ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 4, April 2014

A High-Speed Pipelined Design for CRC-8 ATM-HEC circuit and a Comparative Study with its Parallel and Serial counterparts

Avipsa S. Panda[1], Ganesh L.K. Moganti[2]

Student, School of Electronics, KIIT University, Bhubaneswar, India[1]

Assistant Professor, School of Electronics, KIIT University, Bhubaneswar, India[2]

Abstract: The Cyclic Redundancy Check (CRC) can be termed as a "sophisticated check sum". The Linear Feedback Shift Registers (LFSRs) play a key role in the calculation of the CRC. A simple circuit based on LFSR performs the CRC calculation by means of managing one bit at time. The CRC circuit design can be proposed by implementation of serial design or parallel or pipelining. This paper puts forth the comparison results of all the three design possibilities of CRC-8 ATM HEC.

Keywords: Cyclic Redundancy Check, Pipelining, CRC-8, Serial CRC, Parallel CRC.

I. INTRODUCTION

The cyclic redundancy check (CRC) is "an error-detecting code widely used in digital networks and storage devices to detect accidental changes to raw data" (Wikipedia). The name is such because of the operation, the data verification value, commonly termed as the 'check' is a 'redundancy' and the algorithm is based on 'cyclic codes'. CRC codes bring into play LFSRs to produce a signature based on the contents of any data passed through it. This signature can be used to sense the distortion of bits in the data stream. Common generator polynomials for CRC are:

- CRC-8 ATM-HEC : $x^8 + x^2 + x + 1$
- CRC-8 1-wire bus : $x^8 + x^7 + x^3 + x^2 + 1$
- AN bus : $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$
- RC-16 : $x^{16} + x^{12} + x^2 + 1$
- RC-32 (Ethernet) : $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

In general, all CRC codes are capable of detecting the following:

- All single and double-bit errors
- All burst errors of length less than the degree of the generator polynomial
- Most burst errors having length greater than the degree of the generator polynomial[1]

A. Paper Organization

The Section II deals with the basic conceptual details. Section III explains the implementation of serial, parallel and pipelined CRCs. Section IV brings forth the experimental analysis and results. Section V concludes the paper.

II. BACKGROUND

The CRC circuit can be designed by implementing either serial architecture or parallel architecture or by introduction of pipelining. Serial LFSRs can produce multi-bit arbitrary numbers, but these are highly correlated and fail many statistical tests.

Hence, the parallel designing approach is considered. Parallelism can be explained as "a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can be divided into smaller ones, which are then solved concurrently." Moreover, the power consumption in case of parallel architecture is less, owing to reduced switching activity.

Pipelining can be defined as "an implementation technique where multiple instructions are overlapped in execution." The pipelining is divided into different stages and each stage completes a part of instruction in parallel. Pipelining increases the throughput ("average rate of successful message delivery") over the communication channel. By pipelining, the critical path is reduced and the slack increases.

III. IMPLEMENTATION

This section consists of the circuit of the proposed design and the code (program), done using Xilinx Version 8.2i.

A. Serial CRC-8 ATM-HEC circuit design

In the serial architecture, the flip-flops are clocked every clock cycle and only one bit of information is generated per clock cycle. The output can be taken from input or output of any flip-flop.

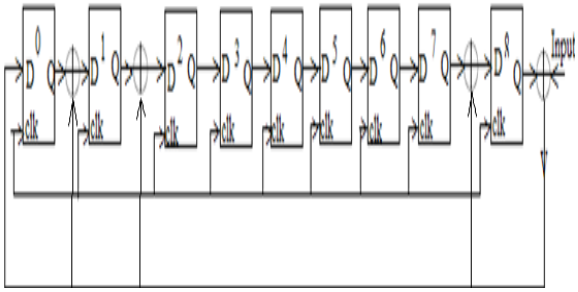

Fig. 1. This figure shows the serial CRC-8 ATM HEC circuit. In the figure, "clk" stands for clock given to the flip-flops, "D" and "Q" stand for the input and output of each flip-flop respectively.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 4, April 2014

