DESIGN AND IMPLEMENTATION OF TREE DECODER FOR SDR APPLICATIONS

Er. Sachin kumar[1]
Assistant Professor,
ECE Department,
MERI College of Engineering & Technology, Sampla
er.sachinvlsi@gmail.com[1]

Er. Jyoti Gupta[2]
Assistant Professor,
ECE Department,
National Institute of Technology, Delhi
jyotig1989@gmail.com[2]

Er. Pushkar Praveen[3]
Assistant Professor,
ECE Department,
Ram-Eesh Institute of Engineering & Technology, Greater Noida
pushkarcdac11@gmail.com[3]

***Abstract*—The key objective of this project is to design a decoder which can be used for hardware purposes. Hardware, here accompanies with software which is more we can discuss as a Software Defined Radio application. The decoder implemented here offers to new radio equipment (SDR), the flexibility of a programmable system. Nowadays, the behavior of a communication system can be modified by simply changing its software. Large tree decoder can be constructed by reusing smaller similar sub-modules. Thus the structure is symmetric. The symmetric and regular structure of tree decoder makes the system easy to design. The structure obeys regularity and modularity concepts of VLSI circuit, thus is easy to fabricate using cell library elements. Design a Tree Decoder proposed architecture for SDR application on FPGA. The Structures made here are hardware synthesizable on FPGA board and are done in a respective manner. The design to be implementing by using Verilog-HDL language. The Simulation and Synthesis by using ISE Xilinx 13.4 tool.**

Keywords- Tree Decoder, Dynamic Decoder, Verilog, FPGA

I. INTRODUCTION

In any communication network, it is common we have two parts in which one is transmitter and other is receiver. At the receivers we use the decoders which are helpful in obtaining the desired information effectively. A decoder takes the coded information from a receive message and changes it into a recognizable form. A decoder is a combinational circuit that converts binary information from n input lines to a maximum of $m=2^n$ output lines.

Design of a high performance and efficient static, dynamic and tree decoder is very important for design of a frequency allocator but the main point is to allocate the specified band to assign a set of inputs which is then obtained at the desired output, thus the developing of a reliable and fast frequency allocator is a big problem in itself. A software-defined radio system, or SDR, is a radio communication system where components that have been typically implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators detectors, etc.) are instead implemented by means of software on a personal computer or embedded system. SDR needs to be reconfigurable to address the needs of flexibility and adaptability of SDR applications.

A. Static Decoder

Static decoders are simple line decoders which can be implemented using basic logic gates. In a hardware based static decoder, the frequency allocation is done in such a manner that it allocates the spectrum to the service providers statically. By the service providers the services are further distributed to the different users respectively.

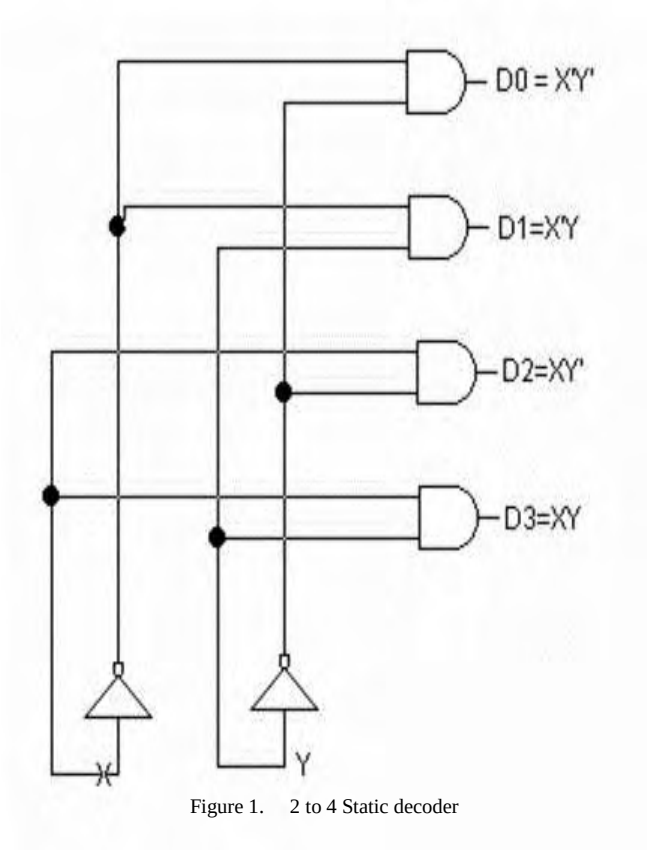

Figure 1. 2 to 4 Static decoder

TABLE 1. TRUTH TABLE FOR 2 TO 4 STATIC DECODER

INPUT		OUTPUT			
X	Y	D3	D2	D1	D0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

II. TREE DECODER

Large tree decoder can be constructed by reusing smaller similar sub-modules. Thus the structure is symmetric. The symmetric and regular structure of tree decoder makes the system easy to design. The structure obeys regularity and modularity concepts of VLSI circuit, thus is easy to fabricate using cell library elements. Tree structure requires fewer components with better area and delay optimization.

