

# **A Comparative Study on Cognitive Radio Implementation Issues**

**J S Banerjee**

Dept of Electronics and Communication  
Engineering,  
Bengal Institute of Technology,  
Kolkata, India

**K Karmakar**

Dept of Computer Science and Engineering  
Narula Institute of Technology,  
Kolkata, India

## **ABSTRACT**

Cognitive Radio (CR) is a comparatively new technology which is based on the revolutionary idea that provides solution to the underutilization of spectrum and spectrum scarcity problem. CR allows a group of users to identify and access to available spectrum resources for their optimum use. In this paper we describe the CR starting from the traditional Radio and Software Defined Radio (SDR). We also introduce the concept of reconfigurability. CR can be implemented using reconfigurable hardware. Therefore, it needs a platform that offers higher performance and reconfigurability. In this paper we describe different existing implementation of CR in reconfigurable platform and make a comparative study of them.

## **General Terms**

Cognitive radio: its implementation techniques on reconfigurable hardware and complexity analysis.

## **Keywords**

Radio communication system, SDR, CR.

## **1. INTRODUCTION**

Wireless communication is a hot topic for research works to the communication and computer engineers. Wireless devices are getting popularity everyday. Electromagnetic spectrum requirements have also increased. However, the process of availability of electromagnetic spectrum is very complex and costly. As a result, availability of spectrum and its proper use has been a major problem across the world. There are various agencies that are responsible for the allocation of frequencies. Firstly, the International Telecommunication Union (ITU) [1] and other International agencies allocate spectrum frequencies to different countries for their use. Secondly, within each country the respective government allocates frequencies to different agencies for different purposes. Finally, these agencies in their turn may reallocate the frequencies to the sub agencies. Huge costs are involved at every stage of allocation and availability. The frequency bandwidths thus available for use against huge money are expected to be used to their optimum capacity. But the matter of concern is that most of these valuable spectrums remain unutilized. According to a survey conducted by the Federal Communications Commission (FCC) [2] in the United States shows that only 30% of the allocated spectrums are properly utilized. Surveys conducted by other researchers and agencies like Office of

Communication (Ofcom) of the United Kingdom [3] also support this finding. One of the major reasons attributing to the underutilization of the costly spectrum is that present day communication devices are not capable enough so that it can use the spectrum to its optimum capacity.

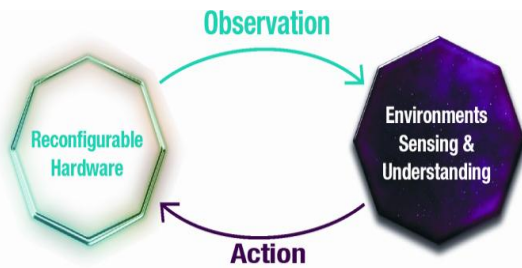
Communication devices like the traditional radio system supports fixed number of channels and multiple frequencies but the problem is that channel and frequency must be chosen at the time of design and not at a later stage because there is no mechanism to bring about any post-design adjustment at the time of and as per actual requirement. In search of a better solution to this problem, a comparatively new concept called Software Defined Radio (SDR) was introduced. It was designed based on the idea of the traditional radio system but with a little change to its design. In SDR radio's physical layer behavior and functionalities are defined in software. Although SDR brings a new concept and an important improvement of the idea of traditional radio, it is not completely fault-free. It does not have any intelligence and thus cannot take any decision of its own. To get over this limitation, a new concept called Cognitive Radio (CR) was improved on the basis of the idea of the SDR. CR removes the limitations of SDR. It has intelligence of its own. Therefore, it can take its own decision. It is capable of attaining awareness of its environment and is also capable of handling complex situations. It invented a new way of solving spectrum underutilization problem. It can sense the surrounding environment and try to find out the electromagnetic spectrum that is not utilized to its optimum capacity. After finding such range of frequencies, CR wants to utilize these properly by allowing the Secondary Users<sup>1</sup> to utilize it in the vacuum slots.

The main difference of CR from its predecessors is that it is defined by software and fully reconfigurable. All the logics in the CR can be implemented in software. Depending on the information CR senses, it can change its behavior and needs. It can be done through physical changes as well as by software instructions. In this chapter we have used the term

---

<sup>1</sup> Licensed users of spectrum band are known as 'Primary user (PU)'. Other unlicensed users of the spectrum band are known as 'Secondary User (SU)'.

