ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 12(29), DOI: 10.17485/ijst/2019/v12i29/147084, August 2019*

Development of IP Core to Transfer DTMF base Caller ID from FPGA to DSP Processor

V. S. Satish Kishore*, Prashant V. Joshi and K. M. Sudharshan

School of Electronics and Communication Engineering, REVA University, Bengaluru – 560064, Karnataka, India;
satishkishore@yahoo.com, prashanthvjoshi@reva.edu.in, sudharshankm@reva.edu.in

Abstract

Objective: As per the technical requirements, N telephone channels are to be processed along with caller ID information for further voice compression and effective storage. **Methods**: The received telephone single ended signal generated by using "Telephone's 2 to 4 wire converter" IC is fed as input to DTMF decoder IC for the caller ID extraction for the telephone call. "Telephone's 2 to 4 wire converter" IC is used to extract the ring detection and On/Off hook detection by which the telephone signal can be received and processed by using less bandwidth and power. **Findings**: Based on the survey, it was found that there is no single chip available in the market to extract caller ID using DTMF for number of telephone lines meaning that for each and every telephone interface one "DTMF based Caller ID extraction" IC has to be used. Similarly SLIC, SLAC and many other ICs available in the market were studied. Based on the comparison of those ICs with reference to their power consumption, foot print size, price, obsolescence, external battery for DC operation etc., it is proposed to go for the discrete solution for the extraction of DTMF based caller ID, Ring detection and On/Off Hook detection which achieves power consumption, power dissipation, cost and PCB layout area. Since the DSP process does not have sufficient number of I/O pins to connect telephone lines related control and data directly, FPGA has been chosen for the design to collect the telephone related control and data for N number of channels using the custom logic interface with the identified ICs. **Applications/improvements:** Test setup is arranged using the eval boards of FPGA, DSP processor, DTMF IC (CMX865A) and Silvertel's AG2120 Eval board to realise the proto type.

Keywords: Caller ID, Dual Tone Multiple Frequency (DTMF) IC, DSP Processor, FPGA, IP Core

1. Introduction

Since the DSP processor does not have sufficient number of I/O pins to connect more telephone lines related control and data directly, FPGA has been chosen for the design to collect the telephone related control and data for N number of channels using the custom logic interface with the identified ICs. These collected telephone data will be transferred to Processor using SPI interface. Many different types of interfaces such as USB, I2C, UART, LAN, McBSP and McASP were studied and discussed. Based on the utilization of those interfaces on the board for other purposes and since SPI interface is more sufficient to transfer the data for more telephone lines, it was recommended to use SPI interface to transfer the data between FPGA and DSP processor.

1.1 Features

- Inhibition of DTMF detection for selective characters.
- Enabling power down mode.

**Author for correspondence*

- Ring detection.
- On/Off hook detection.

2. Literature Survey

DTMF (Dual Tone Multiple Frequencies) is the signaling standard in telecom application that produces simultaneously 2 tones for each pressed key. Dual Tone Multiple Frequencies tones are chosen so that decoding key pressed from tones received is easier. In telecom embedded systems the Dual Tone Multiple Frequencies tone detection is a crucial block. In modern communication systems detection of Dual Tone Multiple Frequencies has been used[1].

A push-button (key phone or touchtone) telephone uses dual-tone multiple frequency signaling. It sends digit by means of combination of two frequencies, one from each of two groups of four frequencies, as shown in Figure 1.

To reduce risk of signal imitation this is done. Since each digit uses two frequencies and are not related harmonically, there is much less chance of the combination being produced by speech or room noise picked up by the telephone transmitter than if a single frequency were used.

In addition to the digits '1' and '0', the telephone keypad has buttons with the symbols '*' and '#'. These are used by SPC exchanges for facilities that are under the control of customers. For example, a customer who wishes incoming calls to be diverted to another telephone may key '*', followed by the appropriate instruction digits, before leaving. On returning, an instruction prefixed by '#' is keyed to remove the call diversion[2].

Coloumn frequencies (Hz)

Row frequencies (Hz)	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

Figure 1. Frequency coding used by pushbutton (key phone or touch tone) telephone sets.

3. Proposed Architecture/Block diagram

FPGA is the caller ID receiving unit. Most Significant Bit (MSB) is output to DSP processor from FPGA. The Figure 2 shows the caller ID Receiving unit.

