An Implementation of I^2C using VHDL for DATA surveillance

Arvind Sahu[1]
M.Tech NIIST BHOPAL
sahuarvind28@gmail.com

Dr. Ravi Shankar Mishra[2]
HOD (EC) NIIST BHOPAL
ravishankarmishra@rediffmail.com

Puran Gour[3]
ASST.Prof.NIIST Bhopal
purangour@rediffmail.com

***Abstract* - Once the I^2C protocol designed for data surveillance system, It is easy to communicate devices with each other without data loss, as well as a excellent speed is being developed with I^2C as compare to any other communication methodology. The main purpose of this technique is not only to communicate the devices but keep in touch with every operation which can be performing by with the help of this protocol. Because of I^2C only uses two wires for communication. It is lightweight, economical, and omnipresent. The design is use in surveillance system to make overall system more efficient and accurate, with the use of I^2C data transmit rate get also increased. This design method was design in VHDL, implementation in FPGA and applied in OV7620 single-chip CMOS VGA color digital camera.**

Key words-I^2C protocol, Data surveillance, FPGA, VHDL SDA, , SCL, OV7620

1. INTRODUCTION

The I^2C bus physically consists of 2 active wires and a ground connection. The active wires, called SDA and SCL, are both bi-directional. SDA is the Serial Data line, and SCL is the Serial Clock line. Every device hooked up to the bus has its own unique address, no matter whether it is an MCU, LCD driver, memory, or ASIC. Each of these chips can act as a receiver and/or transmitter, depending on the functionality. Obviously, an LCD driver is only a receiver, while a memory or I/O chip can be both transmitter and receiver. The I^2C bus is a multi-master bus. This means that more than one IC capable of initiating a data transfer can be connected to it. The I^2C protocol specification states that the IC that initiates a data transfer on the bus is considered the Bus Master. Consequently, at that time, all the other ICs are regarded to be Bus Receivers .Let us take a look at a general 'inter-IC chat' on the bus. Let us consider the following setup and assume the FPGA wants to send data to one of its receivers.

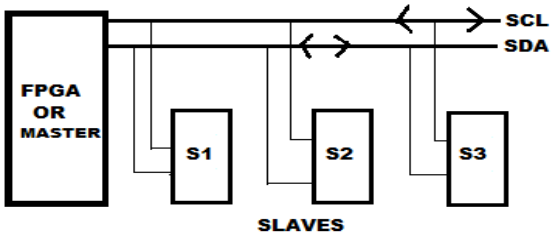


Figure 1 Block diagram of I^2C

Figure 1 shows First, the FPGA will issue a START condition. This acts as an 'Attention' signal to all of the connected devices. All ICs on the bus will listen to the bus for incoming data. Then the FPGA sends the ADDRESS of the device it wants to access, along with an indication whether the access is a Read or Write operation having received the address, all IC's will compare it with their own address. If it doesn't match, they simply wait until the bus is released by the stop condition If the address matches, however, the chip will produce a response called the ACKNOWLEDGE signal. Once the FPGA receives the acknowledge signal, it can start transmitting or receiving DATA. In this case, the FPGA will transmit data. When all is done, the FPGA will issue the STOP condition. This is a signal that the bus has been released and that the connected ICs may expect another transmission to start any moment. We have several states on the bus in this example: START, ADDRESS, ACKNOWLEDGE, DATA, STOP. These are all unique conditions on the bus.

2. I²C Bus Events: The START and STOP conditions

Prior to any transaction on the bus, a START condition needs to be issued on the bus. The start condition acts as a signal to all connected IC's that something is about to be transmitted on the bus. As a result, all connected chips will listen to the bus. After a message has been completed, a STOP condition is sent. This is the signal for all devices on the bus that the bus is available again (idle). If a chip was accessed and has received data during the last transaction, it will now process this information (if not already processed during the reception of the message).

Figure 2. START and STOP conditions

START-The chip issuing the Start condition first pulls the SDA (data) line low, and next pulls the SCL (clock) line low.
STOP-The Bus Master first releases the SCL and then the SDA line.