‘reconfigurability’ several times. In this connection we like to mention that the term has been used to mean mainly to the ‘reconfigurability in the hardware’ (Fig1). Different reconfigurable hardware are available in the market for commercial use like ASIC<sup>2</sup> and FPGA<sup>3</sup>.



**Fig 1: Sensing the environment and make necessary changes**

This paper is organized as follows. Section I has been used as introduction. In section II we discussed different communication devices. In section III previous works like different implementation methods of the CR was described and a comparison among them was drawn. In section IV complexity analysis of these methods were done. Section V concludes the paper.

## 2. RADIO SYSTEM

### 2.3 Traditional Radio System

One of the significant inventions in the history of communication is the invention of radio communication system. It was developed in more than 100 years ago. Radio communication means the transmission of signals through the free space. Research for development of such wireless communication started many years ago and many individuals contributed to this research. As a result, radio can transmit voice as well as data to any part of the world.

Radio communication system constitutes a combination of different electronic devices namely transmitter, receiver and electronic circuits. For transmission of signals, it uses electromagnetic waves. Electromagnetic wave is a new form of energy which travels through the free space. Though development of radio communication technology brought about a revolution in this field, it has some limitations in terms of reconfigurability. Though the traditional radio supports multiple ranges of frequencies, but limitations of it is that it must be chosen at the time of design. Software defined radio (SDR) is a comparatively newer concept which helps remove the limitations of the traditional radio to some extent. But later with the invention of the CR it has been found that it

is a better choice than SDR. We discuss SDR and CR in the next section.

### 2.2 Software Defined Radio

Software defined radio (SDR) is a new concept developed on the basis of the idea of the traditional radio. In SDR different components of the radio communication system like amplifier and filters have been implemented in software instructions rather than the hardware. It can be upgraded easily. It was first used in Europe and USA for ‘defense’ purpose. In a word, SDR can be shown as:

$$\text{Software Defined Radio} = \text{Traditional Radio} + \text{Software defined functionality}$$

As per available records, a group of researchers of ‘Garland Texas division’ first used the term ‘software radio’ in 1984. Later, a group of German scientists designed and developed SDR. However, the term ‘Software Defined Radio’ was used by Joseph Mitola for the first time in 1991. Since then the term ‘Software Defined Radio’ got popularity. In SDR, behavior of the radio is defined and implemented by software. In it, all the major works of typical radio communication system like receiving, transmitting and processing of the signals can be done using software only. Different functionality of all the hardware components of a typical radio is implemented in software. SDR have several parts. Reconfigurable part is the most important part of the SDR. This part can be reconfigured to perform the radio functionality.

### 2.3 Cognitive Radio

We have already mentioned in the previous section on cognitive radio. It is a special type of radio which is intelligent enough to take its own decision. CR provides a unique solution to the approach of spectrum underutilization problem. It can sense the surrounding environment and depending on the information and requirement of the situation, alters its physical layer and reconfigures itself. It can change the configuration so that it can handle complex situations. Moreover, it can adapt to the new situation. According to the FCC [4], CR is a system which “*could negotiate cooperatively with other spectrum users to enable more efficient sharing of spectrum. A cognitive radio could also identify portions of the spectrum that are unused at a specific time or location and transmit in such unused white spaces, resulting in more intense, more efficient use of the spectrum while avoiding interference to other users*”. CR is as an intelligent device which combines the SDR along with intelligence. In a word CR (Fig. 2) can be shown as:

$$\text{Cognitive Radio} = [\text{Software Defined Radio}] + \text{Intelligence} + \text{Reconfigurability}$$

<sup>2</sup> An application-specific integrated circuit (ASIC) is one kind of Integrated circuit (IC)<sup>2</sup>, which is designed for specific use.

<sup>3</sup>FPGA is another reconfigurable hardware used in reconfigurable computing.

