Information encoding and decoding using Residue Number System for {2^{2n}-1, 2^{2n}, 2^{2n}+1} moduli sets

Idris Abiodun Aremu **Kazeem Alagbe Gbolagade**

Abstract- This paper presents the design methods of information encryption and decryption using Residue Number System (RNS). We selected length three moduli set $2^{2n} - 1, 2^{2n}, 2^{2n} + 1$, and design an effective forward conversion for the selected moduli set with 4n + 2 and 8n as delay and Area respectively for the information encryption and reverse converter for the same moduli set with 4n + 3 and 8n + 3 as area and Delay respectively, our proposed scheme out perform with the state of the art in terms of both security and computational efficiency

Index Terms – Decryption, forward conversion, information encryption and Residue Number System.

I. INTRODUCTION

Encryption and decryption of data has recently been widely investigated and developed because there is a demand for a stronger encryption and decryption which is very hard for intrusion. Over the centuries, information security has become a major issue. The Residue Number System (RNS) is a non-weighted number system that utilizes remainders to represent numbers play an important role in information security, RNS has received considerable attention in arithmetic computation and Digital Signal Processing (DSP) applications such as digital filtering, Fast Fourier Transform, Discrete Cosine Transform, etc. This is due to the following inherent properties of RNS: parallelism, modularity, fault tolerance, and carry-free operations [1], [2]. Moduli Selection and Data Conversion are the two most important issues for a successful RNS utilization. Data Conversion can be categorized into forward and reverse conversions. The forward conversion involves converting a binary or decimal number into its RNS equivalent while the reverse conversion is the inverse operation, i.e., it involves converting RNS number into binary or decimal. Relatively, reverse conversion is more complex.

II. BACKGROUND

A. Residue Number System (RNS)

RNS comprises a set of moduli which are independent of each other. An integer is represented by the residue of each of the modulus and arithmetic operations are based on residues individually. The advantage of using the RNS over the conversional system includes "carry-free" operation, fault tolerance, parallelism and

Manuscript received August, 2017

Idris Abiodun Aremu, Computer Science, School of Technology, Lagos State Polytechnics, Lagos, Nigeria, 2348025273062

Kazeem Alagbe Gbolagade,college of Information Communication Technology, Kwara State University, Ilorin, Nigeria, 2348109668798,

modularity. These inherent features make RNS to be widely used in Digital Signal Processing (DSP) applications such as digital filtering, convolution, fast Fourier transform and image processing [3], [4], [5].

Chinese Remainder Theorem (CRT):

$X = \left|\sum_{i=0}^{n} m_i \left|m_i^{-1}\right|_{m_i} x_i\right|_M$ Such that

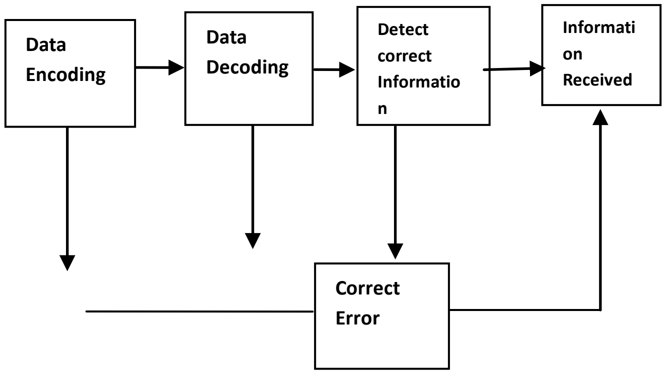
$M = \prod_{i=1}^{n} m_i \quad M_i = \frac{M}{m_i}$

M_i^{-1} is the multiplicative inverse m_i

B. *Fault Tolerance (FT)*

Fault-tolerance for error processing, can be invented in two stages, effective error processing stage, that intends to hide effective fault before occurrence of failure, and hidden error processing that aims to ensure that the faults will not be activated again [6].

Reactive fault-tolerance policy tries to reduce failures when they occur. It can be divided into error processing and fault-treatment techniques. The purpose of error processing is to eliminate errors from the calculation. Error treatment also aims to prevent the reactivation of errors [7, 8].



The rest of the paper is structured as follows. Section 2 provides a brief background on residue number system and fault tolerance. In Section 3, the hardware realization of the proposed system was discussed, and the corresponding algorithm is presented in the same section. Section 4 describes the performance evaluation of the proposed converter and evaluates and compares its performance. The paper is concluded in Section 5.

