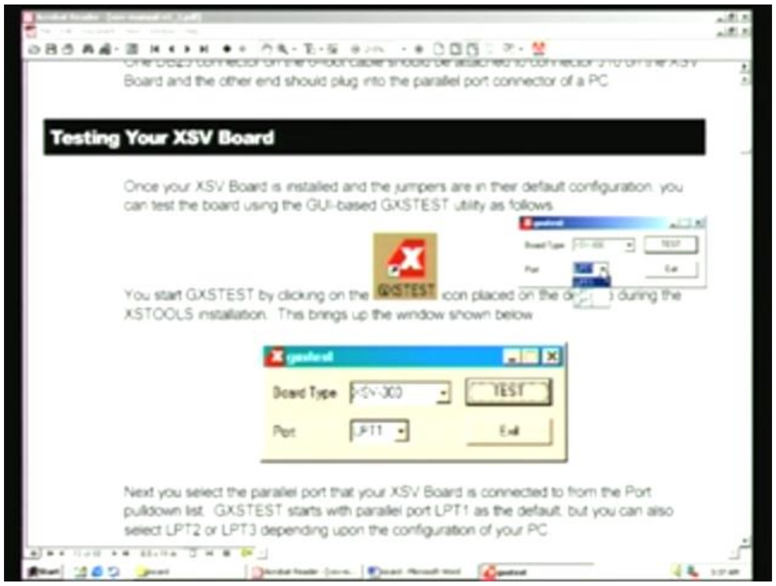
You start GXSTEST by clicking on the GXSTEST icon placed on the d[illegible] ing the XSTOOLS installation. This brings up the window shown below.

gxstest

Board Type: XSV-300 | TEST

Port: LPT1 | Exit

Next you select the parallel port that your XSV Board is connected to from the Port pulldown list. GXSTEST starts with parallel port LPT1 as the default, but you can also select LPT2 or LPT3 depending upon the configuration of your PC.



Board and the other end should plug into the parallel port connector of a PC.

Testing Your XSV Board

Once your XSV Board is installed and the jumpers are in their default configuration, you can test the board using the GUI-based GXSTEST utility as follows.

You start GXSTEST by clicking on the GXSTEST icon placed on the d[illegible] during the XSTOOLS installation. This brings up the window shown below.

gxstest

Board Type: XSV-300 | TEST

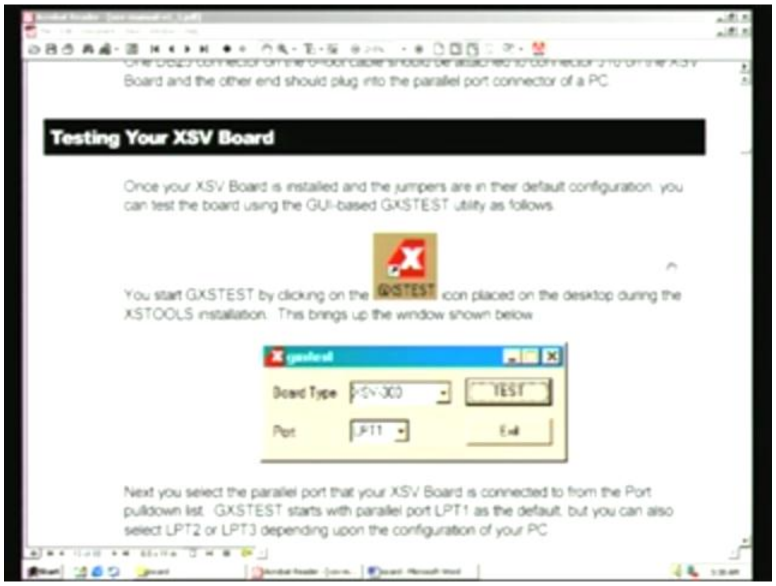
Port: LPT1 | Exit

Next you select the parallel port that your XSV Board is connected to from the Port pulldown list. GXSTEST starts with parallel port LPT1 as the default, but you can also select LPT2 or LPT3 depending upon the configuration of your PC.

This is the icon that was clicked earlier. This will actually configure the system, which we have already seen. If you click here, it will open out a list of boards. Over a dozen boards are available. For example, XSV800 is the last item. 50, 100, 300 and many other boards are available. Once you have selected the desired one, for example, XSV800, you have to do…. What is shown here is 300. We have to select XSV800 in the way I have already shown.

Similarly, suppose you have connected to the host some other port connection. For example, instead of LPT1, had you connected LPT2 or LPT3, then you have to once again select by this drop-down menu. After this, you have to just click on TEST. Immediately, the test sequence will be followed. I am not going to do this because this is a regular thing, which you can very easily do yourself. This is what you have to do. You have to merely click on TEST. First select the board, then select the port and then click on TEST. If the test is not passed, it will tell you the reasons for not passing and you will be led step by step in order to correct. If you cannot correct through all those steps, you will have to get in touch with the manufacturer. Once you are done with the test, you can exit by pressing this button. For example, I can exit here.

(Refer Slide Time: 34:26)



Board and the other end should plug into the parallel port connector of a PC

Testing Your XSV Board

Once your XSV Board is installed and the jumpers are in their default configuration, you can test the board using the GUI-based GXSTEST utility as follows.

You start GXSTEST by clicking on the GXSTEST icon placed on the desktop during the XSTOOLS installation. This brings up the window shown below.

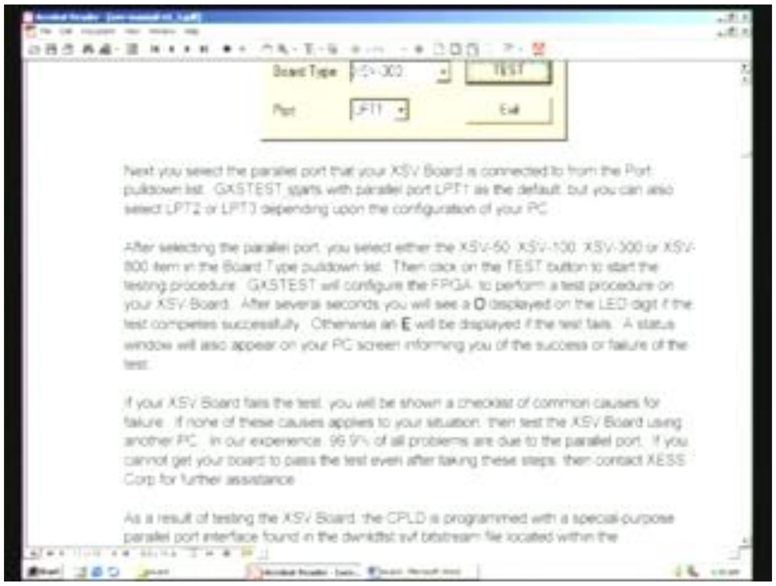
Board Type: XSV-300

Port: LPT1

Next you select the parallel port that your XSV Board is connected to from the Port pulldown list. GXSTEST starts with parallel port LPT1 as the default, but you can also select LPT2 or LPT3 depending upon the configuration of your PC.

I will read out what I have said in case you find it difficult to read it yourself. Once your XSV board is installed and the jumpers are in their default configuration, you will be told what jumpers to install – the sequence will also be told step by step and failure, if any, will also be reported. It will help you to the maximum extent possible. You can test the board using the GUI-based GXSTEST utility as follows. You start GXSTEST by clicking on the icon that we have already seen placed on the desktop during the XESS tools installation. You only need to do this only during the installation time – you do not need this for subsequent use. So also the clock – if you are not going to change the clock, which we will be seeing next. By clicking on this, you get this, which we have already seen.