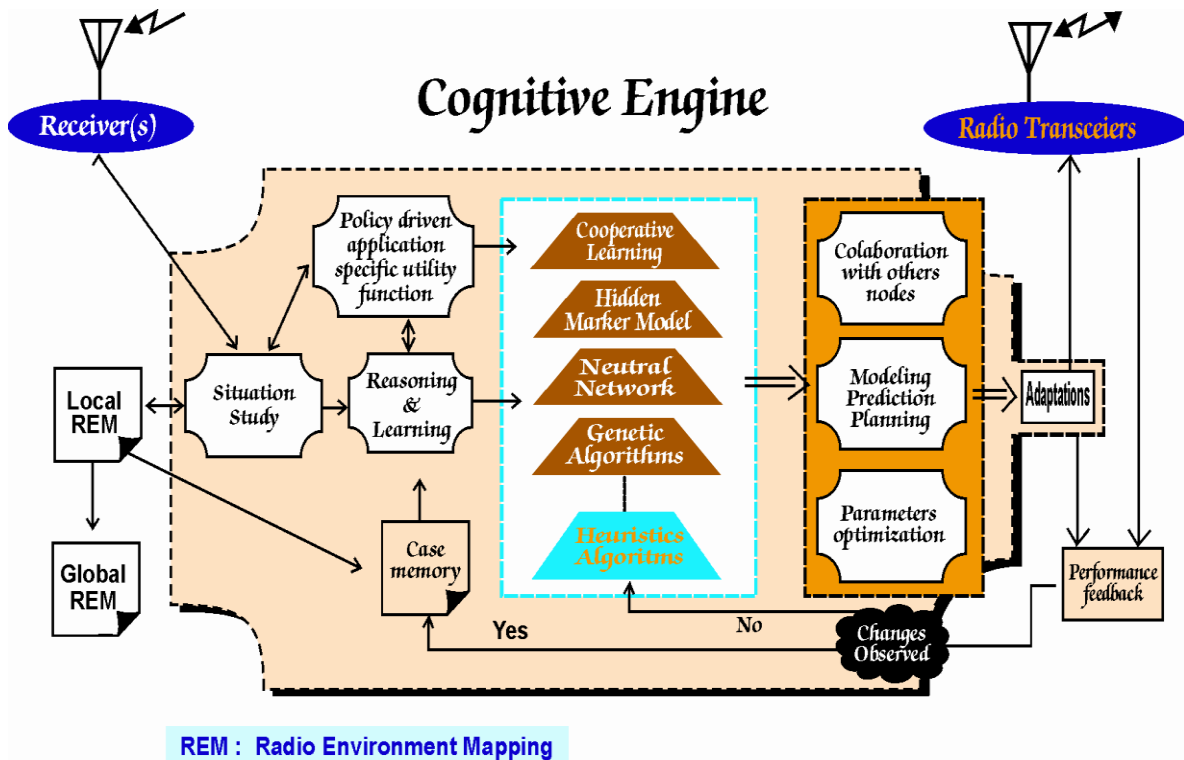


Fig. 2: Block diagram of cognitive radio

Joseph Mitola first used the term ‘Cognitive Radio’ in 1998. Later after one year in 1999 Mitola and Gerald Q Maguire published an article on CR. In his PhD thesis [5] Mitola explains: “The term cognitive radio identifies the point in which wireless personal digital assistants (PDAs) and the related networks are sufficiently computationally intelligent about radio resources and related computer-to-computer communications to detect user communications needs as a function of use context and to provide radio resources and wireless services most appropriate to those needs”.

Figure2 describes the architecture of the CR. Main part of the CR is its Cognitive Engine (CE), also known as the ‘brain’ of it. It is that part of the CR which has its own intelligence and can take its own decision based on different parameters and situations. Another important part of the design is that Radio Environment Map (REM) which is the central database of the CR. It keeps different types of information in its records. In the receiver and transmitter different sensor nodes are placed to sense the surrounding environment. These sensor nodes sense and store the information into the REM. From the current data and past stored information CR can reconfigure itself. Data in the REM database also updated in different times based on the information available through the sensors.