The Figure 3 shows block diagram. "Telephone's 2 to 4 wire converter" IC is used to extract the ring detection and On/Off hook detection by using which the telephone signal can be received and processed by using less bandwidth and power.

The different types of interfaces are shown in Figure 4. UART, USB, McBSP, SPI and I^2C are within DSP Processor. Memory mapped register core is inside DSP Processor. Codec is used for compressing the audio.

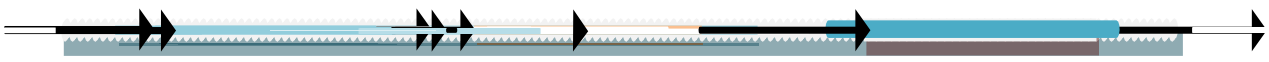

Figure 2. Caller ID receiving unit.

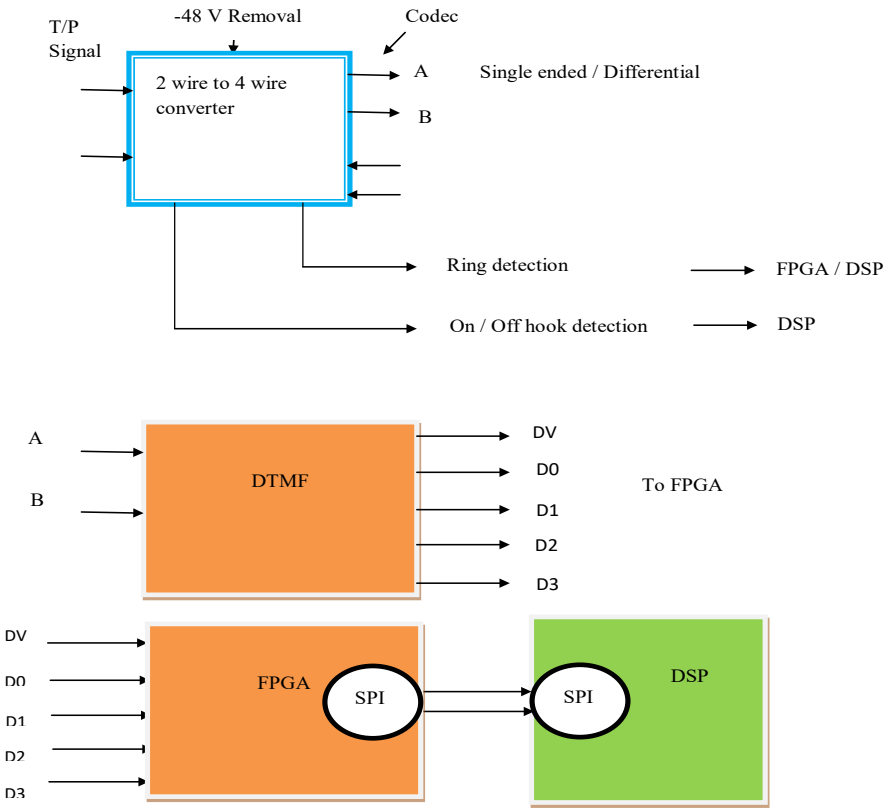

Figure 3. Block diagram.

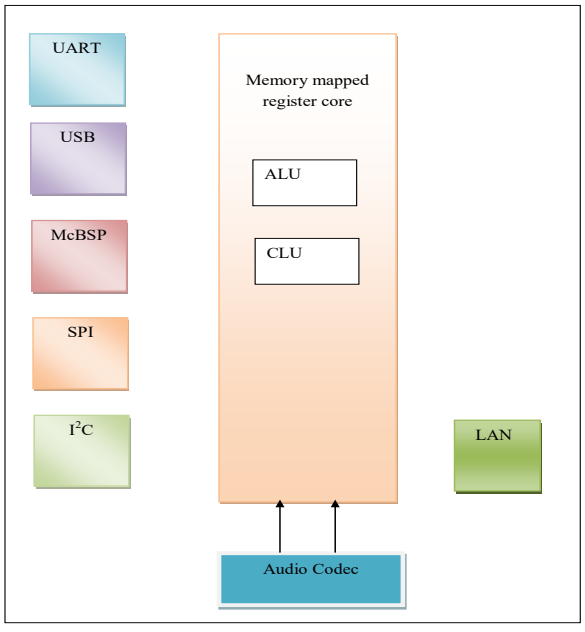

Figure 4. Different types of interfaces.