(Refer Slide Time: 35:27)



Next you select the parallel port that your XSV Board is connected to from the Port pulldown list. GXSTEST starts with parallel port LPT1 as the default, but you can also select LPT2 or LPT3 depending upon the configuration of your PC.

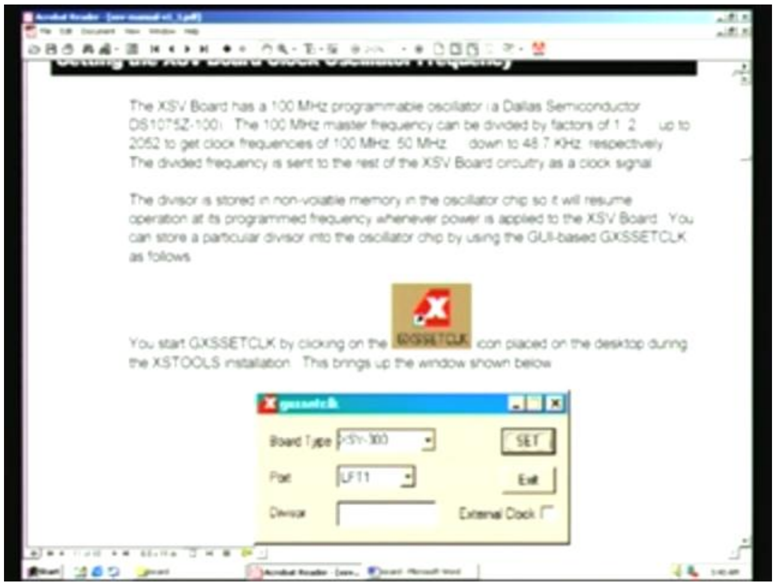
After selecting the parallel port, you select either the XSV-50, XSV-100, XSV-300 or XSV-800 item in the Board Type pulldown list. Then click on the TEST button to start the testing procedure. GXSTEST will configure the FPGA to perform a test procedure on your XSV Board. After several seconds you will see a **O** displayed on the LED digit if the test completes successfully. Otherwise an **E** will be displayed if the test fails. A status window will also appear on your PC screen informing you of the success or failure of the test.

If your XSV Board fails the test, you will be shown a checklist of common causes for failure. If none of these causes applies to your situation, then test the XSV Board using another PC. In our experience, 99.9% of all problems are due to the parallel port. If you cannot get your board to pass the test even after taking these steps, then contact XESS Corp for further assistance.

As a result of testing the XSV Board, the CPLD is programmed with a special-purpose parallel port interface found in the dwnldtst.svf bitstream file located within the

Next, you select the parallel port that your XSV board is connected to from the port. We have seen this also. Then GXSTEST starts with the parallel port LPT1 as the default, but you can also select LPT2 or LPT3, depending upon the configuration of your PC. This is for different boards that we have already seen. I am not going to read out everything – I am going to only read the salient points. GXSTEST will configure the FPGA to perform a test procedure on your XSV board. This is what TEST is for. After several seconds, you will see a 0 displayed on the LED digit if the test completes successfully. If it is not successful, E will be displayed and a status window will also appear on your PC screen informing you of the success or failure of the test. This is what we have already seen. If your XSV board fails the test, you will be shown a checklist of common causes for failure. If none of these causes applies to your situation, then test the XSV board using another PC. This describes their experience. Otherwise, you have to get in touch with the manufacturer.

(Refer Slide Time: 36:42)



The XSV Board has a 100 MHz programmable oscillator (a Dallas Semiconductor DS1075Z-100). The 100 MHz master frequency can be divided by factors of 1, 2, ... up to 2052 to get clock frequencies of 100 MHz, 50 MHz, ... down to 48.7 KHz, respectively. The divided frequency is sent to the rest of the XSV Board circuitry as a clock signal.

The divisor is stored in non-volatile memory in the oscillator chip so it will resume operation at its programmed frequency whenever power is applied to the XSV Board. You can store a particular divisor into the oscillator chip by using the GUI-based GXSSETCLK as follows.

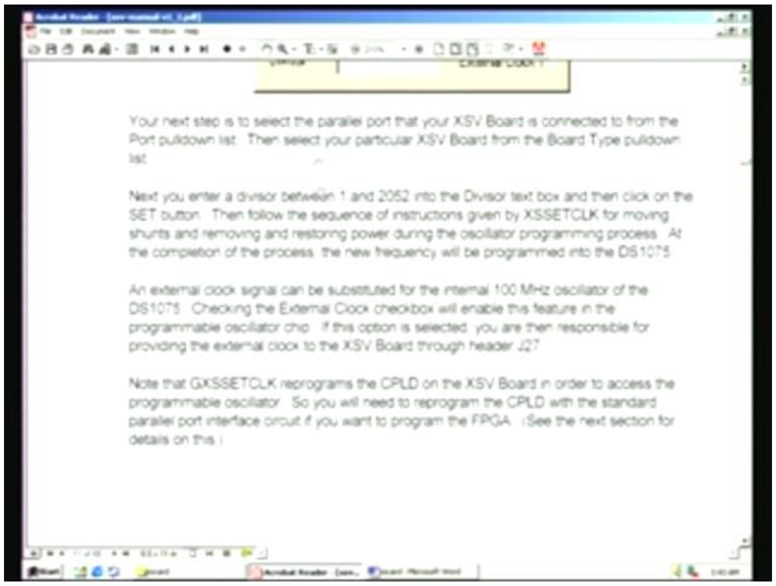
You start GXSSETCLK by clicking on the GXSSETCLK icon placed on the desktop during the XSTOOLS installation. This brings up the window shown below.

Next is to program the clock. You have to do the setting in order to program the clock. For example, as a default, you can run your FPGA-based design at 100 Megahertz. There will be a divisor, which we will see shortly. If that divisor is 1, you will get 100 Megahertz of the operating frequency. Suppose you put the divisor at 2, you will get 50 Megahertz and so on – you can operate at different frequencies. For example, the XSV board has a 100 Megahertz programmable oscillator and the chip used for this purpose is also mentioned.

The master frequency can be divided by factors 1, 2, right up to 2052. This is what you can put in the menu that is going to come here. Corresponding to 2052, you will get 48.7 Kilohertz. This will be the operating frequency of your actual board and the divided frequency is sent to the rest of the board circuitry as a clock signal. You can say that this is the master clock after the division is incorporated. The divisor is stored in non-volatile memory in the oscillator chip and so, it will resume operation at the programmed frequency whenever power is applied to the XSV board. You can store a particular divisor into the oscillator chip by using this clock that we see as an icon here.

You can click on this in order to start. By clicking on this clock icon on the desktop during the XSTOOLS installation, out comes the pop-up window, which is similar to what we have already seen for test except that it is for clock. Once again, you have to select your board type. You have to select 800 and LPT1, which is already there. Here, you have to key in 1 or 2 depending upon how many Megahertz of operation you want. For example, if you put 2 (I think we are going to use divisor 2), that will give corresponding 50 Megahertz operation. Once this is done, you can just click on SET and finally, you can exit. Suppose you do not want the clock generated in this fashion, you can also feed an external clock by putting this tick. The external clock will be taken, in which case you have to also install the jumper right on the board, which you can get from the manual. We are not going to use this feature. We are going to use this divisor and we are going to put 2 for our traffic controller application, which we are going to take up next.