Figure 2 show that. A single message can contain multiple Start conditions. The use of this so-called “repeated start" is common in I²C.A Stop condition ALWAYS denotes the END of a transmission. Even if it is issued in middle of a transaction or in the middle of a byte. It is "good behavior" for a chip that, in this case, it disregards the information sent and resumes the "listening state", waiting for a new start condition.

2.1- I²C Bus Events: Transmitting a byte to a receiver

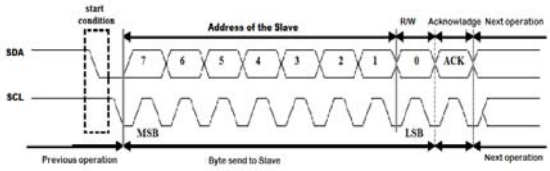


Figure 3. Transmitting a byte to a receiver

Figure 3 shows that receiver on the bus (address) and will select the mode of operation. The meaning of all following bytes depends on the receiver. A number of addresses have been reserved for special purposes. One of these addresses is reserved for the "Extended Addressing Mode". As the I²C bus gained popularity, it was soon discovered that the number of available addresses was too small. Therefore, one of the reserved addresses has been allocated to a new task to switch to 10-bit address is mode. If a standard receiver (not able to resolve extended addressing) receives this address, it won't do anything (since it's not its address).If there are receivers on the bus that can operate in the extended 10-bit addressing mode, they will ALL respond to the ACK cycle issued by the master. The second byte that gets transmitted by the master will then be taken in and evaluated against their address. Even in 10-bit extended addressing mode, Bit 0 of the first byte after the Start condition determines the receiver access mode ('1' = read / '0' = write).

2.2- I²C Bus Events: Getting ACK from a receiver

When an address or data byte has been transmitted onto the bus then this must be ACKNOWLEDGED by the receiver(s). In case of an address: If the address matches its own then that receiver and only that receiver will respond to the address with an ACK. In case of a byte, transmitted to an already addressed receiver then that receiver will respond with an ACK as well. The receiver that is going to give an ACK pulls the SDA line low immediately after reception of the 8th bit transmitted, or, in case of an address byte, immediately after evaluation of its address. In practical applications this will not be noticeable. This means that as soon as the master pulls SCL low to complete the transmission of the bit

- SDA will be pulled low by the receiver
- The master now issues a clock pulse on the SCL line
- The receiver will release the SDA line upon completion of this clock pulse
- The bus is now available again for the master to continue sending data or to generate a stop condition.

In case of data being written to a receiver, this cycle must be completed before a stop condition can be generated. The receiver will be blocking the bus (SDA kept low by receiver) until the master has generated a clock pulse on the SCL line.

2.3 I^2C Bus Events: Receiving a byte from a receiver

Once the receiver has been addressed and the receiver has acknowledged this, a byte can be received from the receiver if the R/W bit in the address was set to READ (set to '1'). The protocol syntax is the same as in transmitting a byte to a receiver, except that now the master is not allowed to touch the SDA line. Prior to sending the 8 clock pulses needed to clock in a byte on the SCL line, the master releases the SDA line. The receivers will now take control of this line. The line will then go high if it wants to transmit a '1' or, if the receiver wants to send a '0', remain low.

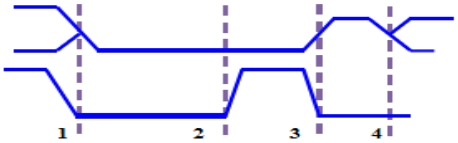


Figure 4. Receiving a byte from a receiver

Figure 4 shows that All the master has to do is generate a rising edge on the SCL line (2), read the level on SDA (3) and generate a falling edge on the SCL line (4). The receiver will not change the data during the time that SCL is high. (Otherwise a Start or Stop might inadvertently be generated.) During (1), the receiver may change the state of the SDA line. In total, this sequence has to be performed 8 times to complete the data byte. Bytes are always transmitted MSB first.The meaning of all bytes being read depends on the receiver. There is no such thing as a “universal status register". You need to consult the data sheet of the receiver being addressed to know the meaning of each bit in any byte transmitted.

3. Surveillance using I^2C

Data surveillance, also known as data surveillance, compiles personal information from various sources to investigate or monitor people’s daily activities and interactions amongst each other. Data surveillance has proven to be far cheaper and more effective than traditional forms of surveillance with I^2C and is quickly increasing in popularity throughout a number of different fields. However, the next generation of data surveillance systems will replace these components with newer digital LAN cameras, complex image processing, and data-over-IP routing. They will no longer be simply camera systems, but also data communication system.

