Dynamic Channel Coding Reconfiguration in Software Defined Radio

Ahmed Sadek[1], Hassan Mostafa[1,2], Amin Nassar[1]
[1]Department of Electronics and Communications Engineering, Cairo University, Giza, Egypt,
[2]Center of Nano-electronics and Devices, AUC and Zewail City of Science and Technology, Egypt.
{ ahmd.sadk@gmail.com, hmostafa@uwaterloo.ca, amin.nassar@yahoo.com }

***Abstract*— Digital Front End Reconfiguration is considered one of the most promising techniques to implement the Software Defined Radio (SDR) and the Cognitive Radio (CR), allowing the same set of hardware to accommodate Multi-Standard Communication Systems (MSCS). The benefit increases when the reconfiguration is not only dynamic but also takes place in real time without the need to switch off the system. This work shows the advantages of using the Dynamic Partial Reconfiguration (DPR) technique in implementing the SDR convolutional encoders for 2G, 3G, LTE, and WIFI communication standards. Experimental results reveal that the DPR implementations improves the SDR implementation area and power consumption by 67% and 64% compared to the full implementation of these standards. The full implementation design and the DPR-based design are implemented and experimentally tested on Xilinx Virtex 5 design kit XUPV5-LX110T.**

***Keywords*— Software Defined Radio; Dynamic Partial Reconfiguration; FPGA; Convolutional encoders**

I. INTRODUCTION

Wireless communication standards are continuously changing and upgrading to support new features and enable new technologies. Therefore, both of the base station and the user terminal need to adopt dynamic communication chains capable of supporting multiple standards which is denoted by Software Defined Radio (SDR) [1]. SDR implementation enables compact system implementation to reduce the Silicon area and the power consumption and lengthen the battery life. Moreover, SDR implementation makes it easy to accommodate new standards with limited hardware modifications which reduces the production time and cost. FPGA is a programmable IC configured to execute a certain application. The utilization of the FPGA in the digital signal processing is increasing due to the fact that it is the best compromise between the configurability and the speed. The flexibility of the FPGA allows it to be used in the hardware implementation of the SDR. The aim of this work is to highlight the benefits of using the DPR capability of the FPGA in the implementation of the SDR compared to the conventional full implementation.

This work compares between two different convolutional encoders system. The first one is denoted by General Encoder Module (GEM) where all encoders exist on the chip and a multiplexer is used to switch among them. The second convolutional encoder system uses the DPR technique where one encoder loaded to the chip at a time and is denoted by Single-Loaded Encoder Module (SLEM). The SLEM module exhibits lower area and power consumption compared to the GEM module at the expense of more latency delay and memory overheads.

The rest of the paper is organized as follows. Section II gives some background on the dynamic partial reconfiguration feature. Section III shows how the SDR system is used to implement the MSCS system. Section IV explains the GEM and SLEM implementations and simulation results. Some conclusions are drawn in Section V.

II. DYNAMIC PARTIAL RECONFIGURATION

As the complexity of the communications system increases, the upgrading of the fixed and mobile standards is growing. This upgrading requires extra effort and financial resources as well as compatibility with the old standards. One of the future trends is the hardware reusability which means using the same hardware to define multiple standards. This hardware reusability is achieved by using the DPR feature of the FPGA. The benefits of the DPR are its flexibility of redefining system modules, real time functioning while being reconfigured, saving power by replacing unused modules, and reducing the cost of accommodating new designs [2]. Reconfigurable FPGAs are divided into two categories. In the first category, a full image is loaded to the FPGA to run a certain application. This image does not change during run time but new values are set in the registers to change the mode of operation of the application. This approach is called parametrization [3]. The second type is reconfigurablity in which a new image is loaded to the FPGA to execute a new application. FPGA reconfiguration is either full or partial. In the full FPGA reconfiguration, the FPGA has to stop working to download the new bit stream. On the other hand, in the partial FPGA reconfiguration, a part of the bit stream is updated and loaded in the FPGA. The following are the main factors that define the DPR:

A. Configuration mode

Table I shows the different configuration modes for Xilinx FPGA Virtex-5 [4]. There are two methods for loading the configuration data to the FPGA, which is dependent on the way of reaching the configuration plan:

- Internal: Using internal softcore processors such as MicroBlaze or hardcore processor such as PowerPc through Internal Configuration Access Port (ICAP). ICAP primitive is a SelectMap-like protocol that provides access to the internal configuration memory [4].