(Refer Slide Time: 39:30)



Your next step is to select the parallel port that your XSV Board is connected to from the Port pulldown list. Then select your particular XSV Board from the Board Type pulldown list.

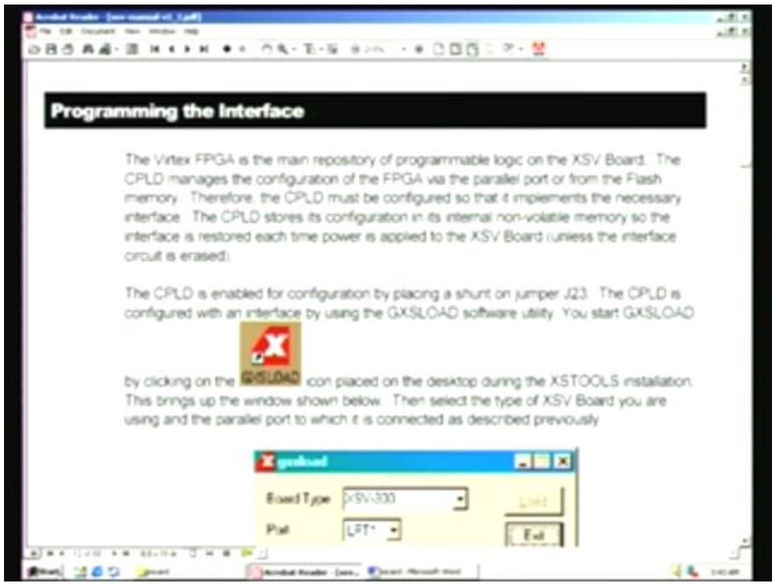
Next you enter a divisor between 1 and 2052 into the Divisor text box and then click on the SET button. Then follow the sequence of instructions given by XSSETCLK for moving shunts and removing and restoring power during the oscillator programming process. At the completion of the process, the new frequency will be programmed into the DS1075.

An external clock signal can be substituted for the internal 100 MHz oscillator of the DS1075. Checking the External Clock checkbox will enable this feature in the programmable oscillator chip. If this option is selected, you are then responsible for providing the external clock to the XSV Board through header J27.

Note that GXSSETCLK reprograms the CPLD on the XSV Board in order to access the programmable oscillator. So you will need to reprogram the CPLD with the standard parallel port interface circuit if you want to program the FPGA. (See the next section for details on this.)

We have already described this and I do not have to go through all this.

(Refer Slide Time: 39:45)



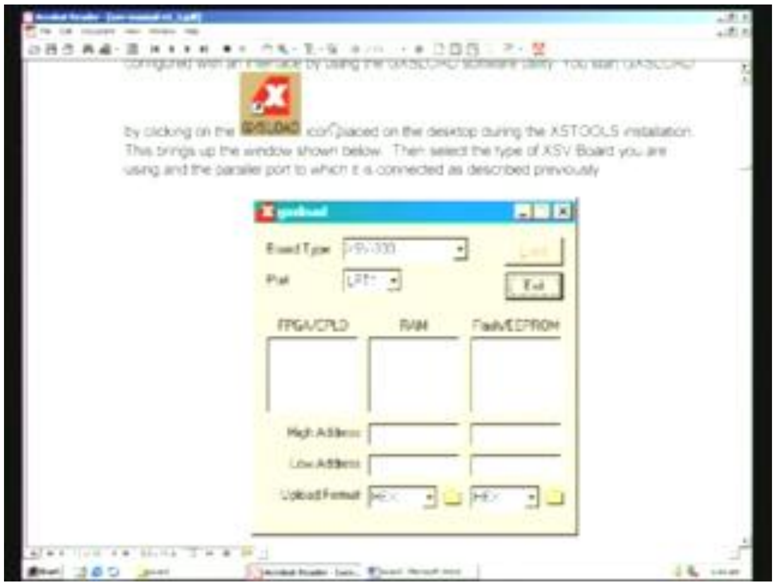
Programming the Interface

The Virtex FPGA is the main repository of programmable logic on the XSV Board. The CPLD manages the configuration of the FPGA via the parallel port or from the Flash memory. Therefore, the CPLD must be configured so that it implements the necessary interface. The CPLD stores its configuration in its internal non-volatile memory so the interface is restored each time power is applied to the XSV Board (unless the interface circuit is erased).

The CPLD is enabled for configuration by placing a shunt on jumper J23. The CPLD is configured with an interface by using the GXSLOAD software utility. You start GXSLOAD by clicking on the GXSLOAD icon placed on the desktop during the XSTOOLS installation. This brings up the window shown below. Then select the type of XSV Board you are using and the parallel port to which it is connected as described previously.

For this, the icon is here. This is clock. This is precisely what we got for the clock. If I say Exit, it will go off.

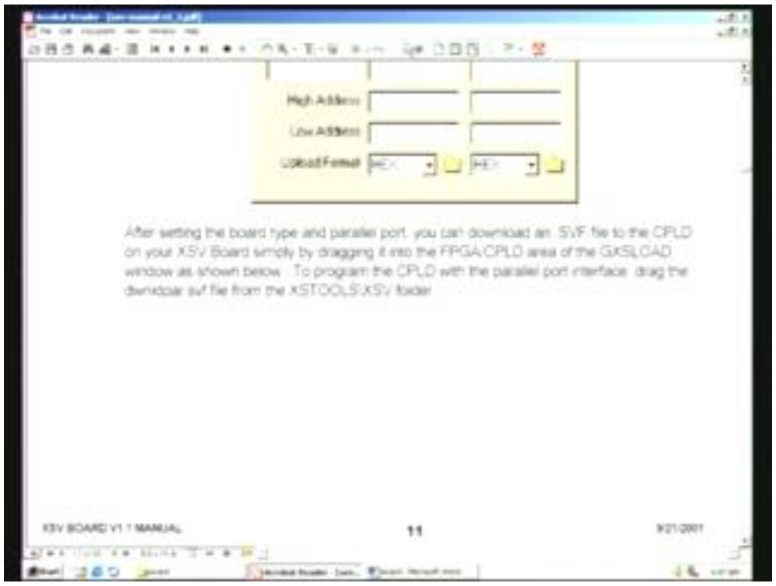
(Refer Slide Time: 40:06)


by clicking on the GXSLOAD icon placed on the desktop during the XSTOOLS installation. This brings up the window shown below. Then select the type of XSV Board you are using and the parallel port to which it is connected as described previously

Next, we have to load the bit stream – what you have already developed during the last right up to the place and route. That is what you need to download in order to download your application code. Prior to doing this, you also need to initialize CPLD because that particular CPLD is for managing the whole configuration. You can also put your codes and execute small codes. Major codes will always be residing in the FPGA Virtex-II. This particular menu is, once again, for board selection. For example, you can select XSV800 here and the port, which we have been seeing all along. This is for LPT1 because we have connected LPT1 in this case.

