SMART TV WITH MULTI SCREEN

Shantakumar B. Patil
Professor and HOD
Department of Computer Science & Engineering
Nagarjuna College of Engineering & Technology, Bengaluru, India.
Visvesvaraya Technological University

Premjyoti Patil
Professor

Suyash Sorte
Professor

Neha Sharma
Students 6th Sem

ABSTRACT

No longer do you need to decide which channel to display on your lone screen. Instead, the viewer gets access to all of the content he could want, with the only decision being simply where to point his eyes at any moment of time. Because of the advancements in technologies, the expectation of the young generation is increasing, sky is the limit. They always expect things whatever they dreamt must come true. They want to experience same comfort in real life too. This is true and possible with advancements happening in technical fields. In this paper we are proposing one such advanced technology, which finds its application in the field of television. With the advancement of technologies, today the TV channels are also enhanced. Every day we can find one or the other channels, coming with unique new programs. With the help of single screen TV we may miss some of the very important programs which are telecasted by the other channels. To overcome this problem we are presenting in this paper, a method which can split the TV screen into multiple sub screens, depending upon our requirement. Out of these screens we can zoom an interested one and remaining will be on the top screen of the zoomed channel. Before presenting this paper we have surveyed many technical papers on this topic. In the existing system we can split the single screen to multiple screens, but zooming of one screen hides all the remaining screens or else we can see multiple screens simultaneously. Recent technological and economic developments have led to widespread availability of multi-core CPUs and specialist accelerator processors such as graphical processing units (GPUs). The accelerated computational performance possible from these devices can be very high for some applications. Software languages and systems such as NVIDIA’s CUDA and Khronos consortium’s open compute language (OpenCL) support a number of individual parallel application programming paradigms. A single FPGA device at different rates. FPGA-based control systems can enforce critical interlock logic and can be designed to prevent I/O forcing by an operator. However, unlike hard-wired printed circuit board (PCB) designs which have fixed hardware resources, FPGA-based systems can literally rewire their internal circuitry to allow reconfiguration after

I. HIGH-CAPACITY FPDS

Prompted by the development of new types of sophisticated field-programmable devices (FPDs), the process of designing digital hardware has changed dramatically over the past few years. Unlike previous generations of technology, in which board-level designs included large numbers of SSI chips containing basic gates, virtually every digital design produced today consists mostly of high-density devices. This applies not only to custom devices like processors and memory, but also for logic circuits such as state machine controllers, counters, registers, and decoders. When such circuits are destined for high-volume systems they have been integrated into high-density gate arrays. However, gate array NRE costs often are too expensive and gate arrays take too long to manufacture to be viable for prototyping or other low-volume scenarios. For these reasons, most prototypes, and also many production designs are now built using FPDs. The most compelling advantages of FPDs are instant manufacturing turnaround, low start-up costs, low financial risk and (since programming is done by the end user) ease of design changes.

II. FPGA

An FPGA is a device that contains a matrix of reconfigurable gate array logic circuitry. When a FPGA is configured, the internal circuitry is connected in a way that creates a hardware implementation of the software application. Unlike processors, FPGAs use dedicated hardware for processing logic and do not have an operating system. FPGAs are truly parallel in nature so different processing operations do not have to compete for the same resources. As a result, the performance of one part of the application is not affected when additional processing is added. Also, multiple control loops can run on a single FPGA device at different rates. FPGA-based control systems can enforce critical interlock logic and can be designed to prevent I/O forcing by an operator. However, unlike hard-wired printed circuit board (PCB) designs which have fixed hardware resources, FPGA-based systems can literally rewire their internal circuitry to allow reconfiguration after
the control system is deployed to the field. FPGA devices deliver the performance and reliability of dedicated hardware circuitry.