- External: Using external controller DSP or another FPGA as a master. The reconfigured FPGA is used in slave mode and the Serial mode, JTAG or SelectMap are used to access the configuration memory [4].

TABLE I. CONFIGURATION MODE [4]

Configuration Mode	Type	Max Clock	Data Width	Max Bandwidth Bps (bps/8)
ICAP	Internal	100 MHz	32-bit	400 MBps
SelectMap	External	100 MHz	32-bit	400 MBps
Serial Mode	External	100 MHz	1-bit	12.5 MBps
JTAG	External	66 MHz	1-bit	8.25 MBps

B. Reconfigurable Module Style based

Depending on the design size, the needed reconfiguration is either difference based or module based:

- Difference Based: used for small designs to edit the connections of few LUTs. The bit stream file contains the difference between the different applications [5].
- Module Based: used for large design changes by reconfiguring a complete block with a new one [6].

C. Configuration Memory array type

There are two different methods for reconfiguring the FPGA logic depending on the internal architecture:

- 1D: A complete array column is reconfigured with the new bit stream such as the DPR feature implemented in Xilinx Virtex II. Fig. 1-a shows how a complete D1 column is reconfigured with a D2 column [7].
- 2D: It is similar to the memory structure as some specific cells are accessed not a complete column. This type exists in recent FPGAs families such as Xilinx Virtex 4, 5, 6, and 7. Fig. 1-b shows that D1 block is reconfigured with D2 block [2].

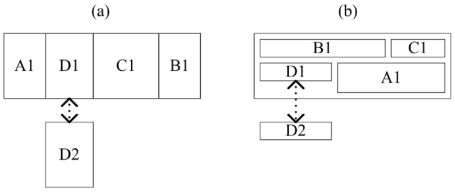

Fig. 1. Configuration memory array types.

D. Reconfiguration type

Reconfiguring part of the FPGA depends on how the data to be altered on the FPGA:

- Static: In this type, part of the FPGA is reconfigured while suspending the work of the FPGA. In this reconfiguration type, the communication systems will halt during the reconfiguration process [8].
- Dynamic: real-time reconfiguration during the normal FPGA operation, which reduces the reconfiguration time overhead. In this reconfiguration type, the communication systems are functioning during the new image reloading [8].

III. MULTI-STANDARD COMMUNICATION SYSTEM

Recently, increasing the connectivity among people is the main driving force for the wireless technology progress. This progress includes creating distinct standards such as 2G, 3G, LTE, WIFI, and Bluetooth. Unfortunately, the multi-standard communication system results in reducing the battery life due to the increased power consumption and inefficient utilization of the radio frequency spectrum.

SDR is one of the solutions that tackle the multi-standard receiver problem by reusing the same hardware to implement the multi-standard system with software reconfiguration [1]. Another approach that resolves the radio spectrum utilization problem is the Cognitive Radio (CR). CR is a way of using the available spectrum without interfering with the existing users. The implementation of the SDR with the CR concept leads to the MSCS by redefining the hardware blocks to accommodate different communication standards.

Fig. 2 simplifies the general block diagram of the wireless communication system. The smart antenna, RF front-end, Digital to Analog Converter (DAC), Analog to Digital Converter (ADC), and the Digital Signal Processing (DSP) module. The main focus of this work is on how the DPR enhances the area and power consumption compared to the full implementation method in the DSP module.

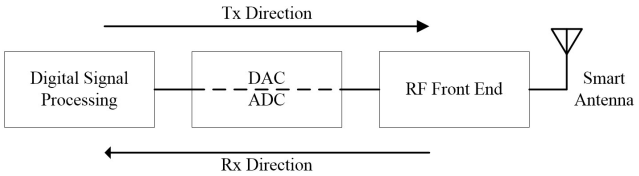

Fig. 2 Block diagram of communcation system.

2G, 3G, LTE and WIFI are widely used wireless communication standards. Implementing all these standards on the same Integrated Circuit (IC) results in larger Silicon area and higher power consumption for each channel. However, this area and power consumption can be improved if these standards are implemented by using dynamic and reconfigurable systems.