3.1 Module 1

The Figure 5 shows SPI to SPI communication. All slave select pins of the DSP Processor are connected to FPGA for 16 channel data transfer with channel ID. It is the approach 1 of module 1.

The inputs to the FPGA from DTMF are D0, D1, D2, D3 and DV (Data Valid) signals. The output OE (Output Enable) from FPGA is input to DTMF. DTMF is selected when OE is high. In Figure 6 PWDN should not be connected to high enabling power down mode[3].

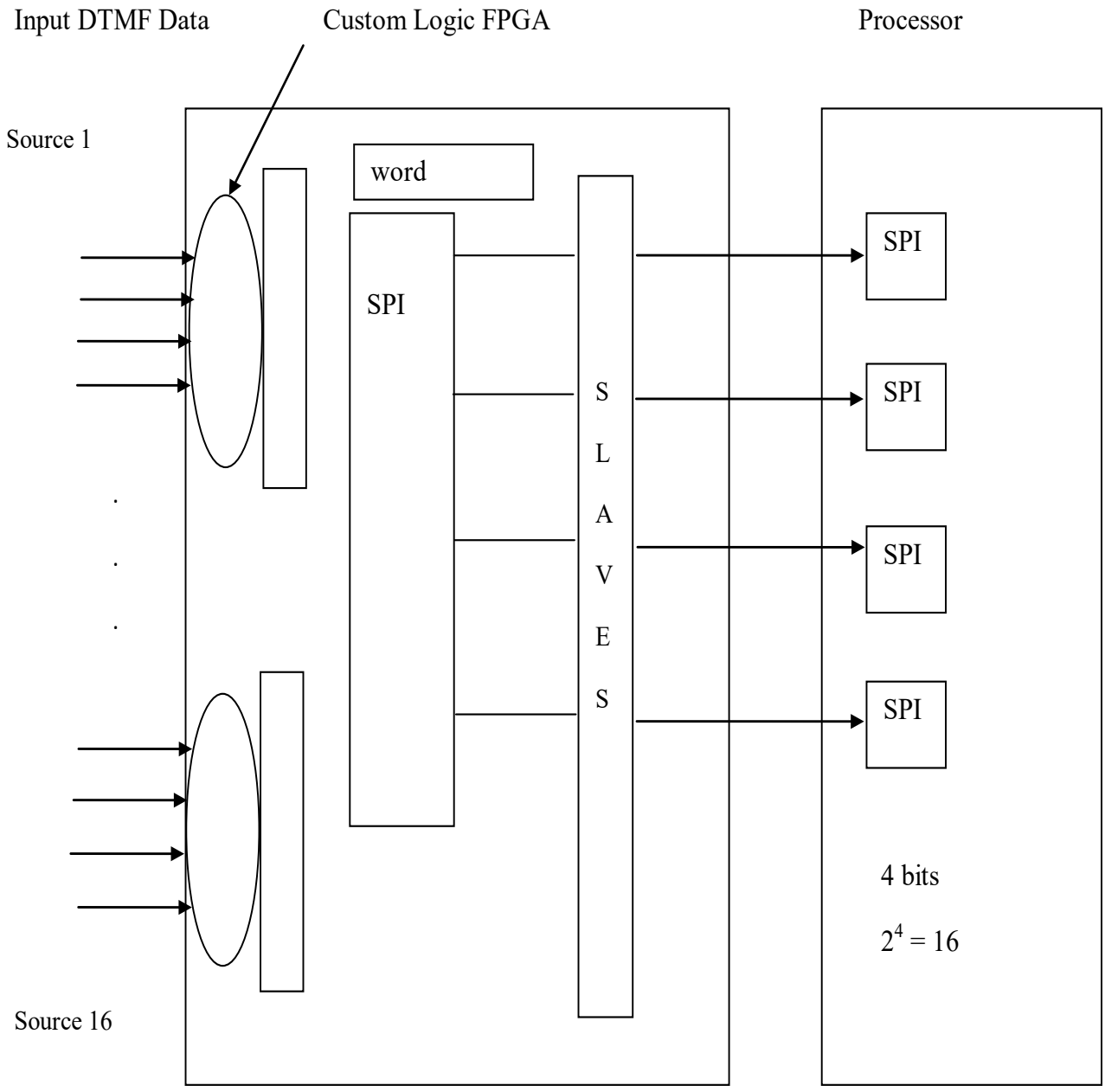

Figure 5. SPI to SPI communication.