A.1 Xilinx Code

```
module serial(q,i,clk,preset);
output reg [8:0]q;
input i,clk,preset;
reg y,x1,x2,x8;
always @(posedge clk)
begin
if(preset)
begin
q<=9'b101010101;
y<=1'b0;
x1<=1'b1;
x2<=1'b1;
x8<=1'b1;
end
else
begin
q[0]<=y;
x1<=q[0]^y;
q[1]<=x1;
x2<=q[1]^y;
q[2]<=x2;
q[3]<=q[2];
q[4]<=q[3];
q[5]<=q[4];
q[6]<=q[5];
q[7]<=q[6];
x8<=q[7]^y;
y<=q[8]^i;
end
end
endmodule
```

B. Parallel CRC-8 ATM HEC circuit design

In the parallel architecture, the output of *i*- successive cycles are generated in one cycle. Parallel architecture is often used with high-speed data, because in such cases, CRC calculation becomes quite difficult in a serial architectural domain. The randomization of the bits generated is more and the power consumption is less as compared to the serial architecture.

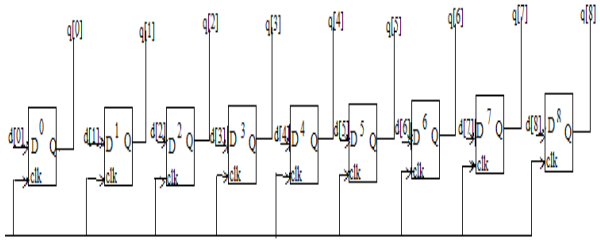

Fig. 2. This figure shows the parallel CRC-8 ATM HEC circuit. In the figure, d[i] stands for inputs and q[i] stands for the outputs of the flip-flops; where i= 0, 1, 2, … 8 and"clk" stands for clock given to the flip-flops.

B.1 Xilinx Code

```
module parallel(q,clk,preset,d);
output reg [8:0]q;
input [8:0]d;
input clk, preset;
always @(posedge clk)
begin
if(preset)
begin
q<=9'b101010101;
end
else
begin
q[0]<=d[0]^1;
q[1]<=d[1]^1;
q[2]<=d[2]^1;
q[3]<=d[3]^0;
q[4]<=d[4]^0;
q[5]<=d[5]^0;
q[6]<=d[6]^0;
q[7]<=d[7]^0;
q[8]<=d[8]^1;
end
end
endmodule
```

C. Pipelined CRC-8 ATM HEC circuit design

Pipelining reduces the "effective critical path" by bringing in latches along the "critical datapath". By doing so, either the clock frequency or the sampling speed is increased or power consumption is reduced at the same speed. Here, the pipelining is done by using the "look-ahead pipelining" algorithm, and the major objective is to reduce the iteration bound.

Iteration bound can be described as "the maximum of all loop bounds" and the loop bound for a particular loop can be defined as "t/w", where "t" is number of XORs in the loop and "w" is the no. of delay elements in the loop". [2]

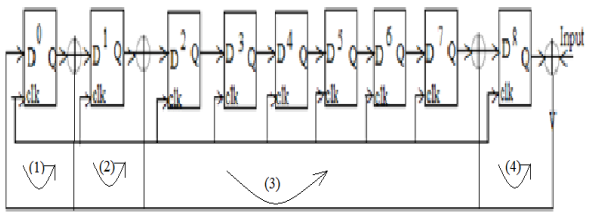

Fig. 3(a). This figure shows the loops for serial CRC-8 ATM HEC circuit. L_1 = 1/1 = 1; L_2=2/1 = 2; L_3 = 2/6 = 1/3; L_4 = 2/1 = 2 ; L_i is the loop bounds for the respective loops, i=1,2,3,4. Iteration bound = 2.

The aim for pipelining is to reduce the iteration bound of the above circuit, i.e., reduction of loop bounds for loops 2 and 4.