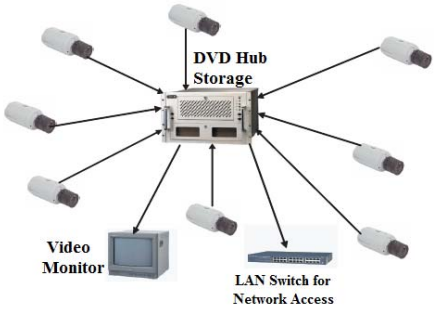


Figure 5. Data Surveillance System

3.1 DATA Compression and Image Processing

There are many different standards for data data compression, with the most popular including JPEG, H.263, Motion JPEG, MPEG, and Wavelet. The type of compression used has an impact on hardware system requirements, including memory, data rate, and storage space. However, next-generation surveillance systems will probably use the H.264 standard due to its compression efficiency. Efficiency is a key factor in the transmission of high-quality data over a bandwidth-limited network. For example, a color transmission at 30 fps at 640 x 480 pixels requires a data rate of 26Mbytes/sec. This data rate must be reduced (compressed) to a more manageable data rate that can be routed over a twisted pair of copper wires. constant bit rate (CBR) and variable bit rate (VBR).

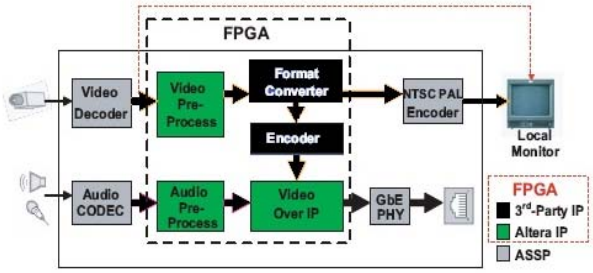


Figure 6. Surveillance camera system with FPGA

Figure 6 show that a surveillance system CBR limits the data rate for a real-time communication channel with limited bandwidth. However, when CBR compresses high-motion details, image quality is lost and results in image blocks on the display. VBR allows the data rate to adaptor the motion or absence of motion, which is extremely useful for data surveillance system storage. H.264compression with VBR provides the best efficiency for security data storage. Pre- and post processing techniques, such as de-interlacing, scaling, noise reduction using 2D filtering, and color space conversion, are also critical parts of a data surveillance system. These functions are included in Data and Image Processing.

3.2 High Performance

Performance not only applies to compression, but also pre- and post processing functions. In fact, in many cases these functions consume more performance than the compression algorithm itself. Examples of these functions include scaling, de interlacing, filtering, and color space conversion. For data surveillance, the need for high performance rules out processor-only architectures. They simply cannot meet the performance requirements with a single device. A state-of-the-art DSP running at 1 GHz cannot perform H.264 HD decoding or H.264 HD encoding, which is about ten times more complex than decoding. FPGAs are the only programmable solutions able to tackle this problem. In some cases, the best solution is a combination of an FPGA plus an external DSP processor.

4. Interfacing and result verification inModelSim-

The overall coding part can be write on VHDL and simulate on ModelSim It can be represented with the help of RTL diagram of I^2C and verified. There are many different techniques to interface FPGA with **OV7620** single-chip CMOS VGA, **CAMERA** in place of normal communication I^2C interfacing technique is implemented and result can be verified with ModelSim,

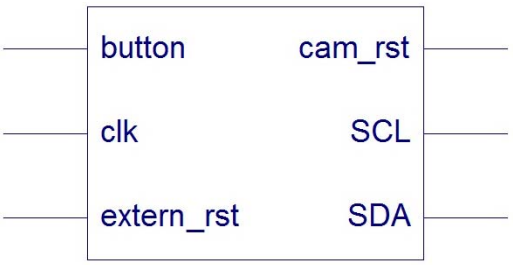


Figure 7. RTL diagram

4.1 Description of Transition States-

The transition is start from clk line, the clock is around 100 ps will be provided to the system, The position of the butten is set to 1 by forcing the value and the position of the external_rst will be 1,the transition being started but it is blank transition. The proper transition started only when the value of external_rst is forced by 0 while forcing the value suddenly the SDA line goes High to Low means 1 to 0 after a few instant SCL line also goes High to Low means 1 to 0.indicate that FPGA start sending the START signal to all interfaces Slaves. When this start signal get collected by Slaves, the start sending the ACK signal. This ACK signal will be received by the FPGA, the actual transmission will be start .The communication is serial communication technique and value can be change for every rising edge of the SCL line.