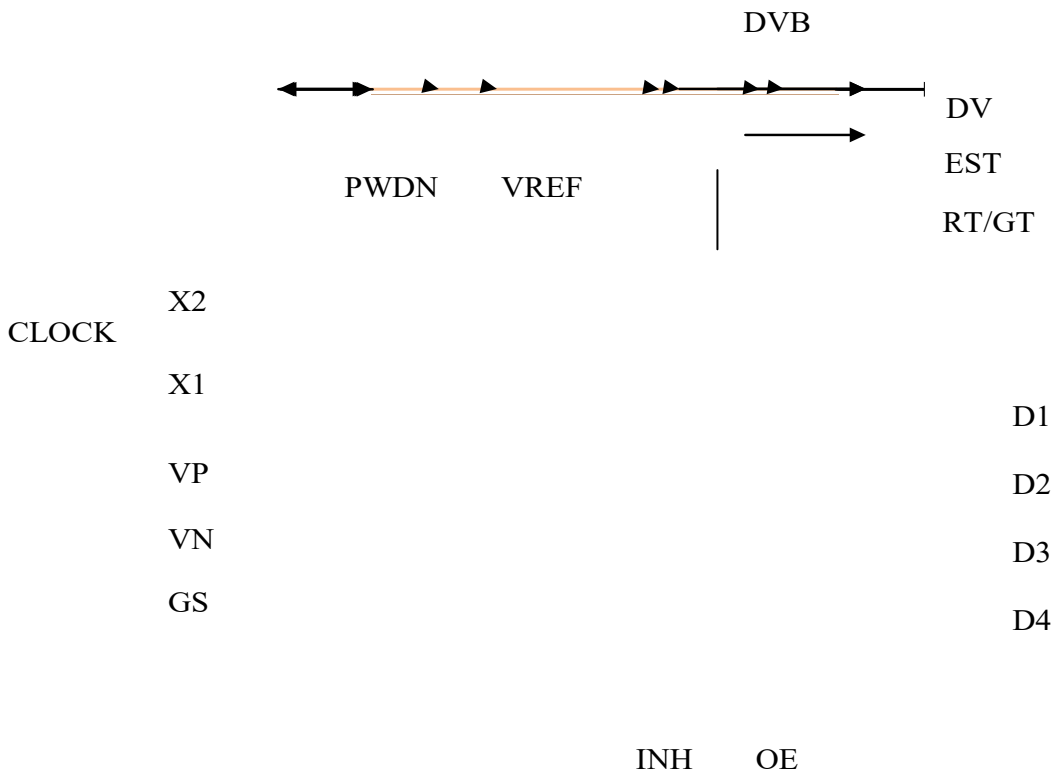

Figure 6. PWDN (pull down or pull low or weak low) should not be connected to high.

3.2 Features

- Inhibition of DTMF detection for selective characters.
- Enabling power done mode.
- Ring detection.
- On/Off hook detection.

4. Serial Peripheral Interface between DSP and FPGA

The telephone input signals are sent from DTMF decoder to Texas Instruments DM8168 processor through FPGA when the dialer dials the valid caller ID. The telephone signals can be received and processed by FPGA and DSP. The generated DTMF data is to be sent to the FPGA.

The DTMF data and caller ID is to be sent to the FPGA. Inputs to the FPGA are 4 bits DTMF data and data valid output signal from DTMF.

For each channel data is stored in a shift register. Master is in receiving mode. Master should know at what time slave is sending data. For that we have to use interrupt only. FPGA clock is 100 MHz (directly feeding to FPGA). For 4 slaves SPI clock is 24 MHz in the DSP processor DM8168. If one slave is used SPI clock is 48 MHz in the DSP processor DM8168[4].

The FPGA is divided into 4 slaves. Each slave can handle 4 channels. The 4 channels will have their respective shift registers (CH1SR1, CH2SR2, CH3SR3 and CH4SR4). Whenever a character is read for any channel interrupt is generated. Interrupt at the same time writes into Status Register and Control Register when the data is sent.

DSP processor will read and clear the Status Register to zero. The SPI can interrupt the DSP when it has completed a character transfer and when it has completed a frame transfer. Alternatively the DSP can poll the status registers.