## 2.4 Reconfigurable Computing

In the last few years tremendous growth in the semiconductor industry has been noticed. Due to the wide applications of semiconductors in different fields it has become a major area of research. Among different applications of the

semiconductors industry one of the most prominent areas of research is ‘reconfigurable computing’. The concept of ‘reconfigurable computing’ has been in use for some days. Reconfigurability means changing of the hardware structure with the help of programming as and when required. It helps providing a new way to the existing system. It helps adapt to various situations, add new functionalities to the existing one and upgrade to newer ones from the old versions of the systems at any subsequent stage after initial design and implementation. Some reconfigurable hardware supports more flexibility like providing the facility of partial reconfiguration, where the original system can be reconfigured in part without affecting the normal functions of the other parts of the system. Whereas in some systems, which does not support partial reconfiguration, cases when whenever any changes need to occur in the system, the entire system may need to configure again from the beginning. Implementation of the concept of reconfigurability needs supports from the hardware and the architecture. But all kinds of hardware do not support configurability. There are some specialized hardware which are configurable-friendly. Among the configurable hardware available in the market for commercial use ASIC and FPGA are widely used. In the following section we have discussed different existing implementation techniques of CR on reconfigurable hardware.

### 3. PREVIOUS WORKS

#### 3.1 Implementation

CR can be implemented completely on the reconfigurable hardware. Implementation consists of two different parts: Hardware implementation and Software implementation. In the following section we are giving brief description of these two types of implementation.

#### 3.2 Hardware Implementation

Depending on the information sensed by the sensors, functionality of the CR can be changed. It can be implemented on the reconfigurable hardware like ASIC and FPGA. Though it can be implemented on ASIC successfully but limitation of such implementation is that such architecture is not flexible because ASIC is a onetime programmable chip. FPGA provides a better solution to this problem. It is flexible enough and can be programmed multiple times. Functionality of the modules as well as of the architecture can be changed even after implementation on the FPGA platform. Using different instructions the same FPGA can be tuned to different applications. Moreover, some FPGA supports partial reconfiguration. Therefore, CR can be implemented on FPGA at running time and its functionality can be changed at a later stage depending on the requirements. Parameters which control the functionality of the components can also be changed at a later stage. It adds flexibility to the implementation technique. Examples of such frameworks where CR have been implemented on reconfigurable hardware have been shown in the next section.

#### 3.3 Software Implementation

Implementation of CR on the reconfigurable hardware can be done with the help of a set of software instructions. Reconfigurable hardwares can be tuned to a specific application by means of software instructions only. CR can implement complex algorithms. Designers use C++ languages for implementation of such algorithms. For hardware implementation, other programming languages, like VHDL and Verilog can be used. These are hardware programming language used to program the reconfigurable hardware to tune to a specific application. After compilation, VHDL / Verilog programs generate executable codes known as 'software bitstream'. These bitstreams are downloaded on the hardware to configure it to a specific application. Once configured, static architecture is sufficient and works fine. Reconfiguration is required only when the CR senses the surrounding environment and decides that accordant changes in the configuration is required to maintain a balance with the outside world. These changes must be made with extreme care and intelligence. To achieve this goal both hardware and software implementations are required. To transfer data between software and hardware components balance is needed to maintain between the software and hardware components.

#### 3.4 Implementation of Cognitive Radio using FPGA:

##### 3.4.1 Method1

Several research works have been carried out on how to 'implement cognitive radio on FPGA platform' practically. In the following section example of such an implementation has been discussed:

An excellent method for implementation of cognitive radio nodes was proposed by J. Lotze *et. all* [13,14,15,16]. According to the conceptual framework proposed by them, a practical implementation can be divided into following three different parts:

I) A virtual architecture [15]: Framework for implementation is designed without considering the underlying implementation details. The architecture thus designed can be mapped directly to the FPGA board. It consists of two different parts: the Processor Subsystem (PS) and the Customizable Processing Subsystem (CPS). Duty of the PS is to execute the software instructions whereas the duty of the CPS is to implement them on the FPGA board. The objective of the CPS is to implement it as a partially reconfigurable area. This area can be reprogrammed without hampering the normal activity of the other parts. CPS is used for radio configuration.