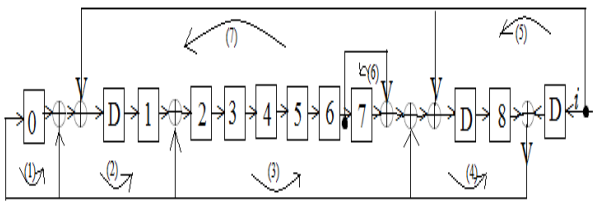

Fig. 3(b). This figure shows the pipelined circuit for CRC-8 ATM HEC. The loop bounds (L_i) are: L_1 = 1/1 =1 ; L_2 = 3/2 ; L_3 = 3/6 = 1 /2; L_4 = 3/2;L_5 = 2/3; L_6 = 1/1 = 1; L_7 = 5/8. Iteration bound = 3/2.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 4, April 2014

C.1 Xilinx Code

```
module pipelined(q, i, clk, seed, preset);
output reg [8:0]q;
input i, clk, seed, preset;
reg [3:1]d;
reg y,a;
reg [5:1]w;
always @(posedge clk)
begin
if(preset)
begin
q<-9'b101010101;
w<=5'b10101;
y<=1'b0;
d<=3'b010;
a<=1'b0;
end
else if(seed)
begin
d[3]<=i;
q[0]<=y;
w[1]<=q[0]^y;
w[2]<=w[1]^i;
d[1]<=w[2];
q[1]<=d[1];
a<=q[1]^y;
q[2]<=a;
q[3]<=q[2];
q[4]<=q[3];
q[5]<=q[4];
q[6]<=q[5];
q[7]<=q[6];
w[3]<=q[7]^q[6];
w[4]<=w[3]^y;
w[5]<=w[4]^i;
d[2]<=w[5];
q[8]<=d[2];
y<=q[8]^d[3];
end
else
begin
y<=q[8];
q[0]<=y;
w[1]<=q[0]^y;
w[2]<=w[1];
d[1]<=w[2];
q[1]<=d[1];
a<=q[1]^y;
q[2]<=a;
q[3]<=q[2];
q[4]<=q[3];
q[5]<=q[4];
q[6]<=q[5];
q[7]<=q[6];
w[3]<=q[7]^q[6];
w[4]<=w[3]^y;
w[5]<=w[4];
d[2]<=w[5];
q[8]<=d[2];
end
end, endmodule
```

IV. EXPERIMENTAL RESULTS

The results shown include the RTL view and Simulation of the circuit, done using Xilinx 8.2i; the layout done using Cadence Encounter Tool and the comparison table, for which the results have been obtained using Cadence RC Tool.

A. Serial CRC-8 ATM HEC circuit

Fig. 4(a). The above figure shows the RTL view of the circuit.

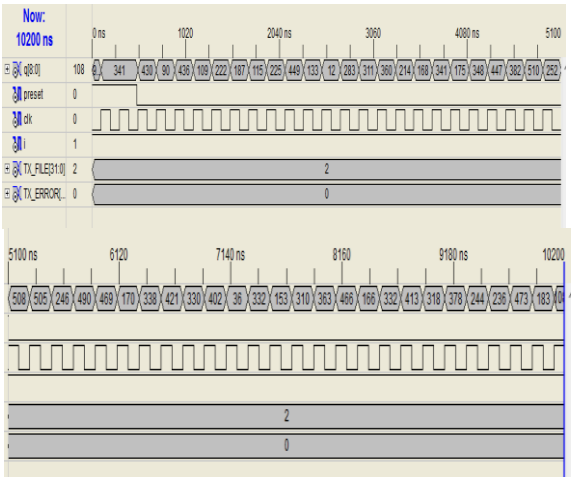

Fig. 4(b). The above figure shows the simulation of the circuit (Run Time: 10200ns).