4.1.1.Ist Stage Transition – The transition is start from clk line, the clock is around 100 ps will be provided to the system, The position of the butten is set to 1 by forcing the value and the position of the external_rst will be 1,the transition being started but it is blank transition. The proper transition started only when the value of external_rst is forced by 0 while forcing the value suddenly the the SDA line goes High to Low means 1 to 0 after a few instant SCL line also goes High to Low means 1 to 0.indicate that FPGA start sending the START signal to all interfaces Slaves . When this start signal get collected by Slaves, the start sending the ACK signal. This ACK signal will be received by the FPGA, the actual transmission will be start .The communication is serial communication technique and value can be change for every rising edge of the SCL line. The ACK received by FPGA,a Slave start sending device Id i.e. 40H[0100 0000] every bit of the device Id interchange serially one by one and the states will be change for every rising edge like [Low-High-Low-Low-Low-Low-Low-Low]. After the device Id being send by Slaves and received by FPGA receiver sends the ACK signal after receiving the ACK signal the Address of the register is ready for sending i.e 15 [0100 0101] every bit of the device Id interchange serially one by one and the states will be change for every rising edge like [Low- High-Low-Low-Low-High-Low-High].When the Register is properly build up to the FPGA, receiver again send the ACK signal after this Slave start sending data i.e 00[0000 0000] every bit of the device Id interchange serially one by one and the states will be change for every rising edge like[Low-Low-Low-Low-Low-Low-Low-Low] The data of the Register decide the parameter which want to controlled but the device Id and the Address of the Register is fixed.

When the device the Register is

Register 01 - rw: PS gain control

Device Id- 40 $(0100,0000)_2$

Location - 15 $(0001, 0101)_2$

Data - 00$(0000, 0000)_2$

PS<6:0> - white balance value for the PS channel.

The formula is:

PS_gain=1+ (PS<7:0> - [80])/[100]; range (0.5x ~ 1.5x).

PS<7> - Sign bit. If “1”, PS channel gain increase; “0” gain decrease.

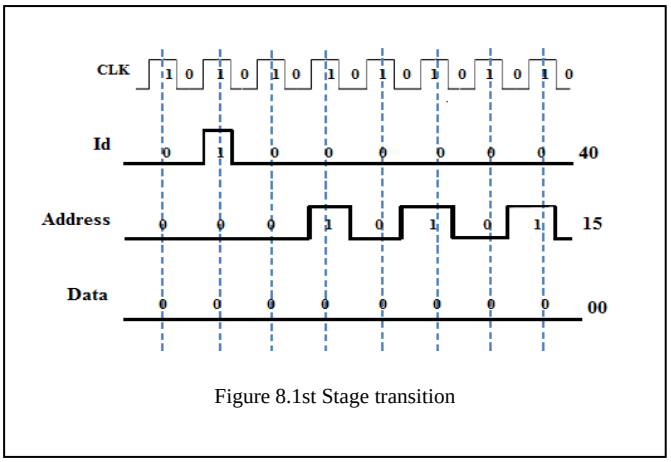


Figure 8.1st Stage transition

4.1.2.IInd Stage Transition – When the stop signal is being received by all connecting slaves. Master and Slave are ready for net state transition. The transition get start from SDA line, SDA line may change its state from High to Low means 1 to 0 after few peco second SCL line also change its state from High to Low. It is start signal to all connecting slaves, a Slave who want to be interface send the ACK signal. The ACK received by FPGA,a Slave start sending device Id i.e. 40H[0100 0000] every bit of the device Id interchange serially one by one and the states will be change for every rising edge like [Low-High-Low-Low-Low-Low-Low-Low]. After the device Id being send by Slaves and received by FPGA receiver sends the ACK signal after receiving the ACK signal the Address of the register is ready for sending i.e. 17 [0001 0111] every bit of the device Id interchange serially one by one and the states will be change for every rising edge like [Low- Low-Low-High-Low-High-High-High]. When the Register is properly build up to the FPGA, receiver again send the ACK signal after this Slave start sending data i.e. 41[0100 0001] every bit of the device Id interchange serially one by one and the states will be change for every rising edge like[Low-HighLow-Low-Low-Low-High-Low-Low].The data of the Register decide the parameter which want to controlled but the device Id and the Address of the Register is fixed.