Proposed Algorithm

The proposed algorithm is described using the following theorems:-

Consider the computation of the residue of an arbitrary integer X with respect to a modulus M. since X is represented as 6n - bit binary number. $x_{6n-1}, x_{6n-2}, x_{6n-3} \ldots\ldots x_1x_0$ and its residue with respect to m may be expressed as

$$|X|_m = |x_{6n-1}x_{6n-2}x_{6n-3}\ldots\ldots\ldots\ldots x_1x_0|_m \quad (1)$$

Of which an equivalent expression is

$$|X|_m = |2^{6n-1}x_{6n-1} + 2^{6n-2}x_{6n-2} + \ldots + 2^0x_0|_m \quad (2)$$

Proof: It can be demonstrated that

Modulus $2^{2n} - 1$

$$|2^{2n}|_{2^{2n}-1} = |2^{2n} - 1 + 1|_{2^{2n}-1} = 1 \quad (3)$$

Modulus $2^{2n} + 1$

$$|2^{2n}|_{2^{2n}+1} = |2^{2n} + 1 - 1|_{2^{2n}+1} = -1 \quad (4)$$

Using three moduli set of $m_1 = 2^{2n} - 1, m_2 = 2^{2n}\ and\ m_3 = 2^{2n} + 1$ an any integer X with dynamic range M = ⌊0 , $2^{6n} - 2^{2n}$ ⌋ where the upper end of the range is m_1, m_2, m_3 is uniquely defined by a residue set r_1, r_2, r_3 where $r_1 = |X|_{m_i}$ and X is a 6n bits

$$|X|_m = |x_{6n-1}x_{6n-2}x_{6n-3} \ldots\ldots\ldots\ldots x_1x_0|_m \quad (5)$$

Residues are obtained by nominally dividing X bym_i. The residue r_2 is the easiest to compute because the n least significant bits constitute the remainder where x is divided by 2^{2n} hence r_2 is the number represented by the least significant 2n bit of x. These bits are obtained by nominally shifting to the right by n bits.

In order to determine the $r_1\ and\ r_3$ we first partition X into three 2n – bits blocks of B1, B2 and B3 respectively,

$$B1 = \sum_{j=4n}^{6n-1} x_j 2^{j-4n} \quad (6)$$

$$B2 = \sum_{j=2n}^{4n-1} x_j 2^{j-2n} \quad (7)$$

$$B3 = \sum_{j=0}^{2n-1} x_j 2^{2n} \quad (8)$$

From addition of equation (6), (7), and (8) respectively produce equation 9 as follows

$$X = B12^{4n} + B22^{2n} + B3 \quad (9)$$

The residue r_1 is then obtained as follows

$$r_1 = |X|_{2^{2n}-1} \quad (10)$$

$$r_1 = |B_1 2^{4n} + B_2 2^{2n} + B_3|_{2^{2n}-1} (11)$$

$$r_1 = ||B_1 2^{4n}|_{2^{2n}-1} + |B_2 2^{2n}|_{2^{2n}-1} + |B_3|_{2^{2n}-1}|_{2^{2n}-1} \quad (12)$$

$$|B_1 2^{4n}|_{2^{2n}-1} = |B_1 2^{2n} \ . 2^{2n}|_{2^{2n}-1} \ (13)$$

$$= |B_1 .1.1|_{2^{2n}-1} = |B_1|_{2^{2n}-1} \quad (14)$$

$$|B_2 2^{2n}|_{2^{2n}-1} = |B_2|_{2^{2n}-1} \quad (15)$$

$$r_1 = |B_1 + B_2 + B_3|_{2^{2n}-1} \quad (16)$$

The residue r_3 is then obtained as follows

$$r_3 = |X|_{2^{2n}+1} \quad (17)$$

$$r_3 = |B_1 2^{4n} + B_2 2^{2n} + B_3|_{2^{2n}+1} \quad (18)$$

$$r_3 \ ||B_1 2^{4n}||_{2^{2n}+1} + ||B_2 2^{2n}|_{2^{2n}+1} + |B_3|_{2^{2n}+1}|_{2^{2n}+1} \quad (19)$$

$$|B_1 2^{4n}|_{2^{2n}+1} = |B_1 . 2^{2n} + 1 - 1 \ 2^{2n} + 1 - 1|_{2^{2n}+1}$$

$$= |B_1|_{2^{2n}+1} \quad (20)$$

$$|B_2 2^{2n}|_{2^{2n}+1} = |B_2 2^{2n} + 1 - 1|_{2^{2n}+1}$$

$$= |-B_2|_{2^{2n}+1} \quad (21)$$

$$r_3 = |B_1 - B_2 + B_3|_{2^{2n}+1} \quad (22)$$

Using the result of r_1 *and* r_3 *respectively* that is equation (16) and (22) the hardware structure of the propose forward converter can be obtained (fig.1)