Here, you will see that there is one field for FPGA as well as CPLD. It is here that you have to drop the actual bit files. You need to have ready another window open for the bit files – the folder in which you have put your bit file. Just pick it from there and drop it here. After that, this Load, which is not energized right now, will get activated. Once you press the Load button, it will download the bit stream that is available here. Suppose you want to put from a file into a RAM. We have already seen that 2 megabytes of RAM is available or a flash RAM is also available – that can be put here in this. This column is for RAM and this is for the flash RAM or EEPROM. Once you say RAM or flash RAM, there will be a low-order address as well as high-order address. It is starting from this address to address, whatever is available in the file. Transfer that particular file's contents into this particular RAM, whose address from Low Address here and ends up with High Address. You can use that for this. In order to configure the bit stream, you do not require any of these features. We may not be using this. Before that, let me read what it has to say. We have to click on the GXSLOAD icon placed on the desktop during the XSTOOLS installation. This brings up the window shown below. Then, select the type of XSV board you are using and the parallel port to which it is connected as described previously.

(Refer Slide Time: 43:10)

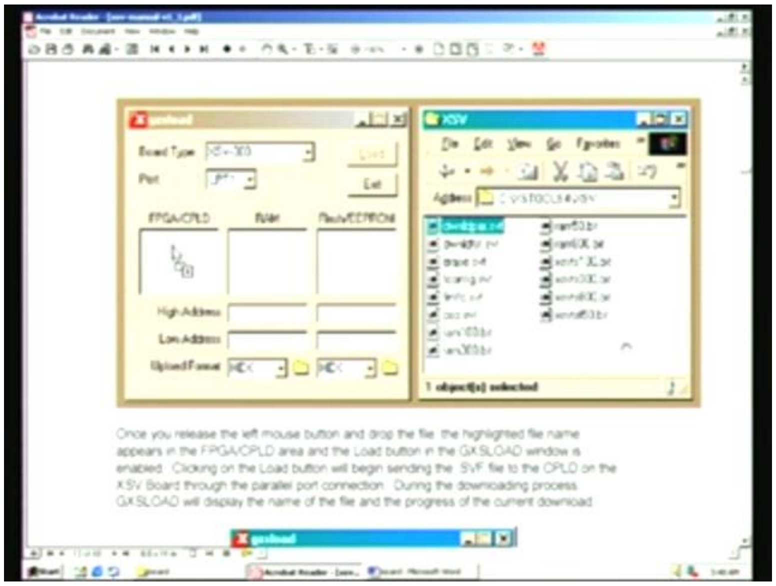


After setting the board type and parallel port, you can download an .SVF file to the CPLD on your XSV Board simply by dragging it into the FPGA/CPLD area of the GXSLOAD window as shown below. To program the CPLD with the parallel port interface, drag the dwnldpar.svf file from the XSTOOLS\XSV folder.

XSV BOARD V1.1 MANUAL 11 9/21/2001

After setting the board type and parallel port, you can download a .SVF file to the CPLD on your XSV board simply by dragging it into the FPGA/CPLD area of the GXSLOAD window as shown. To program the CPLD with the parallel port interface, drag this file. This is the file you have to drag in order to program the CPLD. This stands for download parallel and it is a .SVF extension file. It is in the XSTOOLS/XSV folder. Once you drop it in this field, Load will be energized and then, you have to press Load.

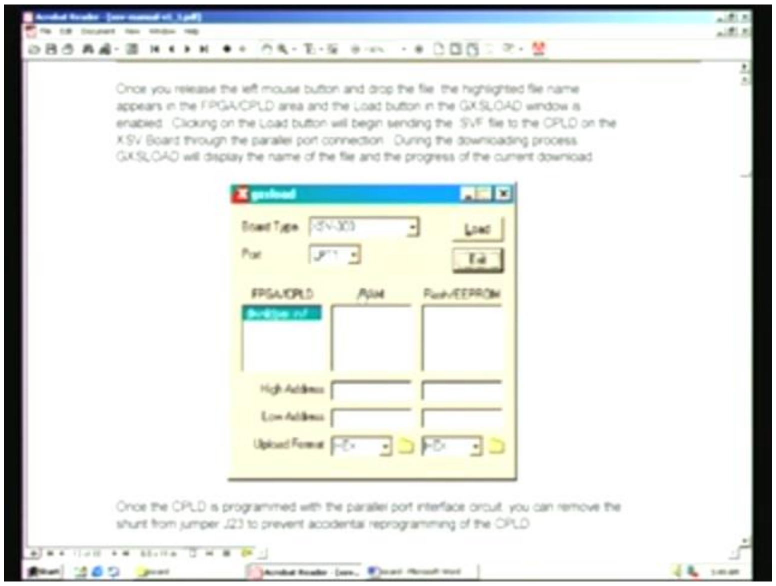
(Refer Slide Time: 44:07)



Once you release the left mouse button and drop the file, the highlighted file name appears in the FPGA/CPLD area and the Load button in the GXSLOAD window is enabled. Clicking on the Load button will begin sending the SVF file to the CPLD on the XSV Board through the parallel port connection. During the downloading process GXSLOAD will display the name of the file and the progress of the current download.

We also have another menu displayed here. You can see another window open. You go to that particular folder, for example, CXSTOOLS4/XSV and you have to pick up downloadpar (par is short form for parallel). You have to pick up that .SVF file. Just click on the mouse, do not release it and drag it there. You will see that this file is accompanying here. Drop it here when you come into this. Once you drop, that file will be listed here and the Load button will be energized. You can press Load and then, the actual downloading will take place, taking this as the source file and CPLD will be programmed. Once you release the left mouse button and drop the file, the highlighted file name appears in the FPGA/CPLD area and the Load button in the gxsload window is enabled. Clicking on the Load button will begin sending the .SVF file to the CPLD on the XSV board through the parallel port connection. During the downloading process, GXSLOAD will display the name of the file and the progress of the current download. This is what we have already seen.

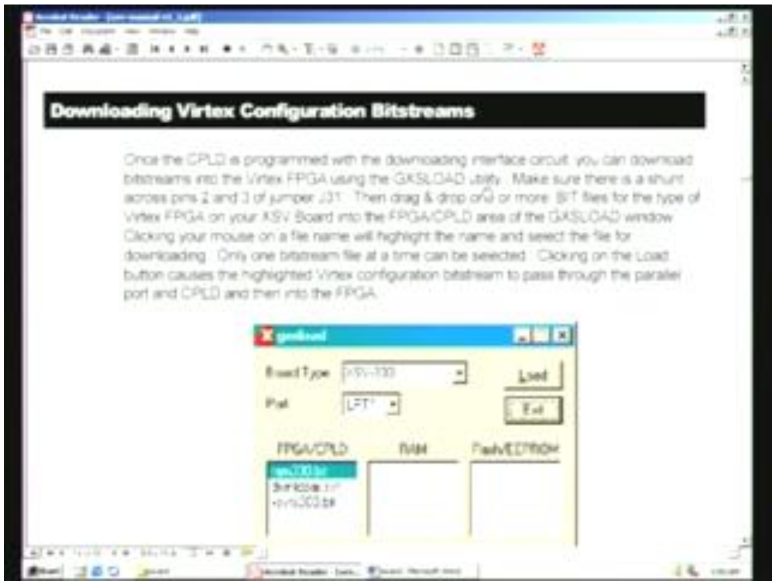
(Refer Slide Time: 45:26)


Once you release the left mouse button and drop the file, the highlighted file name appears in the FPGA/CPLD area and the Load button in the GXSLOAD window is enabled. Clicking on the Load button will begin sending the SVF file to the CPLD on the XSV Board through the parallel port connection. During the downloading process GXSLOAD will display the name of the file and the progress of the current download.

Once the CPLD is programmed with the parallel port interface circuit, you can remove the shunt from jumper J23 to prevent accidental reprogramming of the CPLD.