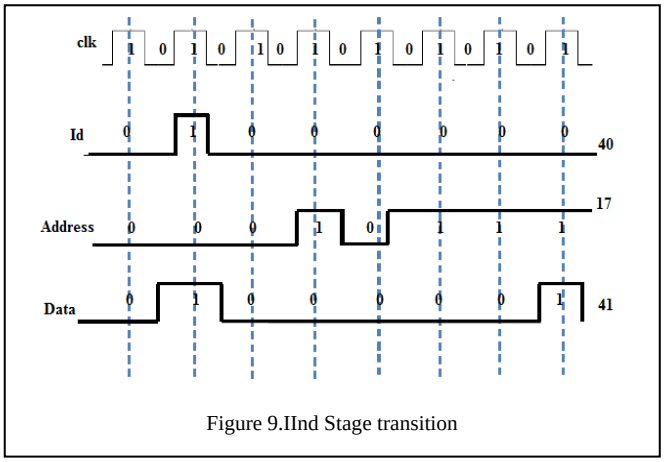


Figure 9.IInd Stage transition

When the device the Register is
Register 02 – rw: PSuration control
Location 17(0001, 0111)2
Data 041(0100, 0001)2
PS<7:0> - PSuration adjustment for the UV channel based on the default setting;
Range (-4dB ~ +6dB). If PS<7:0> >[80], increase; if PS<7:0> < [80], decrease.

4.1.3.IIIrd Stage Transition- When the second transition is finish properly the third transition is get started after getting stop pulse from the Master. Master and Slave are ready for net state transition. The transition get start from SDA line, SDA line may change its state from High to Low means 1 to 0 after few psecond SCL line also change its state from High to Low. It is start signal to all connecting slaves, a Slave who want to be interface send the ACK signal. The ACK received by FPGA,a Slave start sending device Id i.e. 40H[0100,0000] every bit of the device Id interchange serially one by one and the states will be change for every rising edge like [Low-High-Low-Low-Low-Low-Low-Low]. After the device Id being send by Slaves and received by FPGA receiver sends the ACK signal after receiving the ACK signal the Address of the register is ready for sending i.e. 3A [0011,1010] every bit of the device Id interchange serially one by one and the states will be change for every rising edge like [Low-Low-High-High-High-Low-High-Low]. When the Register is properly build up to the FPGA, receiver again send the ACK signal after this Slave start sending data i.e. 16[0001,0110] every bit of the device Id interchange serially one by one and the states will be change for every rising edge like [Low-Low-Low- High-Low-High-High Low].The data of the Register decide the parameter which want to controlled but the device Id and the Address of the Register is fixed.
When the device the Register is
Register 03 - rw: Analog Sharpness control
Device Id- 40(0100,0000)2
Location 3A(0011,1010)2
Data 16(0001, 0110)2
SHP<7:4> - Sharpness Threshold.
SHP<3:0> - Sharpness Magnitude.