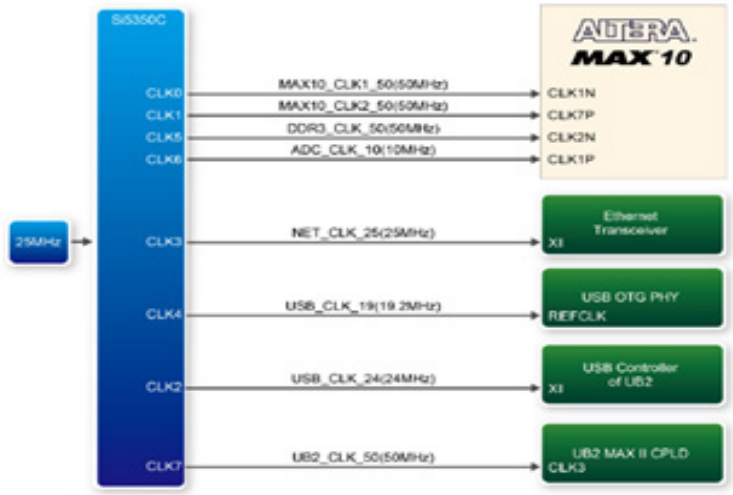

Figure 7. Block diagram of clock distribution on DECA.

The SPI cannot generate DMA events; therefore, it is the task of the DSP processor to receive data from the SPI data registers. Communication with SPI FPGA is carried out completely by SPI module. The SPI module drives the chip select pin of the device and allows the clock to shift data out.

The number of characters sent or received by the SPI module is specified. During the communication with a SPI device, the SPI module will assert the chip select pin until all the characters specified have been transferred. Figure 7 shows the default cock frequency of all external clocks to FPGA[5].

5. Results and Discussions

5.1 Simulation Results

Final result sample output FPGA channel 1 data ("0001"). DTMF caller id is 9 ("1001"), concatenated ch1_dataOp is

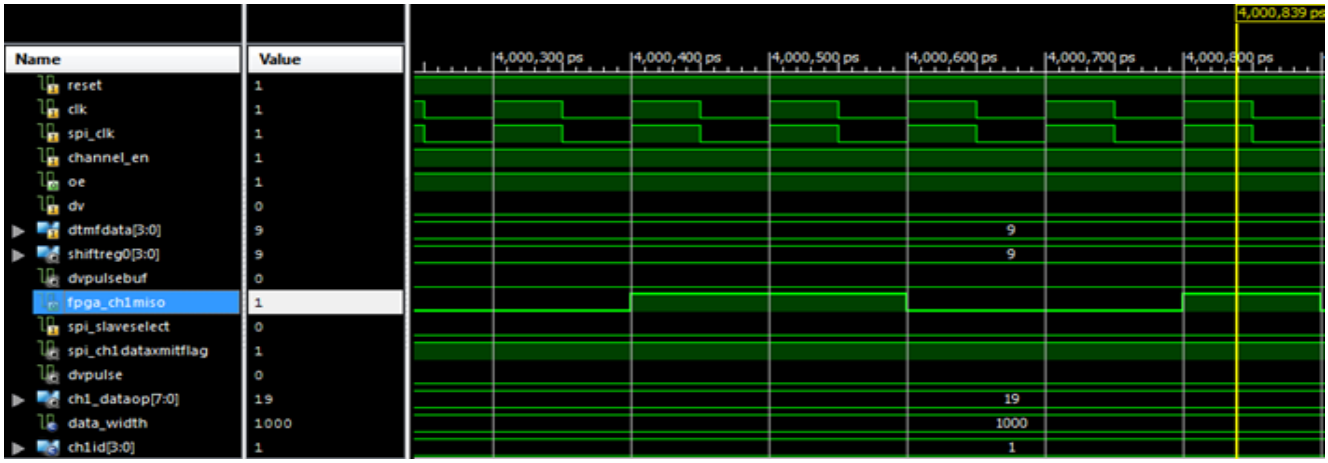

Figure 8. Integrated VHDL code for channel one waveform simulation.

19 and master in and slave out ("00011001") screen shot file. The data width is 8 bits.

Most Significant Bit is sent first sent from FPGA's MISO pin to DSP processor when spi Slave Select signal is '0'. The VHDL code Waveform Simulation for Channel One is shown in Figure 8.