Fig. 3 demonstrates a simple communication channel adaptation that switches among different channel coding schemes. Applying the same concept to other blocks results in a completely reconfigurable system. A design of modulation chains using the DPR technique has been proposed in [9]. In [10], different implementations of channel coding schemes were compared. This paper investigates experimentally how the DPR technique is efficient in implementing multi-standard systems on the Xilinx Virtex 5 design kit XUPV5-LX110T. Furthermore, a study of the trade-off between the power consumption and the reconfiguration time for different MicroBlaze speeds in the DPR design.

The hardware switching is triggered during the handover between the two different communication systems. However, the handover takes hundreds of milliseconds [11] while the hardware reconfiguration takes much shorter time (i.e., in the microseconds range). On the other hand, many studies have been done on offloading the data traffic between cellular systems, e.g. 3G, and a fixed wireless systems, e.g. WIFI [12].

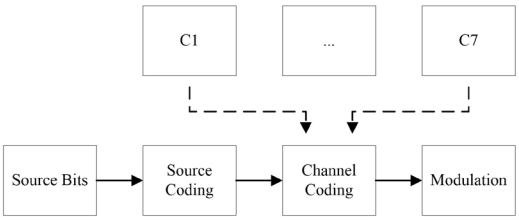

Fig. 3. Reconfiguration of simple communication chain.

IV. Implementation and Experimental Results

Convolutional encoders are forward error correction coding schemes used to add parity symbols for the data sent over a communication channel. These coding schemes decrease the effect of the noisy channel on the sent data. Three main parameters are used in the convolutional encoders [n, k, l], where n is the number of inputs, k is the number of outputs and l is the constraint length which indicates the number of memory elements used. The convolutional encoder rate is n/k. This rate can be increased by using a puncturing at the output of the encoder. Table II shows the characteristics of the different convolutional encoders used in 2G, 3G, LTE and WIFI technologies. Different convolutional encoders reflect different hardware connections between memory elements.

TABLE II. Convolutional Encoders Used In 2G, 3G, LTE and WIFI

Conv. Encoders	System	Channel	Rate (n/k)	Constraint length (l)	Generator polynomials (octal)
C1	2G	TCH/FR Speech	1/2	5	G0 = 31 G1 = 33
C2	2G	TCH/HR Speech	1/3	7	G4 = 155 G5 = 123 G6 = 137
C3	2G	Data	1/3	5	G1 = 33 G2 = 25 G3 = 37
C4	3G	BCH, PCH, RACH, DCH, FACH	1/2	9	G0 = 561 G1 = 753
C5	3G	DCH, FACH	1/3	9	G0 = 557 G1 = 663 G2 = 711
C6	LTE	BCH, DCI, UCI	1/3	7	G0 = 133 G1 = 171 G2 = 165
C7	WIFI 802.11a,g	OFDM channel	1/2	7	G0 = 133 G1 = 171

Two designs are presented in this work. First, GEM is shown in Fig. 4, where all the convolutional encoders exist on the FPGA at the same time and the desired one is selected through a multiplexer. Secondly, SLEM, where encoders are stored on external memory and loaded per request, is shown in Fig. 5. A SoC design using an embedded softcore processor MicroBlaze is utilized with simple operating software in both designs. The processing system consists of a MicroBlaze, SysACE controller to load bit stream files form compact flash, ICAP for internal FPGA configuration and UART for interfacing with PC. The test setup used for this work consist of a PC connected to Xilinx Virtex 5 kit XUPV5-LX110T through a serial cable. A PC terminal emulator Tera-Term is used to communicate with the Kit.

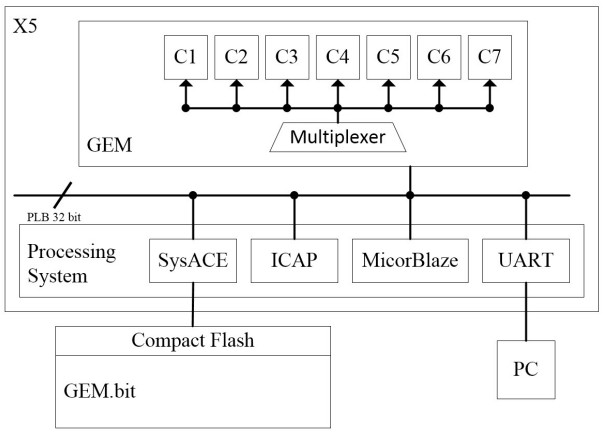

Fig. 4. General Encoder Module (GEM), where multiplexer used to switch among differenet encoders.