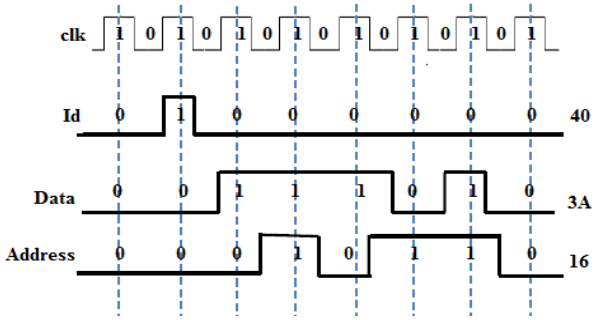
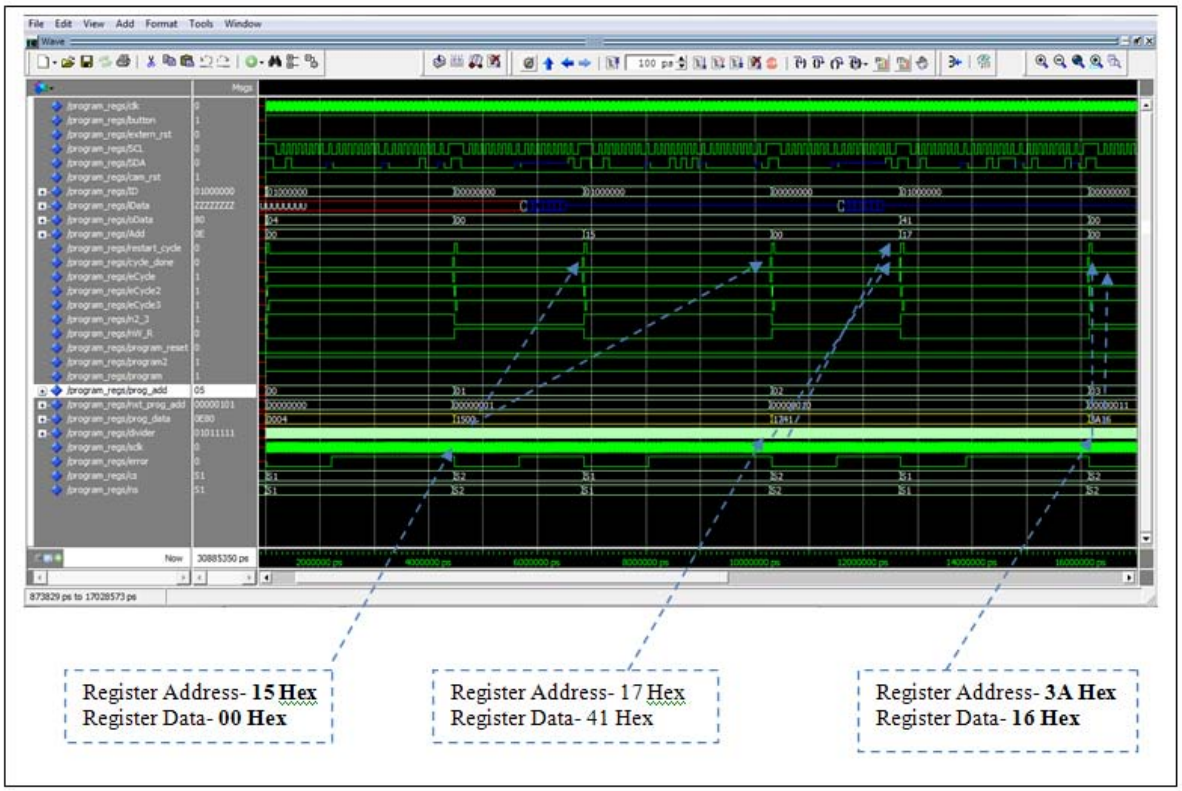


Figure 10.IIIrd Stage transition

5. Interfacing and result verification in ModelSim -The overall coding part can be write on VHDL and simulate on ModelSim It can be represented with the help of RTL diagram of I^2C and verified. There are many different techniques to interface FPGA with OV7620 single-chip CMOS VGA, CAMERA in place of normal communication I2C interfacing technique is implemented and result can be verified with ModelSim,

5.1 RTL Description- A RTL diagram of I2C is shown in figure..A result can be verified on ModelSim and effective and high speed communication link can be implemented and verified. The input values can be forced and Output being verified. The Address **1500**(0001,0110,0000,0000),**1741**(0001,0111,0100,0001), **3A16(**0011,1010,0001,0110) and the Id of the device is fixed i.e. **40**(0100,0000)Hex.



5.2-The I/O Pin description is given below

S.No	Pin Name	Pin Description
1.	Button	It used for reset The FPGA
2.	Clk	It is the clock given for entire system
3.	Extern_rst	It is the external reset to restart the system
4.	CAM_RST	It is the camera reset signal to reset the camera
5.	SCL	It is the Serial Clock line to provide the clock
6.	SDA	Serial Data line, It send the address of the register

4.4 - Advantage of Communication with I^2C

FPGAs are particularly well suited to meet the requirements of many data processing applications. FPGAs have the following characteristics that make them very appealing

- ***High performance:*** HD processing can be implemented in a single FPGA.
- ***Flexibility:*** FPGAs provide the ability to upgrade architectures quickly to meet evolving requirements, while scalability allows use of FPGAs in low-cost and high-performance systems.