5.2 Synthesis Results

The schematic diagram for the channel one FPGA Master Input Slave Output is shown in Figure 9. It consists of MISO, Output Enable (OE) and SPI channel one transmit

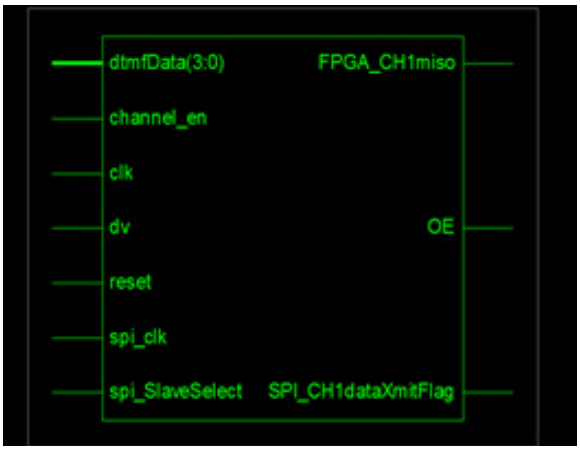

Figure 9. Schematic diagram of channel one.

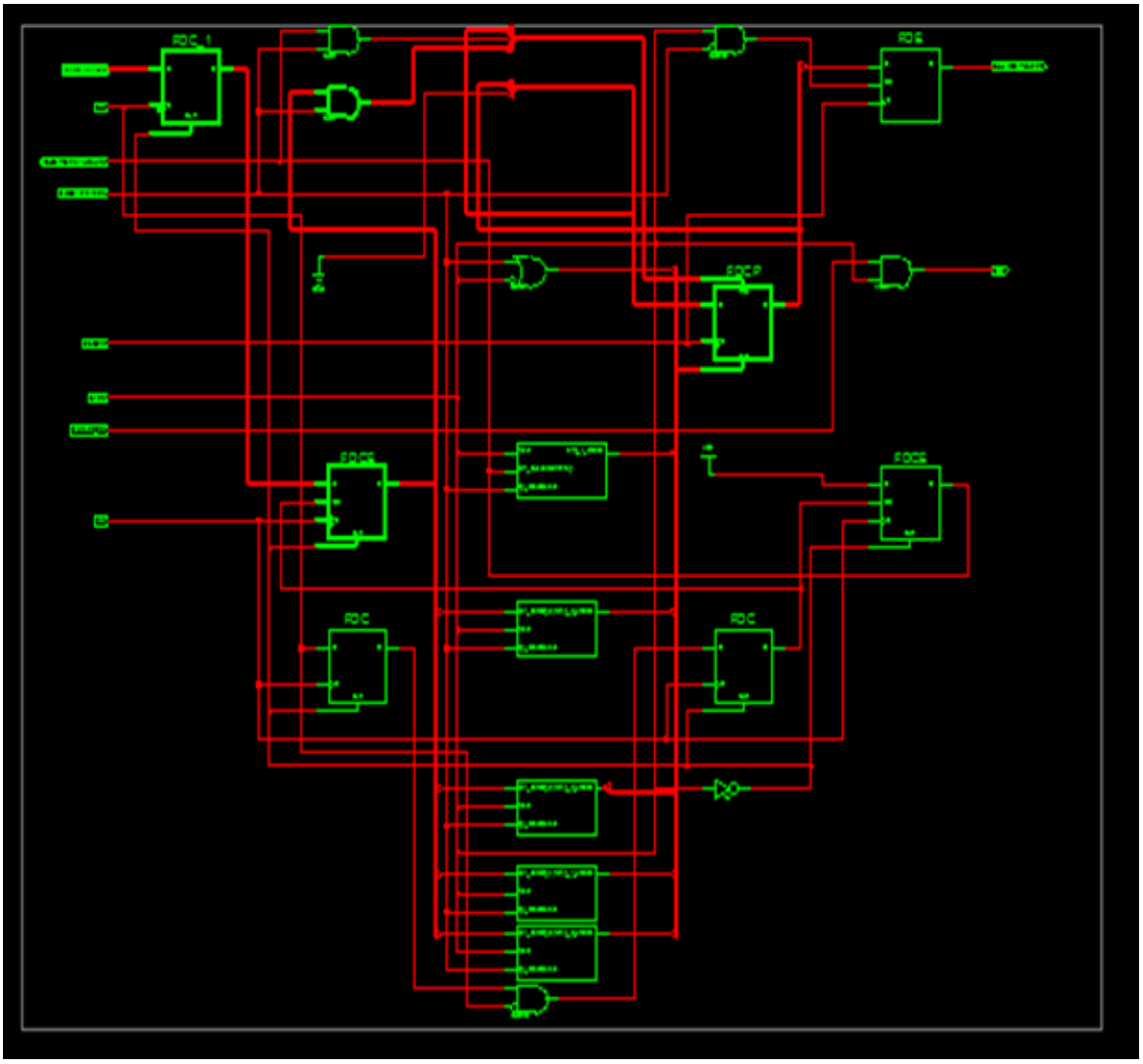

Figure 10. RTL schematic diagram of channel one.