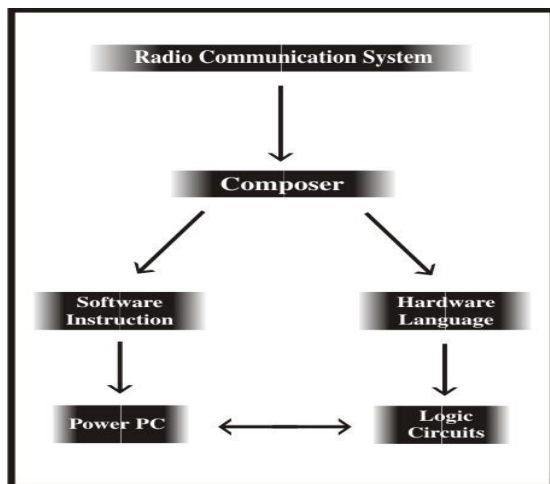
II) An adaptive runtime system [15]: Supports the reconfiguration process and manages all the processes during run time reconfiguration. An XML file will maintain the flow of data, components and other parts required and other details information. System execution is managed by this run time system. According to J. Lotze *et. all* [15] the architecture for implementation is divided into two different parts: processing and control plane. The processing plane performs all the computations required. The control plane is the part which is used to manage the system.

III) High level design tools [15]: Tools that helps such designers of the cognitive radio network who don't have knowledge about FPGA implementation. It helps system-level design and selection of hardware components where the design will be implemented. Two parts of the design is to design the processing chain and the cognitive process. Processing chains are typically described as flow graph. Another aspect of this design is to select which components will be selected for mapping. For this purpose a control specification may be used. In this design, a Composer tool uses the control specification. It will generate the FPGA bitstreams which will be used by the runtime system for reconfiguration.

- 1) Prepare an application flow graph.
- 2) Development of a framework for implementing FPGA based network models.
- 3) It consists of i) Virtual architecture ii) An adaptive run-time system. iii) High Level design tools.
- 4) Virtual framework makes the architecture generic and independent of the final target board on specific FPGA is used and hence portable.
- 5) Design a composer which will start this mapping onto the corresponding hardware.
- 6) Control specification will make decision engine.

**Fig 3: Algorithm for method1**

One of the vital tasks for the cognitive radio is to sense the spectrum to detect any empty slot. Important components of the cognitive radio are transmitter, receiver and the cognitive engine. Transmitter can send any arbitrary frequency. The same transmitter can be used as the receiver also. Depending on the situation the transmitter reconfigures itself as a receiver. In their research works *J. Lotze et. all* [15], used *Xilinx XUP development board* [16] with a *Vertex II-pro FPGA architecture* for implementation of cognitive radio on FPGA. According to the design given in the paper [15], *FPGAs can be used* to be configured and reconfigured as the transmitter and the receiver of the cognitive radio. The composer takes the chains and control specifications and selects the best hardware to implement the module (Fig 4). It can also be used to implement the chain [15]. Cognitive engine is implemented in Power PC processor.



**Fig 4: Flow Diagram for implementation of CR on FPGA according to Method1**

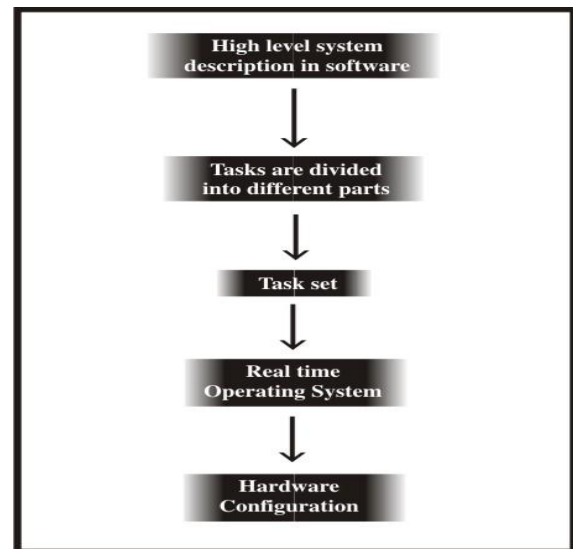
### 3.4.2 Method2

Another method for implementation of cognitive radio nodes was proposed by *Q. Zhang et. all* [10,12]. According to their proposal the design flow consists of the following steps:

Design flow starts with software description. Algorithm consists of a collection of functionalities which can communicate between them. Individual functional process is then compiled and a set of machine level codes were generated. This approach advocates the use of the Real time operating system (RTOS). RTOS selects the appropriate machine code and implements it onto the proper hardware. Thus the hardware is tuned to a particular application and works like a CR.