We have seen this already and so we will skip this. The only difference is that Load has been energized here. Once CPLD is programmed with the parallel port interface circuit, you can remove the shunt. These are all jumper settings, which you can easily follow. I am just skipping all the jumpers. You may get bored listening to all the jumper settings.

(Refer Slide Time: 45:52)

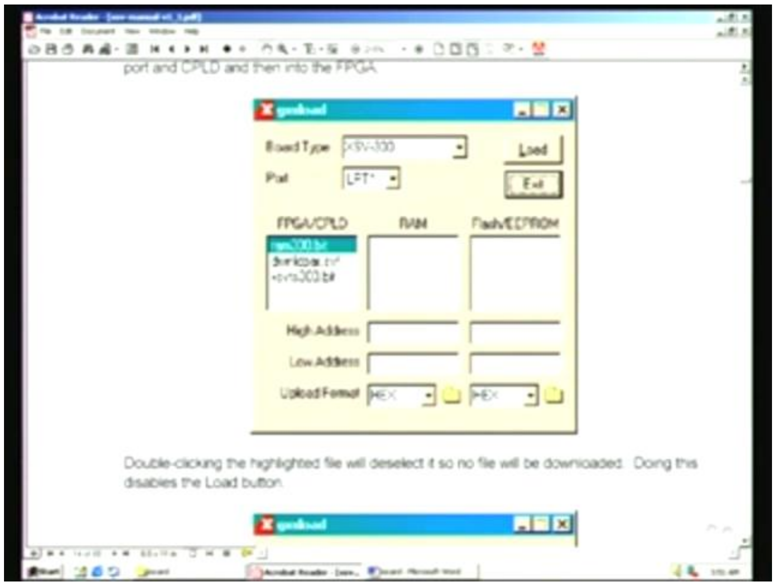

Downloading Virtex Configuration Bitstreams

Once the CPLD is programmed with the downloading interface circuit, you can download bitstreams into the Virtex FPGA using the GXSLOAD utility. Make sure there is a shunt across pins 2 and 3 of jumper J31. Then drag & drop one or more BIT files for the type of Virtex FPGA on your XSV Board into the FPGA/CPLD area of the GXSLOAD window. Clicking your mouse on a file name will highlight the name and select the file for downloading. Only one bitstream file at a time can be selected. Clicking on the Load button causes the highlighted Virtex configuration bitstream to pass through the parallel port and CPLD and then into the FPGA.

Next is the actual application file. For example, you want to have a traffic light controller. You will have a traffic light controller in your folder just as we had the previous folder for CPLD – downloadpar. Here will be listed your traffic_controller.bit. Once the CPLD is programmed with the downloading interface circuit, you can download bit streams into the Virtex FPGA using the GXSLOAD utility. Make sure there is a shunt across this – once again, a reminder that a jumper must be installed. You have to do lots of jugglery with these jumpers. Let me not get into the details. It is very easy to follow – the manual contains step-by-step procedures for all this.

Then, drag and drop one or more .bit files for the type of Virtex FPGA on your XSV board into the FPGA/CPLD area, which is this, of the GXSLOAD window. Clicking your mouse on a file name will highlight the name and select the file for downloading. By clicking this, it will highlight and this will be energized. Only one bit stream file at a time can be selected. Clicking on the Load button causes the highlighted Virtex configuration bit stream to pass through the parallel port and CPLD and then into the FPGA.

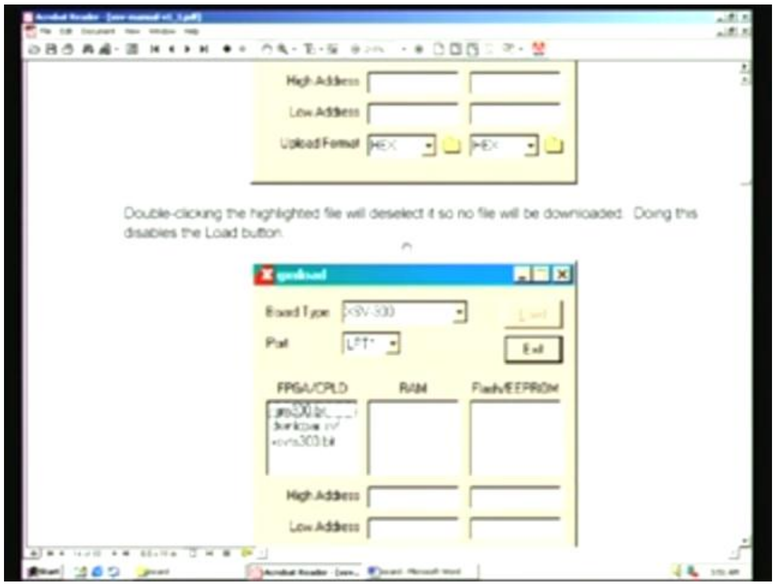
(Refer Slide Time: 47:14)


port and CPLD and then into the FPGA.

Double-clicking the highlighted file will deselect it so no file will be downloaded. Doing this disables the Load button.

Double-clicking the highlighted file will deselect. If you do not want, you can double-click. Just ignore that.

(Refer Slide Time: 47:24)


High Address

Low Address

Upload Format HEX HEX

Double-clicking the highlighted file will deselect it so no file will be downloaded. Doing this disables the Load button.

Board Type XSV-300

Port LPT1

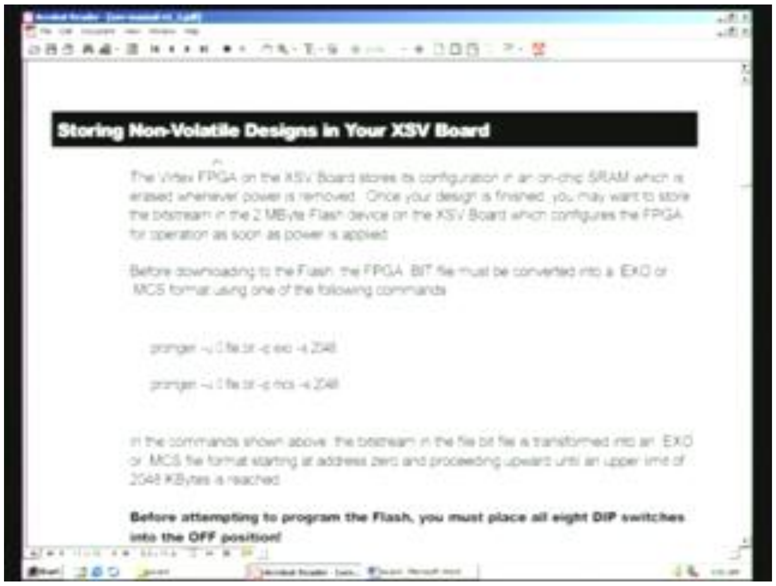
FPGA/CPLD	RAM	Flash/EEPROM
[illegible]		

High Address

Low Address

It disables the load button if you do not want. Once you have clicked Load, downloading actually starts.

(Refer Slide Time: 47:34)


Storing Non-Volatile Designs in Your XSV Board

The Virtex FPGA on the XSV Board stores its configuration in an on-chip SRAM which is erased whenever power is removed. Once your design is finished, you may want to store the bitstream in the 2 MByte Flash device on the XSV Board which configures the FPGA for operation as soon as power is applied.