1) The 4 to 16 Tree Decoder Structure: It is consist of
4 inputs given as: X_0, X_1, S_1, S_2
1 enable input: Enable
16 output stages: Z_0 , Z_1 , Z_2 , Z_3 , Z_4 , Z_5 , Z_6 , Z_7 , Z_8 , Z_9 , Z_{10}, Z_{11} , Z_{12} , Z_{13} , Z_{14} , Z_{15}

It is composed of five number 2 to 4 decoders. The input to 4 to 16 decoder is divided into two equal halves and each half is named accordingly.

i.e. $X = (X_{left} + X_{right})$
where, $X_{left} = (S_2 , S_1)$ and $X_{right} = (X_1 , X_0)$

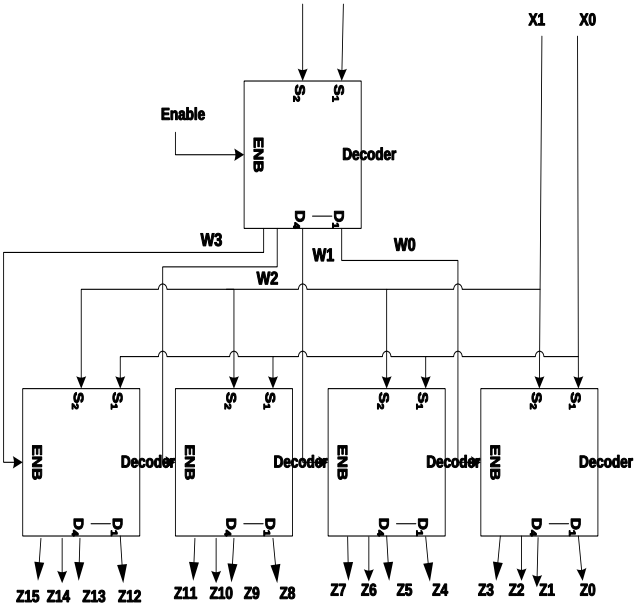

Figure 2. 4 to 16 Tree Decoder

2) The 6 to 64 Tree Decoder Structure: It is consist of:
6 inputs given as: X_0, X_1, X_2, X_3, X_4, X_5
1 enables input: Enable
64 output stages: Z_0 , Z_1 , Z_2 , Z_3 , Z_4 , Z_5 , Z_6 , Z_7 , Z_8 , Z_9 , Z_{10}, Z_{11} , Z_{12} , Z_{13} , Z_{14} , Z_{15} , Z_{16} , Z_{17} , Z_{18} , Z_{19} , Z_{20} , Z_{21} , Z_{22}, Z_{23} ,..... up to Z_{63}

It is composed of five number 2 to 4 decoders. The input to 4 to 16 decoder is divided into two equal halves and each half is named accordingly.

i.e. $X = (X_{left} + X_{right})$
Where, $X_{left} = (X_5, X_4, X_3)$ and $X_{right} = (X_2, X_1, X_0)$

The first 2 to 4 decoder at the top receives the X_{left} inputs of main input and enable as its input lines.

Mathematically,

$$W = Dec (X_{left}, Enable)$$

Where $W = W_0, W_1, W_2, W_3, W_4, W_5, W_6, W_7$ are the eight outputs of 3 to 8 decoder DEC.