- 1) High level system description.
- 2) Tasks are divided into different subtasks.
- 3) Efficiency is judged and depending on some requirements they are analyzed separately first and then together.
- 4) A taskset is formed.
- 5) RTOS is used for mapping the set of tasks to the individual titles.
- 6) Several reconfiguration titles can be used.
- 7) Partial reconfiguration is possible.

**Fig 5: Algorithm for method2**



**Fig 6: Flow Diagram for implementation of CR on FPGA according to Method2**

Comparison: Two different methods for implementation of the CR on reconfigurable platform were discussed in the previous section. Both of them propose different approaches for implementation. Though there are similarities between these two approaches. In the following table (Table1) we have shown a comparison between these two approaches and find out their advantages and drawbacks.

Table 1: Comparison between above two methods

METHOD - 1	METHOD - 2
An application flow graph is prepared.	High level system description is given
Proposed to develop a framework consists of i) Virtual architecture ii) An adaptive run time system iii) High level design tool.	No such generic architecture is proposed.
Tasks are not divided into sub tasks.	Tasks are divided into subtasks.
A composer is prepared which is responsible for actual mapping.	NIL
NIL	Tasks are individually judged for finding their efficiency and also the task sets are formed.
Composer is responsible for actual mapping	RTOS is proposed to meet the dynamic needs of the variables wireless systems.
Based on single processor	More than one hardware titles / processor is proposed to use in the NOC.
Dosen't support partial reconfiguration	Partial reconfiguration is possible.

## 4. COMPLEXITY ANALYSIS

### 4.1 Time Complexity Analysis

#### 4.1.1 Configuration time Complexity:

Let, method1 requires 'D' rounds to complete for its configuration.

Therefore configuration time complexity may be approximated as  $O(D)$

Method2 supports partial reconfiguration. Here, out of 'D' rounds, let 'd' rounds need to be completed to configure partially. It leads to time complexity  $O(d)$  (where  $d < D$ ).

Therefore, we can conclude that  $(O(d) < O(D))$ .

Let, each round consumes 't<sub>c</sub>' time to configure.

Method1 consumes  $(D * t_c)$  configuration time.

Whereas Method2 consumes  $(d * t_c)$  (where  $d < D$ ) as it supports partial configuration time.

Therefore Method2 is faster than Method1 by

$$(D * t_c - d * t_c) = \{(D - d) * t_c\} \text{ time}$$

Let 't<sub>c</sub>' is the unit time.

Therefore method2 is faster than method1 by  $\{O(D - d)\}$ .

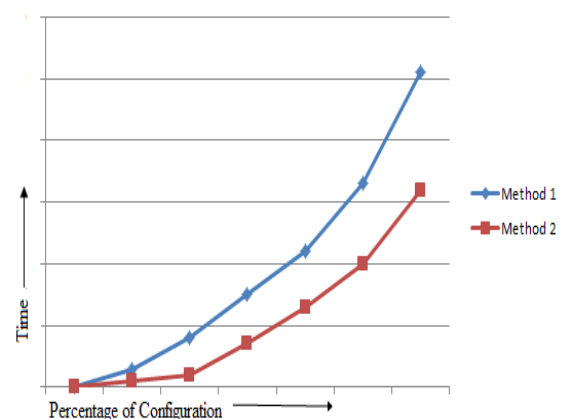


Fig 7: Configuration Time Complexity Graph



#### 4.1.2 Processing time Complexity:

Generally Processing tiles includes general purpose processor, domain specific reconfigurable hardware, DSP, FPGA and ASIC.

Let Method1 & Method2 process  $X_1$  and  $X_2$  number of tiles respectively in a speculated time ' $t_p$ '.