Before downloading to the Flash, the FPGA .BIT file must be converted into a .EXO or .MCS format using one of the following commands:

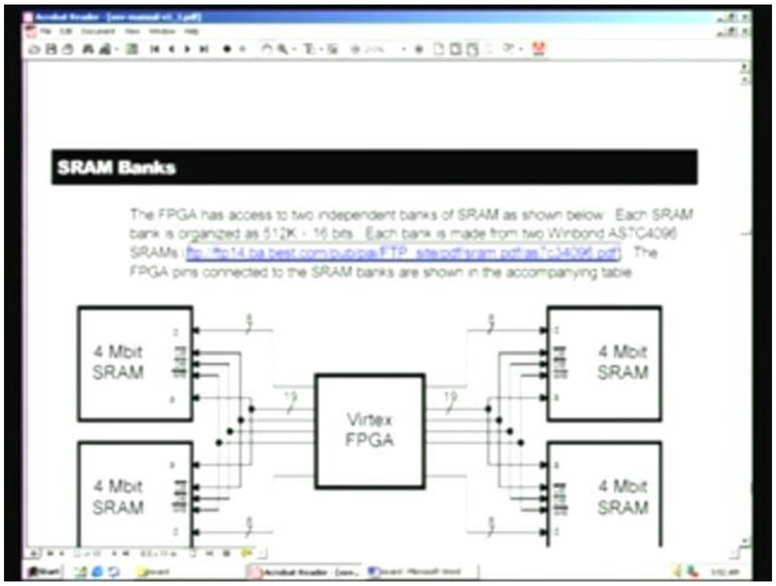
```
promgen -u 0 file.bit -p exo -s 2048
promgen -u 0 file.bit -p mcs -s 2048
```

In the commands shown above, the bitstream in the file.bit file is transformed into an .EXO or .MCS file format starting at address zero and proceeding upward until an upper limit of 2048 KBytes is reached.

Before attempting to program the Flash, you must place all eight DIP switches into the OFF position!

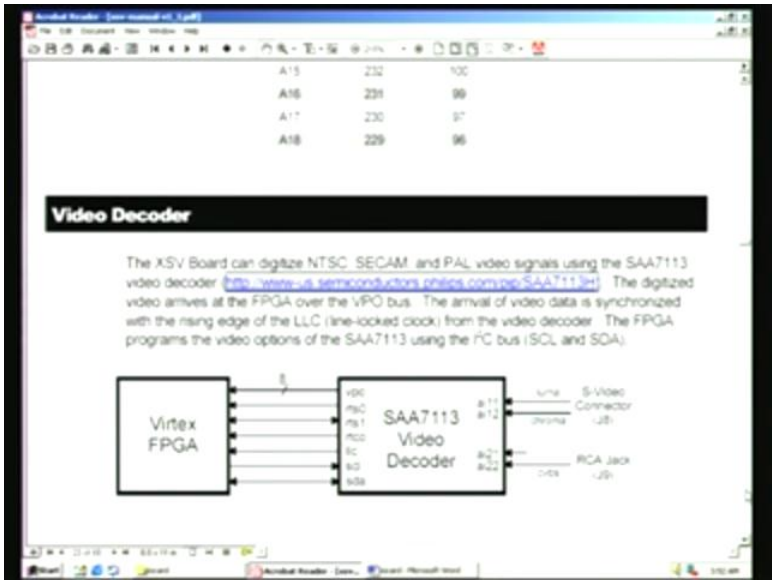
You can have non-volatile storage. We will not go into the details of this. As I said, 2 megabytes flash RAM available. We will skip all this and go to a relevant portion regarding the I/Os that we have already seen. These are all the same. Flash RAM, everything. The manual has all these interconnections.

(Refer Slide Time: 48:05)



SRAM Banks

The FPGA has access to two independent banks of SRAM as shown below. Each SRAM bank is organized as 512K × 16 bits. Each bank is made from two Winbond AS7C4096 SRAMs (ftp://ftp14.ba.best.com/pub/pai/FTP_site/pdf/sram.pdf/as7c34096.pdf). The FPGA pins connected to the SRAM banks are shown in the accompanying table.



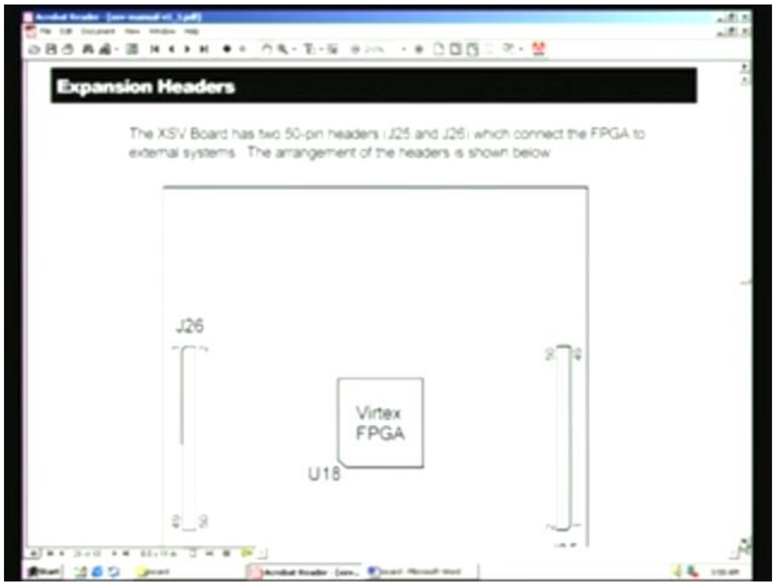
A15	232	100
A16	231	99
A17	230	97
A18	229	96

Video Decoder

The XSV Board can digitize NTSC, SECAM, and PAL video signals using the SAA7113 video decoder (http://www-us.semiconductors.philips.com/pip/SAA7113H). The digitized video arrives at the FPGA over the VPO bus. The arrival of video data is synchronized with the rising edge of the LLC (line-locked clock) from the video decoder. The FPGA programs the video options of the SAA7113 using the I²C bus (SCL and SDA).

You can see the SRAM, pin configuration, video decoder, etc. We will skip all this. We will not go into the details.

(Refer Slide Time: 48:17)

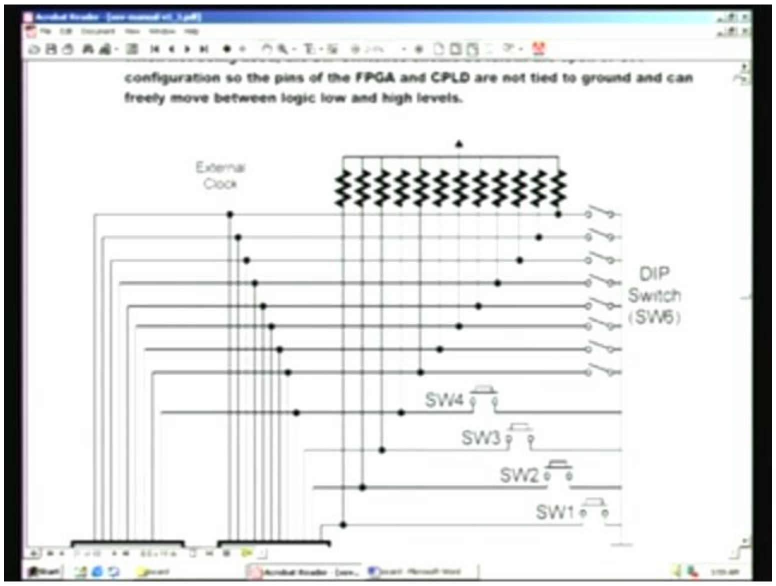


Expansion Headers

The XSV Board has two 50-pin headers (J25 and J26) which connect the FPGA to external systems. The arrangement of the headers is shown below

We have expansion headers, which we have already talked about. We are going to use this in order to connect to our extra I/O card, which we will be dealing with after the FPGA card is shown. That will be used in our first application, namely the traffic controller. For this, we need both these expansion cables connected to this expansion header. Next, we will go on to push button switches, etc.