- ***Low development cost:*** Data development kits from start as low as US$1,095 and include the software tools required to develop a data system using FPGAs.
- ***Obsolescence proof:*** FPGAs have a very large customer base on ship products for many years on. In addition, FPGA designs are easily migrated from one process node to the next.
- ***Plan for lower production costs:*** offers several ways to help plan for the time when products move from lower unit volumes to much higher volumes.

Conclusion.

The ideal surveillance architecture with I^2C will have the following characteristics: high performance, flexibility, easy upgradability, low development cost, and a migration path to lower cost as the application matures and volume ramps. 's FPGAs in conjunction with the feature-rich Data and Image Processing Suite, Data over IP reference design, and partner's compression solutions offer data system designers all the key building blocks needed to produce such a system.

REFERENCES

[1] THE I^2C BUS SPECIFICATION VERSION 2.1 January, 2000, Philips Semiconductors.
[2] Fred Eady "Networking and Internetworking with Microcontrollers" Elsevier, 2004.
[3] J. Bhasker, A VHDL Synthesis Primer, BS Publications 2nd Edition, 2003, pp. 132.
[4] TMS470R1x Inter-Integrated Circuit (I^2C) Reference Guide, Texas Instruments (SPNU223).
[5] Design Ware Inter-IC (I^2C) VIP Data book, Version 1.10a, February 18, 2005, Synopsis.Figure 6. Flow Summary of the Device 1.
[6] P.Venkateswaran, A. Saynal, S. Das, S.K Saynal and R. Nandi , "FPGA Based Efficient Interface Model for Scale-Free Computer Networking using I^2C Protocol" 15th.intl'conf on computing – CIC 2006, Proc. Research in Computing Science: Special Issues – Advances in Computer Science &Eng., ISSN 1870 – 406, pub .National Polytechnic Institute, Mexico, Vol.23, pp 191 -198 , Nov. 21-24,2006
[7] Xavier Righetti Xavier Righetti "Proposition of a Modular I2C-Based Wearable Architecture"IEEE-2010
[8] Jan O. Borchers, Wolfgang Samminger, Max M¨uhlh¨auser "Personal Orchestra: Conducting Audio/Video Music Recordings" IEEE-2002.
[9] A.R.M. Khan, A.P.Thakare, S.M.Gulhane "FPGA-Based Design of Controller for Sound Fetching from Codec Using Altera DE2 Board" International Journal of Scientific & Engineering Research,
[10] A.K. Oudjida, M.L. Berrandjia, R. Tiar, A. Liacha, K. Tahraoui 'FPGA Implementation of I2C & SPI Protocols"a Comparative Study'IEEE 2009
[11] J.M. Irazabel & S. Blozis, Philips Semiconductors, "I2CManual, Application Note, ref. AN10216-0" March 24, 2003.
[12] F. Leens, "An Introduction to I^2C and SPI Protocols," IEEE Instrumentation & Measurement Magazine, pp. 8-13, February 2009.
[13] L. Bacciarelli et al, "Design, Testing and Prototyping of a Software Programmable ^{I2C}/SPI IP on AMBA Bus," Conference on Ph.D.Research in Microelectronics and Electronics (PRIME'2006), pp. 373-376, ISBN: 1-4244-0157-7, Ortanto, Italy, June 2006.
[14] F. Leens, "An Introduction to I^2C and SPI Protocols," IEEE Instrumentation & Measurement Magazine, pp. 8-13, February 2009.
[15] Xilinx Inc., "Virtex-II™ V2MB1000 Development Board User'sGuide". Available:
[16] Arvind Sahu[1] Ravi Shankar Mishra[2],Puran Gour[3], "Design and Interfacing of High speed model of FPGA using I2C protocol" Int. J. Comp. Tech. Appl., Vol 2 (3), 531-536
[17] www.semiconductors.philips.com//I2C/index.hml