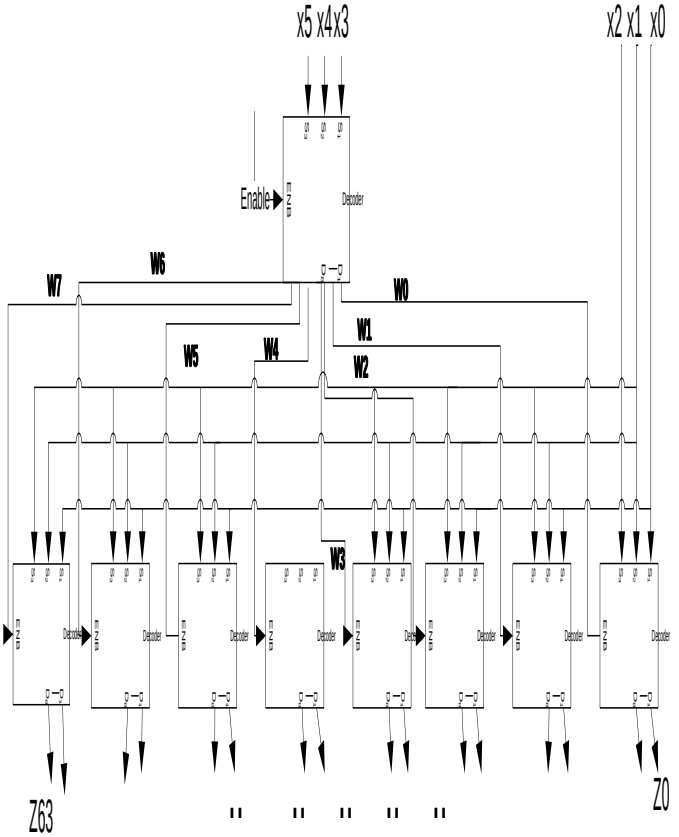

Figure 3. 6 to 64 Tree Decoder

The output lines of this top decoder become the enable input for below decoders at level-2. The inputs to level-2 decoders are the X_{right} part of main input. Thus, the final output is a decoder function of X_{right} and $W_{i..}$

$$Z_{0-63} = \text{Dec}((X_{right}, W_0), (X_{right}, W_1), (X_{right}, W_2), (X_{right}, W_3), (X_{right}, W_4), (X_{right}, W_5), (X_{right}, W_6), (X_{right}, W_7)).$$

Where Z_{0-63} is combined output of level-2 decoders i.e. DEC_0, DEC_1, DEC_2, DEC_3, DEC_4, DEC_5, DEC_6, and DEC_7.

III. PROPOSED ARCHITECTURE FOR FPGA BASED TREE DECODER

The FPGA board Spartan 3E (3s500efg320-5) used to implement the tree decoder has only 8 LEDs to display the output of logic program burnt on it. So, only 8 outputs can be shown at a time on Spartan 3E. This limitation of FPGA board is removed by a new logic encoded in the previous described tree decoder. The logic employed to show 16 output lines in two sets of 8 output lines each is presented next.

As we have to map 16 outputs onto 8 outputs of LEDs, we require one more enable input which works in such a way that it selects higher order 8 outputs if it is logic 1 and selects the lower order 8 outputs when it is logic 0. The function thus implemented can be given as follows.

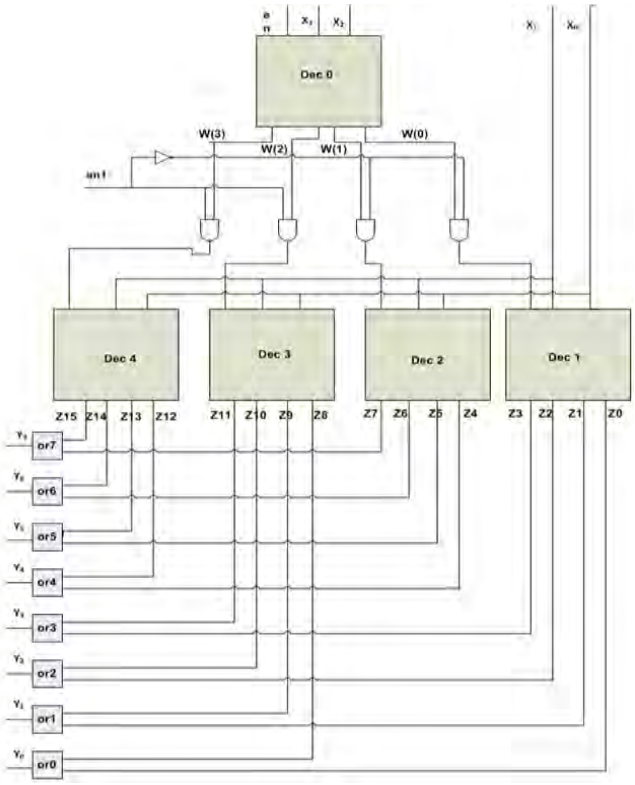

Figure 4. Purposed Architecture for FPGA based Tree Decoder

This structure of tree decoder consists of following parameters.