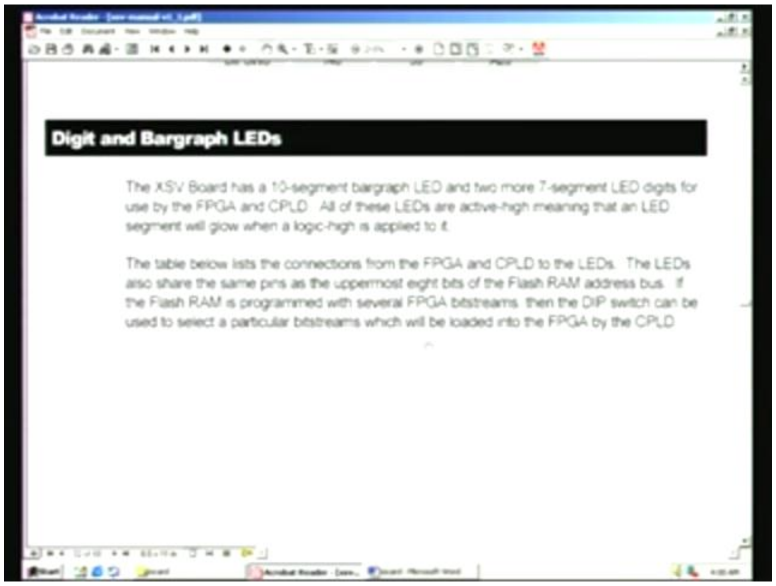
(Refer Slide Time: 49:03)



configuration so the pins of the FPGA and CPLD are not tied to ground and can freely move between logic low and high levels.

For example, this CPLD as well as FPGA are connected to the push button switches in this fashion. We have already seen the DIP switch. It is 8 bits and a single switch. They are all pulled high here as you can see. This is V_{CC} or +5 Volt. All these signals go not only to the FPGA but also to the CPLD. Even user applications can be residing in the CPLD. In FPGA, the application is much longer in size. You also have four push button switches and they are also connected in similar fashion, normally open. They are all pulled to the ground. Whenever you short, this particular I/O pin will go low. Otherwise, it will be pulled high. That is logic 1.

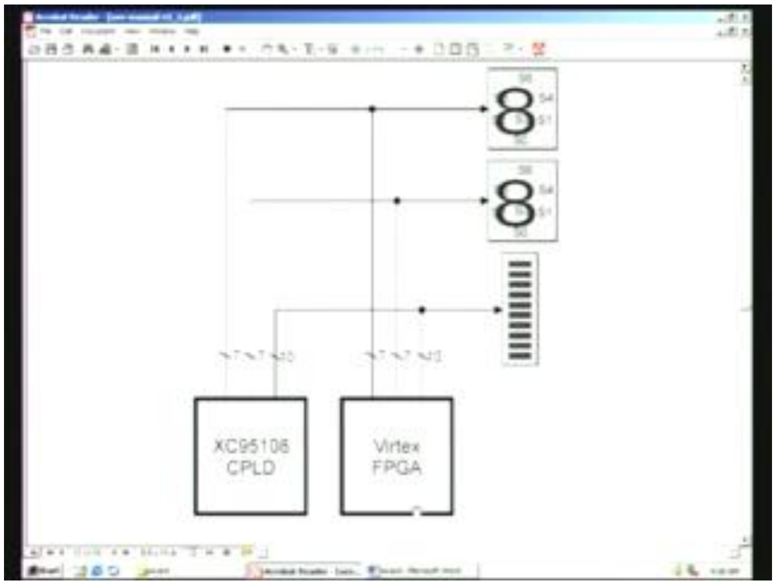
(Refer Slide Time: 49:59)



Digit and Bargraph LEDs

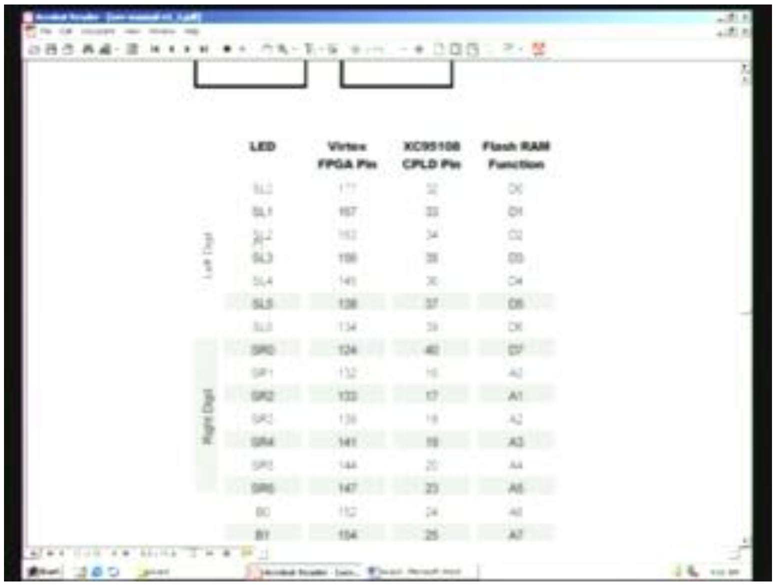
The XSV Board has a 10-segment bargraph LED and two more 7-segment LED digits for use by the FPGA and CPLD. All of these LEDs are active-high meaning that an LED segment will glow when a logic-high is applied to it.

The table below lists the connections from the FPGA and CPLD to the LEDs. The LEDs also share the same pins as the uppermost eight bits of the Flash RAM address bus. If the Flash RAM is programmed with several FPGA bitstreams, then the DIP switch can be used to select a particular bitstreams which will be loaded into the FPGA by the CPLD.

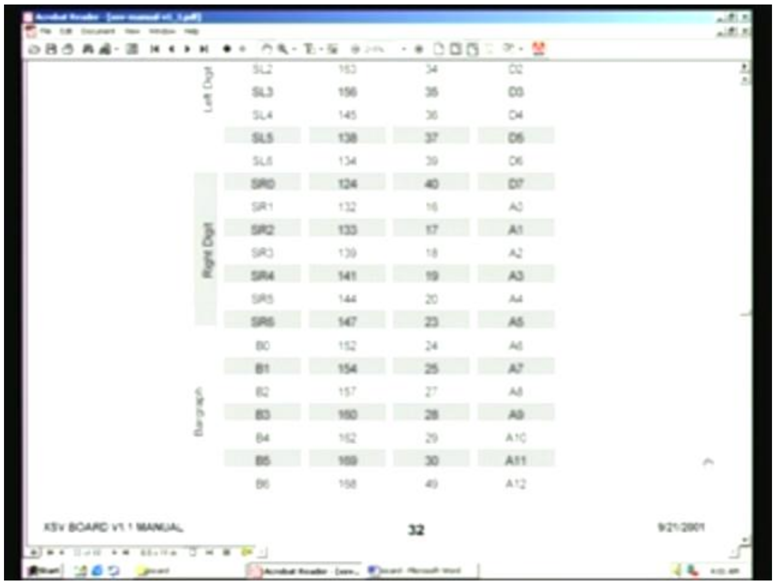


We also have seven-segment LEDs as well as a bar graph here. Once again, they are connected to the CPLD as well as FPGA. They are all in parallel so that depending upon your application, you can use either CPLD or FPGA and download your bit stream. We will use this FPGA but we will not use these LEDs for our traffic light controller but another I/O card, which we will be covering next.