In GEM design, a complete image is loaded from the compact flash. The processing system is responsible for the software interface with the PC through UART and controls the multiplexer to switch among the encoders. In SLEM design, part of the FPGA programmable logic is configured as dynamic reconfigurable area using the Plan-Ahead tool. The FPGA is loaded initially by the convolutional encoder C1 in the dynamic part. Encoders (C1 – C7) are stored in compact flash. The needed encoder is loaded to the FPGA dynamic part on demand to replace the existing encoder. The MicroBlaze is responsible for software interface with the PC and manages the switching among the different partial bit stream files (C1 – C7), using ICAP to reconfigure the FPGA dynamic area with the selected encoder. Experimental test results by using the Xilinx Virtex 5 kit XUPV5-LX110T and the above test setup are presented as follows.

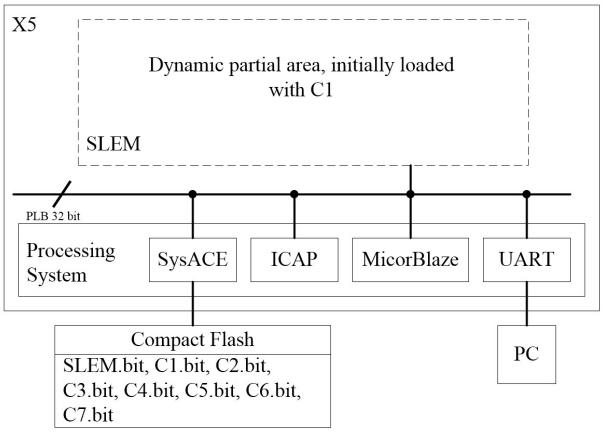

Fig. 5. Single-Loaded Encoder Module (SLEM), one encoder loaded at a time using MicroBlaze.

A. Area

The total area assigned for both designs (i.e., GEM and SLEM) on the Xilinx Virtex 5 kit XUPV5-LX110T is 800 LUTs. The FPGA utilization is calculated with respect to these assigned 800 LUTs. Table III shows the number of LUTs used

for both designs and their utilization percentage. It should be noted that the full implementation of the encoders (C1 – C7) should utilize 258 LUTs (i.e., the sum of the utilized LUTs of each encoder). However, to provide a fair comparison, the Xilinx programming tool optimizes the full implementation code which results in a utilization of only 113 LUTs as listed in Table III. It is evident from Table III that the DPR improves the area of the optimized full implementation from 113 LUTs to a maximum of 38 LUTs with area improvement of 66.4%. In addition, the utilization percentage is improved from 14.1% to a maximum of 4.8%.

TABLE III. AREA UTILIZATION FOR BOTH DESIGNS

Conv. Encoders	SLEM							GEM
	C1	C2	C3	C4	C5	C6	C7	
No. of LUTs	36	37	37	37	38	37	36	113
Utilization (%)	4.5	4.6	4.6	4.6	4.8	4.6	4.5	14.1

B. Memory needed

The initial file size of the GEM and SLEM designs is 3.8 MB. However, the SLEM requires additional memory for the partial bit stream files of each individual encoder. The size of each partial bit stream file is 60kB and accordingly, the extra needed memory for the SLEM is 420kB (because there are 7 encoders). Thus, the memory overhead of the SLEM is 10.8% compared to GEM.

C. Power Estimation

Xilinx Power Analyzer tool (XPA) is used to estimate the power consumption in the convolutional encoder modules. Table IV lists the power consumption of the GEM and SLEM designs for different MicroBlaze frequencies. Once again, it should be highlighted that the calculated power consumption of the GEM full implementation is after performing optimization with the Xilinx configuration tool for fair comparison. It is evident from Table IV that the DPR improves the power consumption up to 64%.