4 inputs given as: X_0, X_1, X_2, X_3

2 enable input: en and en1

16 output stages: Z_0 , Z_1 , Z_2 , Z_3 , Z_4 , Z_5 , Z_6 , Z_7 , Z_8 , Z_9 , Z_{10}, Z_{11} , Z_{12} , Z_{13} , Z_{14} , Z_{15}

It is composed of

1. 5 number of 2 to 4 decoders.
2. 4 AND gates
3. 8 OR gates

TABLE 2. TRUTH TABLE FOR PURPOSED ARCHITURE FOR FPGA BASED TREE DECODER

INPUT						**OUTPUT**							
en1	en	X_3	X_2	X_1	X_0	Z_7	Z_6	Z_5	Z_4	Z_3	Z_2	Z_1	Z_0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0	0	1	0
1	1	0	0	0	1	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	1	0	0
1	1	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	1	1	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	0	1	0	0	0	0	0
1	1	0	1	0	1	0	0	0	0	0	0	0	0
0	1	0	1	1	0	0	1	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0	0	0	0
0	1	0	1	1	1	1	0	0	0	0	0	0	0
1	1	0	1	1	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	0	1	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0	1	0
0	1	1	0	1	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	1	1	0	0	0	0	0	0	0	0
1	1	1	0	1	1	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	1	0	0	0	0
0	1	1	1	0	1	0	0	0	0	0	0	0	0
1	1	1	1	0	1	0	0	1	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	1	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0	0	0	0	0	0	0

The input to 4 to 16 decoder is divided into two equal halves and each half is named accordingly.

i.e. $X = (X_{left} + X_{right})$

where, $X_{left} = (X_3 , X_2)$

and $X_{right} = (X_1 , X_0)$

The first 2 to 4 decoder at the top receives the X_{left} inputs of main input and enable as its input lines.

Mathematically,

$$W = Dec (X_{left}, en)$$

Where $W = W_0, W_1, W_2, W_3$ are the four outputs of 2 to 4 decoder DEC.

The output lines of this top decoder are fed to four different AND gates. The second input to AND gates are from en1 input. The output of these AND gates become the enable input for below decoders at level 2.The two AND gates which works as enable input for higher 8 bit outputs is fed directly with en1 input while the rest two AND gates which work as enable for lower 8 bit outputs is fed through the complement of en1. The inputs to level 2 decoders are the X_{right} part of main input. Lastly, to map 16 outputs for two sets of 8 outputs each, the corresponding bits of higher 8 outputs are Red with corresponding bits of lower 8 outputs. Thus the circuit is implemented.

Mathematically,

When en1 = 0, Z shows lower 8 bits of output for given input combination

and

en1 = 1, Z shows higher 8 bits of output for given input.

Thus,

$$Z = fucn \{en1, Dec ((X_{right} , W_0) , (X_{right} , W_1) , (X_{right} , W_2) , (X_{right} , W_3)) \}$$

Where, Z is the combined output of level 2 decoders i.e. DEC_0, DEC_1, DEC_2 and DEC_3

III. SIMULATION & SYNTHESIS

A. RTL Schematic

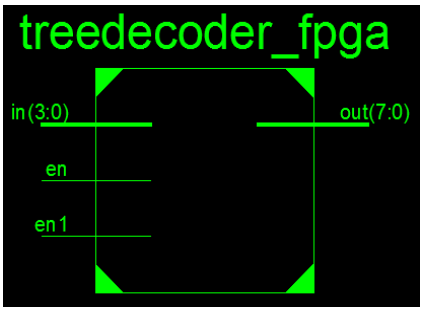

Figure 5. RTL schematic of FPGA based 4 to 16 Tree Decoder for SDR Unit with two Enable I/Ps

B. Detailed Schematic

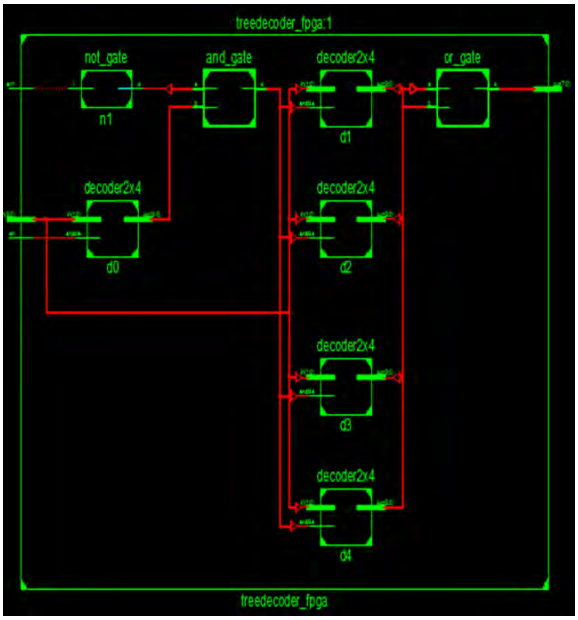

Figure 6. Detailed schematic of FPGA based 4 to 16 Tree Decoder for SDR Unit with 2 Enable I/Ps