(Refer Slide Time: 50:36)

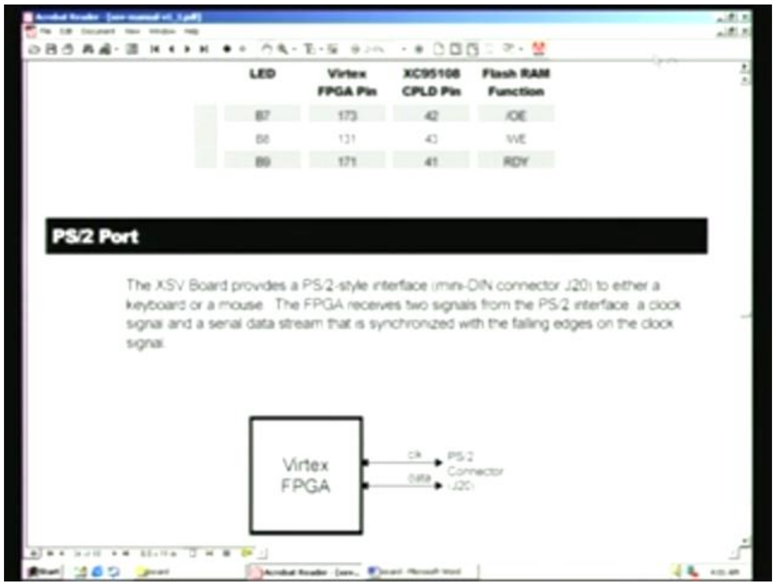


	LED	Virtex FPGA Pin	XC95108 CPLD Pin	Flash RAM Function
Left Digit	SL0	177	32	D0
	SL1	167	33	D1
	SL2	163	34	D2
	SL3	156	35	D3
	SL4	145	36	D4
	SL5	138	37	D5
	SL6	134	39	D6
Right Digit	SR0	124	40	D7
	SR1	132	16	A0
	SR2	133	17	A1
	SR3	139	18	A2
	SR4	141	19	A3
	SR5	144	20	A4
	SR6	147	23	A5
	B0	152	24	A6
	B1	154	25	A7



Left Digit	SL2	153	34	D2
	SL3	156	35	D3
	SL4	145	36	D4
	SL5	138	37	D5
	SL6	134	39	D6
Right Digit	SR0	124	40	D7
	SR1	132	16	A0
	SR2	133	17	A1
	SR3	139	18	A2
	SR4	141	19	A3
	SR5	144	20	A4
	SR6	147	23	A5
Bargraph	B0	152	24	A6
	B1	154	25	A7
	B2	157	27	A8
	B3	160	28	A9
	B4	162	29	A10
	B5	169	30	A11
	B6	158	49	A12

XSV BOARD V1.1 MANUAL 32 9/21/2001



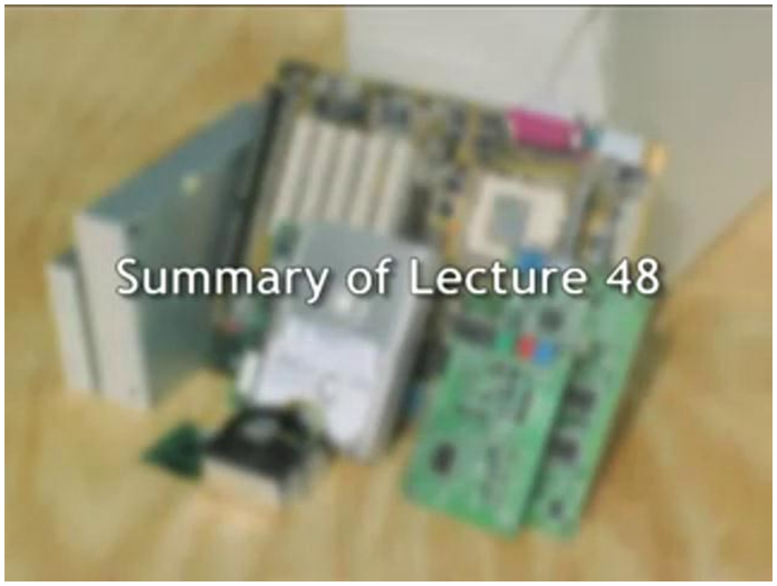
LED	Virtex FPGA Pin	XC95108 CPLD Pin	Flash RAM Function
B7	173	42	/OE
B8	131	43	/WE
B9	171	41	RDY

PS/2 Port

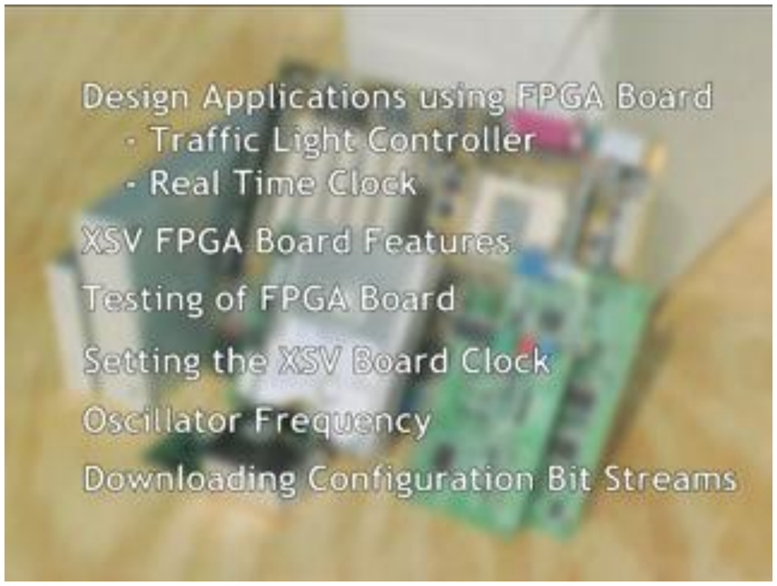
The XSV Board provides a PS/2-style interface (mini-DIN connector J20) to either a keyboard or a mouse. The FPGA receives two signals from the PS/2 interface: a clock signal and a serial data stream that is synchronized with the falling edges on the clock signal.

Some of these LED connections are all shown here. The actual Virtex FPGA pin numbers are all shown here. There are two LEDs: one is left and the other is right. They go under SL0 through 6 for seven segments and there is no decimal point. This is rather a handicap where a decimal point is required. This should be sufficient in order to tackle the FPGA board. In the next class, we will see what this I/O card looks like. We will go into the details and finally, show a demo using this FPGA card as well as I/O card for the traffic controller that we have designed before.

(Refer Slide Time: 51:37)



Summary of Lecture 48



Design Applications using FPGA Board
- Traffic Light Controller
- Real Time Clock

XSV FPGA Board Features

Testing of FPGA Board

Setting the XSV Board Clock

Oscillator Frequency

Downloading Configuration Bit Streams