Tile processing rate of Method1  $= r_1 = (X_1 / t_p)$

Tile processing rate of Method2  $= r_2 = (X_2 / t_p)$

Taking the ratio of the processing rates we find-----

$$(r_2 / r_1) = (X_2 * t_p) / (t_p * X_1) = (X_2 / X_1) \dots\dots\dots(1)$$

According to the second method,  $(X_2 > X_1)$

$$\text{or } (X_2 / X_1) > 1 \dots\dots\dots(2)$$

Substituting the value of (2) in (1), we get  $(r_2 / r_1) > 1$

$$\text{or } (r_2 > r_1) \dots\dots\dots(3)$$

Now, to process x number of tiles,

Processing time by Method1  $(T_1) = (x/r_1)$

and processing time by Method2  $(T_2) = (x/r_2)$

$$\text{Therefore } (T_1/T_2) = (x * r_2) / (r_1 * x) = (r_2/r_1) \dots\dots\dots(4)$$

From (3) it is found that  $(r_2 > r_1)$

$$\text{or } (r_2 / r_1) > 1 \dots\dots\dots(5)$$

Therefore replacing the value of (5) in (4),

$$\text{we get } (T_1/T_2) > 1$$

$$\text{or } (T_1 > T_2) \dots\dots\dots(6)$$

Therefore we can conclude that processing time of method1 is greater than processing time of method2.

#### 4.1.3 Speed up time Complexity:

Method2 has employed parallelism to process say ' $n$ ' number of tasks. Let, ' $k$ ' segments are there to support parallel processing. Here time taken to process individual tasks is  $t$  (say)

Total time taken to process ' $n$ ' number of tasks  $(T_2) = (n * t)$ .

On the contrary 1<sup>st</sup> method does not use parallel processing.

Time taken to process individual tasks is  $(t * k)$

Total times taken to process ' $n$ ' number of tasks are  $(T_1) = (t * k * n)$

Second method is faster than the fast method by

$$(T_1/T_2) \text{ times} = (t * k * n) / (t * n) = k \text{ times.}$$

$$\text{Therefore } T_1 = (k * T_2).$$

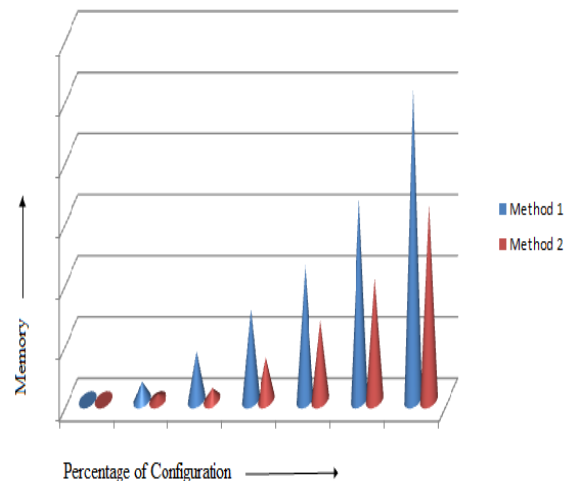
#### 4.2 Space-Complexity Analysis

In method1, enter configuration files are needed to be loaded during ' $D$ ' rounds to complete configuration. Let  $(M * N)$  memory space is required for this. Therefore complexity of method1 is  $O(M * N)$ .

In contrast method2 undergoes only ' $d$ ' (where  $d < D$ ) rounds to startup the process, as it supports partial configuration. Let here memory space consumption is  $(m * n)$  where  $(m < M \text{ and } n < N)$ . Space complexity claimed by method2 is  $O(m * n)$ .

Save in memory space, by second method is

$$O((M * N) - (m * n)) \approx O(\Delta(M * N))$$



**Fig 8: Space Complexity Analysis**

#### 4.3 Efficiency:

A process is considered to be efficient if it consumes less time to process the tasks and has less memory usage as well.

We can consider the efficiency parameter as inverse function of the time-complexity and space complexity in the following manner.

Before going to analysis the efficiency of the processes, let us define few parameters.

Let,  $\eta$  be the efficiency of the processes.

$\zeta_S$  be the Space- complexity of any process.

$\zeta_T$  be the Time - complexity of any process.

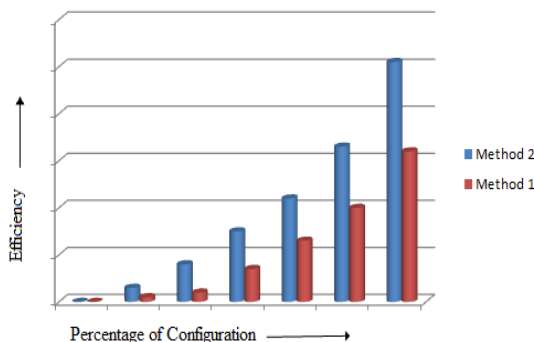
$$\eta = \frac{1}{(\alpha_1 * \zeta_S + \alpha_2 * \zeta_T)}$$

Where,  $\alpha_1 + \alpha_2 = 1$

According to the requirement,  $\alpha_1$  &  $\alpha_2$  parameters are assigned value.