TABLE IV POWER CONSUMPTION

Operating Frequency	SLEM Power Consumption	GEM Power Consumption	Power Improvement
50 MHz	0.08 mw	0.18 mw	56 %
75 MHz	0.12 mw	0.33 mw	64 %
100 MHz	0.17 mw	0.44 mw	61 %

D. Time overhead

The initial configuration time for both designs (i.e., GEM and SLEM) is the same because the initial bit stream file has the same size 3.8 MB for both. In the full implementation there is no time overhead added because all encoders exist. In SLEM, partial bitstream files of size 60 kB is generated for each encoder. Correspondingly, this results in a reconfiguration time overhead. Table V shows the configuration and reconfiguration time for both designs. Also, Table V illustrates how the MicroBlaze frequency affects the configuration and reconfiguration time. According to this table, the reconfiguration time overhead in the SLEM design is insignificant and is less than 1.6%.

TABLE V. CONFIGURATION AND RECONFIGURATION TIME

Design	SLEM (C1 – C7)	GEM
Operating Frequency 50 MHz		
Configuration Time (ms)	19	19
Reconfiguration Time (ms)	0.30	0
Operating Frequency 75 MHz		
Configuration Time (ms)	12.7	12.7
Reconfiguration Time (ms)	0.20	0
Operating Frequency 100 MHz		
Configuration Time (ms)	9.5	9.5
Reconfiguration Time (ms)	0.15	0

V. CONCLUSION AND FUTURE WORK

DPR is a flexible and efficient way of realizing SDR in a Cognitive Radio system. Implementing a library of different encoders and switching among them reduces the system complexity and makes it handy and real time upgradable. Future work will include simplifying the communication system chain and generalizing the concept of DPR for other blocks. This would help in offloading between cellular systems and fixed wireless systems such as WIFI, not only on the network level but also on the hardware level.

ACKNOWLEDGEMENT

This work was partially funded by Zewail City of Science and Technology, AUC, Cairo University, NTRA, ITIDA, SRC, ASRT, the STDF, Intel, Mentor Graphics, MCIT, NSERC.

REFERENCES

[1] J. Mitola, "The software radio architecture," IEEE Communications Magazine, vol. 33, no. 5, pp. 26-38, May 1995.

[2] D. Dye, "Partial Reconfiguration of Xilinx FPGAs Using ISE Design Suite," WP374 (v1.2), May 30, 2012.

[3] H. Harada, Y. KAMIO, and M. FUJISE, "Multimode software radio system by parameter controlled and telecommunication component block embedded digital signal processing hardware," IEICE transactions on communications 83.6, pp. 1217-1228, 2000.

[4] Virtex-5 FPGA Configuration User Guide UG191 (v3.11) October 2012.

[5] E. Etto, "Xilinx XAPP290 Difference-Based Partial Reconfiguration (v2.0)," December 2007.

[6] Xilinx Partial Reconfiguration User Guide UG702 (v14.5) April 2012.

[7] P. Sedcole, B. Blodget, T. Becker, J. Anderson, P. Lysaght, "Modular dynamic reconfiguration in Virtex FPGAs," Computers and Digital Techniques, IEE Proceedings - , vol.153, no.3, pp.157,164, 2 May 2006.

[8] Xilinx Partial Reconfiguration Tools & Techniques trining course, "http://www.xilinx.com/training/fpga/fpga31000-ilt.pdf"

[9] K.A. Arun Kumar, "FPGA implementation of PSK modems using partial re-configuration for SDR and CR applications," India Conference (INDICON), 2012 Annual IEEE , vol. 205, no. 209, pp. 7-9, Dec. 2012.

[10] M. Hentati, A. Nafkha, Xun Zhang, P. Leray, J.-F. Nezan and M. Abid, "The study of the impact of architecture design on cognitive radio," Systems, Signals and Devices (SSD), 2011 8th International Multi-Conference, vol. 1, no. 4, pp. 22-25, March 2011.

[11] T. Janevski, "Traffic analysis and design of wireless IP networks," Artech House, 2003.

[12] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3G using WiFi," Proceedings of the 8th international conference on Mobile systems, applications, and services, ACM, pp 209-222, 2010.