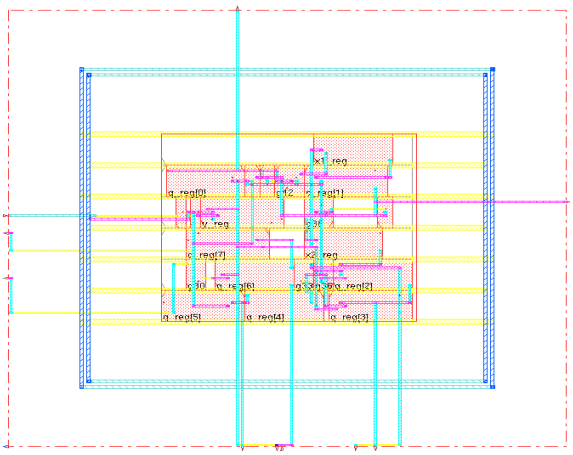

Fig. 4(c). The above figure shows the layout of the serial CRC-8 ATM HEC circuit.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 4, April 2014

B. Parallel CRC-8 ATM HEC

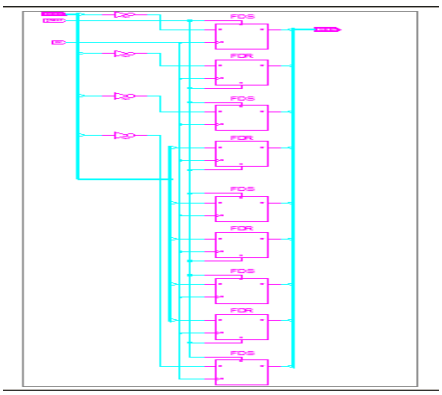

Fig. 5(a). The above figure shows the RTL view of the circuit.

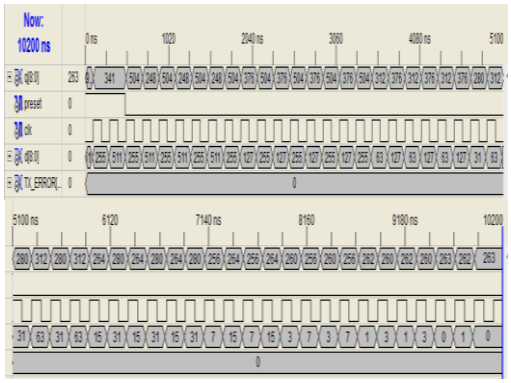

Fig. 5(b). The above figure shows the simulation of the circuit (Run Time: 10200ns).

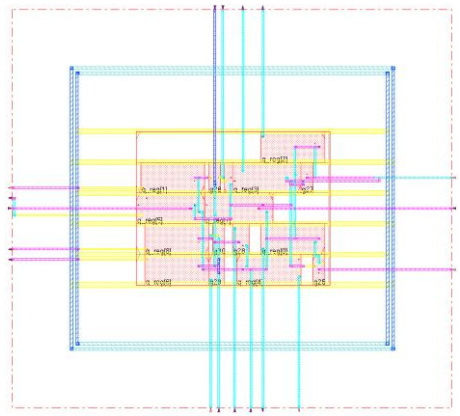

Fig. 5(c). The above figure shows the layout of the circuit.

C.Pipelined CRC-8 ATM HEC circuit

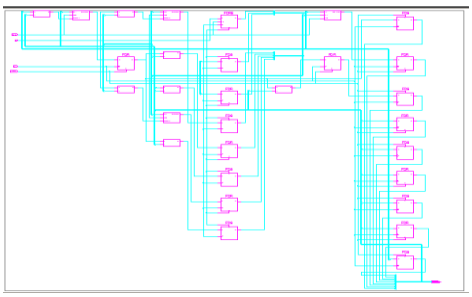

Fig. 6(a). The above figure shows the RTL view of the circuit.