From the above analysis mentioned in paragraph 4.1 & 4.2,

It is found that method2 is more efficient the method1.



**Fig 9: Efficiency Analysis**

## 5. CONCLUSION

In this paper we have shown the architecture of CR, different implement techniques and their performance analysis. CR tries to solve the spectrum shortage problem and optimizes the use of this costly natural resource. But CR is in a very preliminary stage and many problems are yet to be solved. According to the different possible implementation techniques one of the design methods proposes a composer. Future research can be directed towards building a new composer which will be much more intelligent and efficient. There are scopes for designing of new methods and derive high level tools for automation techniques. Research can be possible for deriving a method for finding out how to minimize the energy consumption by the semiconductor chip while configuring the hardware into the CR. Second method described in this paper advocates the use of RTOS. We can write an algorithm for

optimum use of the RTOS. However, the objective of these researches is to set a platform where CR can be implemented efficiently and its functionality can be tuned to its optimum capacity.

## 6. REFERENCES

- [1] The International Telecommunication Union <http://www.itu.int>.
- [2] Federal Communication commission <http://www.fcc.gov>.
- [3] Office of the Communication <http://www.ofcom.org.uk>
- [4] FCC ET Docket No. 03-108, "Facilitating Opportunities for Flexible, Efficient and Reliable Spectrum Use Employing Cognitive Radio Technologies", December 2003.
- [5] J. Mitola III, "Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio". PhD Dissertation, Royal Institute of Technology (KTH), Sweden, May 2000. [http://www.it.kth.se/~jmitola/Mitola\\_Dissertation8\\_Integrated.pdf](http://www.it.kth.se/~jmitola/Mitola_Dissertation8_Integrated.pdf).
- [6] Mitola J., III; Maguire, G. Q., Jr., "Cognitive Radio: making software radios more personal," Personal Communications, IEEE, vol.6, no 4 pp. 13 – 18, Aug 1999.
- [7] K E Nolan, P Sutton, L E Doyle, "An Encapsulation for Reasoning, Learning, Knowledge Representation, and Reconfiguration Cognitive Radio Elements".
- [8] S. haykin, "Cognitive Radio: brain-empowered wireless communication," IEEE J. Select Areas Communication, Vol: 23, No:2, Feb. 2005.
- [9] P P et al. Cognitive radio emergency network – requirements and design. Proceedings of the IEEE Symposium on New Frontiers in Dynamic Spectrum Access, Nov. 2005.
- [10] Q. Zhang, A B J Kokkeler and G. J. M. Smit, "A Reconfigurable Radio architecture for Cognitive Radio in Emergency Networks", September 2006, Manchester U K.
- [11] K. Compton and S. Hauck "Reconfigurable Computing: A Survey of Systems and Software", ACM Computing Surveys, Vol. 34, No 2, No 2, June 2002, pp. 171-210.
- [12] Q Zhang, G J M smit, L T Smit, A Kokkeler, F W Hoeksema, M Heskmp "A Reconfigurable Platform for Cognitive Radio".
- [13] J Lotze, S A Fahmy, J Noguera and L. E. Doyle "An FPGA-based Autonomous Adaptive Radio".
- [14] J Lotze, S A Fahmy, J Noguera, L. E. Doyle and R. Essar "An FPGA-based Cognitive Radio Network".
- [15] J Lotze, S A Fahmy, J Noguera, L. E. Doyle and R. Essar "Development Framework for Implementating FPGA-Based Cognitive Network Nodes".
- [16] J Lotze, S A Fahmy, J Noguera, L. E. Doyle and R. Essar "Generic Software Framework for Applications on FPGAs".
- [17] A Nafkha, C Moy, P Leray, R Seguier and J Palicot "Software Defined Radio Platformfor Cognitive Radio: Design and Hierarchcal Management".