C. Output Waveform

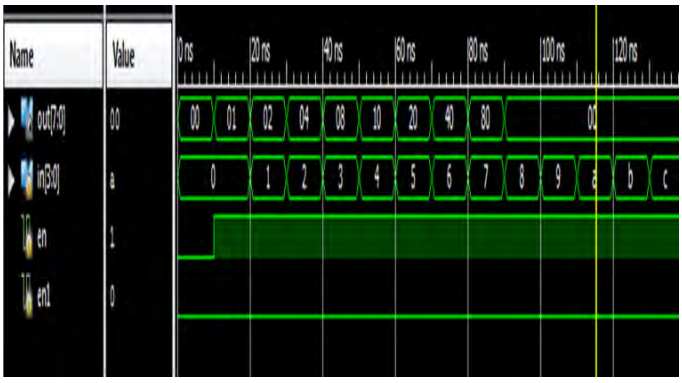

Figure 7. Output Waveform of FPGA based 4 to 16 Tree Decoder for SDR Unit with 2 Enable I/Ps

D. Design Utilization

TABLE 3. DESIGN UTILIZATION

Parameter	% Utilization
Number of Slices	11
Number of 4 input LUTs	20
Number of bonded IOBs	21
Maximum combinational path delay	7.158ns
Total Memory Usage	248404

CONCULSION

From the above results we conclude that the decoder implemented using tree decoding concept requires less number of components. Thus requires less area for fabrication, low power for working and less delay. Apart from these parameters, the circuit structure for a tree decoder is symmetric and regular. Tree decoders are formed of similar sub-modules, thus respects the modularity and regularity concepts of VLSI fabrication requirements. These structures can be modeled by regular blocks which are easily available in cell libraries and requirement of only similar blocks of circuits also easies the process of layout designing and fabrication. Thus, a decoder structure for SDR applications build using Tree Decoding concept is far superior to general static decoders and coincidence decoders.

REFERENCES

[1] A. F. B. Selva, A. L. G. Reis, K. G. Lenzi, L. G. P. Meloni, Member, IEEE and S. E. Barbin, Member, "Introduction to the Software-defined Radio Approach", IEEE Latin America Transactions, Vol. 10, Issue. 1, pp. 1156-1161 Jan. 2012.

[2] Anton S. Rodriguez, Michael C. Mensinger Jr., In SooAhn, and Yufeng Lu, "Model-based Software-defined Radio(SDR) Design Using FPGA", IEEE International Conference on Electro/Information Technology (EIT) 2011, pp. 1-6, May 2011.

[3] Timo, Vogt, and Norbert When, "A Reconfigurable ASIP for Convolutional and Turbo Decoding in an SDR Environment", IEEE Transctions on VLSI, Vol. 16, Issue 10, pp. 1309-1320, Oct 2008.

[4] Arul D. Murugan, Hesham El Gamal, Senior Member, IEEE, Mohamed Oussama Damen, Senior Member, IEEE, and Giuseppe Caire, Fellow, "A Unified Framework for Tree Search Decoding: Rediscovering the Sequential Decoder", IEEE Transactions on Information Theory, Vol. 52, Issue. 3, pp. 933-953, Mar 2006.

[5] Seunghyun Beak, Van Hieu, B., Park, G., Kyungtaek Lee, and Taikyeong Jeong "A New Binary Tree Algorithm Implementation with Huffman Decoder on FPGA", IEEE Transactions on communication, vol. 47, NO. 10, Oct. 1999.

[6] J. Mitola, "Software radios: survey, critical evaluation and future directions," IEEE Aerospace and Electronics Systems Magazine, vol. 8, pp. 25-36, April 1993.

[7] Rahul Kumar, R.C.Joshi and Raju, "A FPGA partial reconfiguration design approach for RASIP SDR", IEEE Annual on India Conference (INDICON) 2009, pp. 1-4, Dec 2009.

[8] T. R. Padmanabhan, B. Bala Tripura Sundari, "Design Through Verilog HDL" Wiley India Pvt Ltd, Jan 1, 2008.

[9] M. Morris Mano, "Digital Design", second edition, Jan 1, 1984.

[10] Charles H. Roth Jr., Larry L. Kinney, "Fundamentals of Logic Design", 2004.