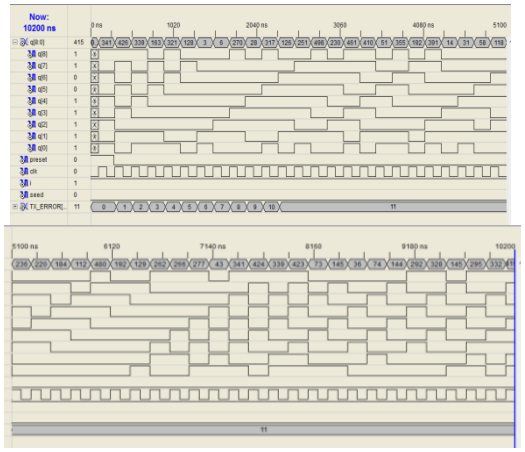

Fig. 6(b). The above figure shows the simulation of the circuit (Run Time: 10200ns).

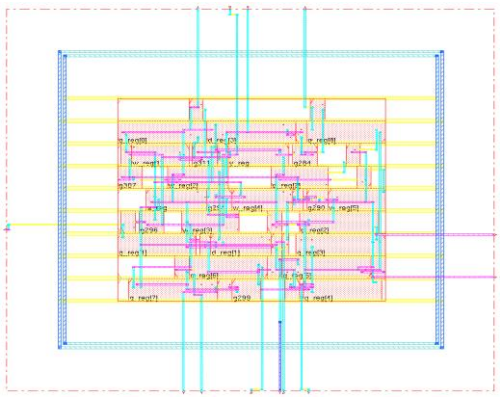

Fig. 6(c). The above figure shows the layout of the circuit.

TABLE I
RESULTS OF THE DIFFERENT ARCHITECTURES PROPOSED FOR THE CRC-8 ATM HEC CIRCUIT

Parameters	Serial Architecture	Parallel Architecture	Pipelined Architecture
Leakage Power (nW)	17.722	14.632	34.279
Timing Slack (ps)	4812	4842	7818
Area (No. of cells) (μm^2)	725 (21)	562 (16)	1360 (37)
Rise Slew| Fall Slew	80|57	28|29	70|52
Pre-CTS: Setup Time (all)	4.548	4.617	7.576
Post-CTS: Setup Time (all)	4.548	4.617	7.301
Post-CTS: Hold Time (all)	0.206	5.051	0.174
Post-Route: Setup Time (all)	4.552	4.614	7.290
Post-Route: Hold Time (all)	0.210	5.052	0.179

V. CONCLUSION

Though by using the pipelined architecture, we get high area and leakage power; but a reduced critical path, high timing slack, high setup time is obtained. The slew rate is higher than that obtained using a parallel architecture but is lower as compared to the serial architecture, which

indicates the pipelined circuit is faster as compared to its parallel counterpart, but is a bit slower than the serial architecture.

ACKNOWLEDGMENT

The authors would like to thank the Dean, School of Electronics, KIIT University, for providing proper laboratory facilities.

REFERENCES

[1] P. Harika, B.V.V. Satyanarayana, "FPGA Based high speed Cyclic Redundancy Check," IJERT, vol. 2, issue 3, Mar. 2013.

[2] S. Singh, S. Sujana, I. Babu, K. Latha, "VLSI Implementation of Parallel CRC using Pipelining, Unfolding and Retiming", IOSR Journal of VLSI and Signal Processing, vol. 2, issue 5, pp. 66-72, May 2013.

BIOGRAPHIES

Avipsa Sudarsan Panda is currently pursuing M.Tech. degree in VLSI and Embedded Systems from KIIT University, Odisha, India and has received a B.Tech. degree in Electronics and Electrical from KIIT University, Odisha, India.

Ganesh L.K. Moganti is presently working as Assistant Professor in KIIT University, Odisha, India. He has achieved his M.Tech. degree from MANIT, Bhopal, India and his B.Tech. degree in Electronics and Communication from Gudlavalleru Engineering College, Andhra Pradesh,India. He has keen interest in Testing of VLSI circuits and has successfully completed many projects on it.