A single FPGA can replace thousands of discrete components by incorporating millions of logic gates in a single integrated circuit (IC) chip. The internal resources of an FPGA chip consist of a matrix of configurable logic blocks (CLBs) surrounded by a periphery of I/O blocks shown in Fig. 20.1. Signals are routed within the FPGA matrix by programmable interconnect switches. Signals are routed within the FPGA matrix by programmable interconnect switches and wire routes.

In an FPGA logic blocks are implemented using multiple level low fan-in gates, which gives it a more compact design compared to an implementation with two-level AND-OR logic. FPGA provides its user a way to configure:

- The intersection between the logic blocks and
- The function of each logic block.

Logic block of an FPGA can be configured in such a way that it can provide functionality as simple as that of transistor or as complex as that of a microprocessor. It can be used to implement different combinations of combinational and sequential logic functions. Logic blocks of an FPGA can be implemented by any of the following:

- Transistor pairs
- Combinational gates like basic NAND gates or XOR gates
- n-input Lookup tables
- Multiplexers
- Wide fan-in And-OR structure.

Routing in FPGAs consists of wire segments of varying lengths which can be interconnected via electrically programmable switches. Density of logic block used in an FPGA depends on length and number of wire segments used for routing. Number of segments used for interconnection typically is a tradeoff between density of logic blocks used and amount of area used up for routing. Simplified version of FPGA internal architecture with routing is shown in Figure 1.

Fig. 1 Simplified Internal Structure of FPGA

The High-Definition Multimedia Interface is provided for transmitting digital television audiovisual signals from DVD players, set-top boxes and other audiovisual sources to television sets, projectors and other video displays. HDMI can carry high quality multi-channel audio data and can carry all standard and high definition consumer electronics video formats. Content protection technology is available. HDMI can also carry control and status information in both directions. This specification completely describes the interface such that one could implement a complete transmission and interconnect solution or any portion of the interface. The underlying Transition Minimized Differential Signaling (TMDS)-based protocol and associated electrical signaling is described in detail. The mechanical specification of the connector and the signal placement within the connector are described. A device that is compliant with this specification is interoperable with other compliant devices through the configuration and implementation provided for in this specification. Mechanical, electrical, behavioral and protocol requirements necessary for compliance are described for sources, sinks and cables.

III. MULTI FORMAT PRODUCTION SWITCHER

4K operations with a suite of products routers, fiber infrastructure, and multi-format production switcher that support multi screen TV production and distribution. Fiber interfacing modules and world-class routing systems make multi screen TV content, sourced as four SDI streams at 1080 50/59p, available to the world-leading switcher, which supports a mix of SD, HD, 1080p, and now 4K. This enables broadcasters and production companies to manage the transition to multiscreen TV at no additional cost, whilst continuing to support hybrid SD, HD, 1080p, and 4K operations. Integrated multi screen TV distribution and routing are key in maintaining a reliable and efficient workflow in today’s emerging multi screen TV production environment. As an example, consider a pair of cameras acquiring content for a multi screen TV program. Each of the two multi screen TV four stream SDI signals are converted into
single mode fiber optic signals, and Snell’s IQCWM09 fiber optic CWDM module, in turn, would multiplex these eight fiber streams into one fiber output, suitable for transmission by Snell’s IQOTX80 multichannel fiber transmitter over considerable distances. Continuing our example, at the studio Snell’s IQCWM09 demultiplexer and IQORX80 multi-channel fiber receiver convert the signals back to two groups of four SDI streams, which can then be routed precisely together to the appropriate destination, such as the Kahuna 360, via Snell’s Sirius 800 series router. The 11-RU Kahuna 360 system can accept incoming multi screen TV feeds, mix them with 1080p, and output the result as multi screen TV, 1080p, or both formats simultaneously, helping broadcast and media organizations to adopt new workflows easily as opportunities arise.