flag status. The inputs to the FPGA are reset, clock, SPI clock, channel enable, data valid, DTMF data D0~D3 and SPI slave select signals. The RTL Schematic diagram of channel one is shown in Figure 10.

5.3 Device Utilization Summary

It consists of number of slices and number of IOs. From the Table 1 the bounded input output buffers percentage is 15%.

5.4 Timing Report

The timing report is shown in Figure 11. It consists of dv, clk and spi_clk signals.

5.5 Area Constraint Ratio

The area constraint ratio is shown in Figure 12. It shows found area constraint ratio of 100 (+ 5) on block Channel One, actual ratio is 6.

Table 1. Device utilization summary

Logic Utilization	Used	Available	Utilization
Number of Slices	12	192	6%
Number of Slice Flip Flops	16	384	4%
Number of 4 input LUTs	15	384	3%
Number of IOs	13		
Number of bonded Input-Output-Buffers	13	86	15%
Input-Output-Buffer Flip Flops	4		

```
TIMING REPORT

NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.
      FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT
      GENERATED AFTER PLACE-and-ROUTE.

Clock Information:
------------------
-----------------------------------+------------------------+-------+
Clock Signal                       | Clock buffer(FF name)  | Load  |
-----------------------------------+------------------------+-------+
dv                                 | BUFGP                  | 4     |
clk                                | BUFGP                  | 7     |
spi_clk                            | BUFGP                  | 9     |
-----------------------------------+------------------------+-------+
```

Figure 11. Timing report.

```
Found area constraint ratio of 100 (+ 5) on block ChannelOne, actual ratio is 6.

Final Macro Processing ...

Processing Unit <ChannelOne> :
INFO:Xst:741 - HDL ADVISOR - A 3-bit shift register was found for signal <temp_7> and currently occupies 3 logic cells (1 slice
Unit <ChannelOne> processed.

=========================================================================
Final Register Report

Macro Statistics
# Registers                                          : 20
 Flip-Flops                                          : 20

=========================================================================
```

Figure 12. Area constraint ratio.

5.6 Timing Summary

Speed Grade: -6

Minimum period: 3.53 ns (Maximum Frequency: 282.725 MHz).

Minimum input arrival time before clock: 5.406 ns.

Maximum output required time after clock: 7.085 ns.

Maximum combinational path delay: 8.936 ns.

Total memory usage is 160572 kilobytes.

6. Conclusion

The SPI Slave Controller designed is successfully, checked and verified by behavioral simulation and functional verification for all the given test cases using customized test bench in Intel Quartus Development Design suite. ModelSim in conjunction with Quartus Prime software is used to generate VHDL code Simulation Waveform.

A Test controller (for hardware checking of the above slave controller) is designed, successfully checked and verified by behavioral simulation and functional verification for the given test cases using a customized test bench in the Intel Quartus Prime Development suite. This Test controller is used to verify the functionality and operation of the Slave controller in the actual hardware.

From the expertise gained through this project, any product using the FPGA can be realized. Implementation using FPGA can be further taken into ASIC design.

7. References

1. Bhavanam SN, Siddaiah P, Ramana Reddy P. FPGA based efficient DTMF detection using Split Goertzel algorithm with optimized resource sharing approach. 2014 Eleventh International Conference on Wireless and Optical Communications Networks; 2014. https://doi.org/10.1109/WOCN.2014.6923072.

2. Flood JE. Telecommunication switching, traffic and networks. Signalling: Pearson Publ, India; 2016. p. 205–7..
3. HT9170D DTMF Receiver Datasheet. 2009. https://www.holtek.com.tw/documents/10179/11842/9170v111.pdf.
4. DSP Processor DM8168 Datasheet. 2015. http://www.ti.com/lit/ds/symlink/tms320dm8168.pdf.
5. DECA User Manual. 2015. https://static5.arrow.com/pdfs/2015/7/9/0/13/56/85/arrowd_/manual/deca_user_manual_rev1.pdf