IV. HTML 5

HTML 5 is the most recent version of Hyper Text Markup Language (HTML), a language proposed by Opera Software for Web (World Wide Web) content presentation. HTML is in the process of continual development since the day it was created in early 1990s and HTML5 is the fifth version of HTML standard. HTML5 covers the features of HTML4, XHTML 1 and DOM2HTML. It is a formatting language that programmers and developers use to create documents on the Web. With a number of new elements, attributes, 2D and 3D graphics, video, audio elements, local Storage facility, local SQL database support and a lot more exciting features, HTML 5 has brought a monumental change in World Wide Web. HTML 5 is nothing but using shorthand for continuous innovation in a client-centered application with new tags on a general development framework with CSS3 and JavaScript. It supports both desktop deployment and mobile deployment. Smart phones like Apple iPhone, Google Android, and phones running Palm WebOS have gained huge popularity with HTML 5 based rich Web applications. HTML 5 contains a number of new and easily understood elements in various areas. Some of these are as follows: HTML controls used in UI Multimedia file supports like video and audio Database support like local SQL database A number of new Application Programming Interfaces (APIs) These elements can be used in interactive websites with little customization to add an attractive effect and enhance the performance. Apart from the above features, the beauty of HTML5 comes with the fact that it supports the development of Web applications that can be used in offline mode when there is not a prolonged and constant access to the internet due to usage policies like social networking APIs (such as Facebook, Twitter and so on). To store the data locally and to allow the applications to run offline, HTML5 provides three different APIs and these are: Web storage: Can be used for basic localstorage with key-value pairs Offline storage: Can be used to save files for offline use indexDB: Supports relational database storage

V. HDL

In Electronics, A Hardware Description Language(HDL) is a specialized computer language used to design the structure, design and operation of electronic circuits and most commonly digital logic circuits.

The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC) (circuit diagrams were previously used to specify the configuration, as they were for ASICs, but this is increasingly rare). HDLs form an integral part of electronic design automation (EDA) systems, especially for complex circuits. Using HDL we can extract the various functions of the CPU. The FPGA consists of various transistors, linear transistors and diodes we can extract all these functions using HDL.

VI. RESULTS

![Selection of channels for the screen](image)

- The user has the option to select a list of four channels out of which one has to be run on the main screen. The main screen channel has both the audio as well as the visuals, whereas the other 3 channels have only the video mode.
Fig. 3 Screen one is zoomed and it is playing.

- Once the list of channels is selected this is how our TV screen would appear. We can see that the Screen1 is playing as the main screen.
- Let us say after some time the user wants to switch to the Screen2 on the top. The user has to just select this and the swapping of the Screens takes place as shown in figure 4.

Fig. 4 Screen two is zoomed and it is playing.

- Now Screen2 has both audio and visual mode whereas Screen1 now has only the visual mode.

REFERENCES

[1]. Skreen TV by Michael Carney 13 November 2014

[2]. 4K UHDTV -Opportunity or Hype by Snell


[5]. Take Control of Digital TV, Second Edition November 2007. Clark Humphrey. All rights reserved.

[6]. Enriched Multiscreen TV Experience toolkit By Daniel Ockeloen, Kati Hyyppä and Pieter van Leeuwen

[7]. From Television to Multi-Platform Less from More or More for Less? By Gillian Doyle

[8]. TV AND MEDIA By ERICSSON

[9]. The Multi-Screen Marketer Interactive Advertising Bureau (IAB)

[10]. Future Of Television And Media By EY

[11]. Hardware Description Languages: Voices From Tower of Babel By G.J. Lipovski

[12]. Fundamentals of HDL By Cyril P. R.

[13]. Advanced FPGA Design By Steve Kilts

[14]. FPGA Based Implementation of Signal Processing Systems By Roger woods, John Mc Allister and Ying Yi

[15]. HTML5 Foundations By Matt West

[16]. Dataprocessing on FPGA’s By Jens Teubner

[17]. VHDL By Douglas Perry

[18]. A Verilog HDL Primer By J. Bhaskar

[19]. Circuit Design with VHDL By Volnei A. Ped Roni

[20]. Programming HTML5–JS-CSS3