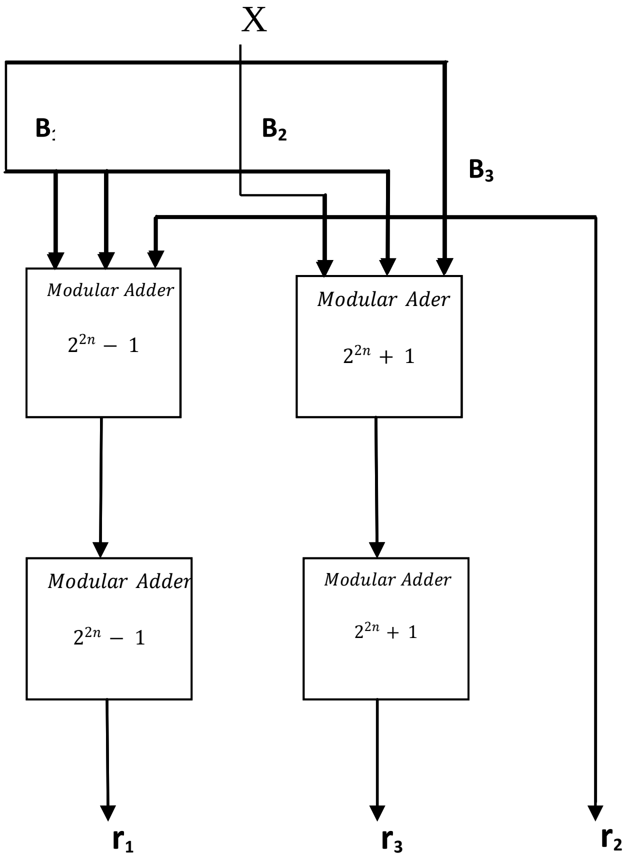


Fig. 1. Block diagram for the proposed forward converter

Hardware Design

Delay

2n -1 + 2n -1 and 2n + 1 + 2n +1

= 4n – 2 and 4n + 2

Delay = 4n + 2

Area

4n – 2 + 4n + 2

= 8n

Example 1. Given that X = 22 and take n =1 using the moduli sets $\{m_1 = 2^{2n} - 1, m_2 = 2^{2n}\ and\ m_3 = 2^{2n} + 1\}$ from equation (23) $\left\lfloor \frac{x}{m_2} \right\rfloor = |2.5.1 - 4.2 - 2.3.2|_{15} = |110|_{15} = 5$ and from (22) $X = 5.4 + 2 = 22$

$$= |u_1 - u_2 - u_3|_{m_1 m_3}$$

$$x_1 = \underbrace{x_{1,2n-2} x_{1,2n-3} x_{1,2n-4} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ x_{11} x_{00}}_{2n-1} \quad (24)$$

$$x_2 = \underbrace{x_{2,2n-1} x_{2,2n-2} x_{2,2n-3} \ \ldots\ldots\ x_{21} x_{20}}_{2n} \quad (25)$$

$$x_3 = \underbrace{x_{3,2n} x_{3,2n-1} x_{3,2n-2} \ldots\ldots x_{3,1}}_{2n+1} x_{1,0} \quad (26)$$

$$\boldsymbol{u_1 = \left|2^{2n-1} m_3 x_1\right|_{m_1 m_3}}$$
$$\boldsymbol{= \left|2^{2n-1} m_3 x_1\right|_{m_1 m_3}}$$

$$\boldsymbol{u_1} = \underbrace{11 \ldots .11}_{4n\ \text{bits}}\ \underbrace{x_{1,2n-2} x_{1,2n-3} x_{1,2n-4}\ \ldots\ldots\ldots x_{11} x_{10}}_{2n-1\ \text{bits}}$$

$$\boldsymbol{u_2 = |-m_2 x_2|_{m_1 m_3}}$$

$$\boldsymbol{u_2} = \underbrace{11\ldots.11}_{\text{2n bits}}\ \underbrace{x_{2,2n-1}x_{2,2n-2}x_{2,2n-3}\ \ldots\ldots x_{21}x_{20}}_{\text{2n bits}}$$

$$\boldsymbol{u_2} = \underbrace{\overline{\mathbf{00}}\ldots\overline{\mathbf{00}}}_{\text{2n bits}}\ \underbrace{\overline{\boldsymbol{x}}_{\mathbf{2,2n-1}}\overline{\boldsymbol{x}}_{\mathbf{2,2n-2}}\overline{\boldsymbol{x}}_{\mathbf{2,2n-3}}\ \ldots\ldots\overline{\boldsymbol{x}}_{\mathbf{21}}\overline{\boldsymbol{x}}_{\mathbf{20}}}_{\text{2n bits}}$$

$$\boldsymbol{u_3} = \underbrace{11\ldots.11}_{\text{4n bits}}\ x_{3,2n}\underbrace{x_{3,2n-1}x_{3,2n-2}\ \ldots\ldots\ldots\ldots x_{31}x_{30}}_{2n+1}$$

$$\overline{\boldsymbol{u_3}} = \underbrace{\overline{\mathbf{00}}\ldots00}_{\text{4n bits}}\ \underbrace{x_{2,2n-1}x_{2,2n-2}x_{2,2n-3}\ \ldots\ldots\overline{x}_{21}x_{20}}_{2n-1}$$

Here

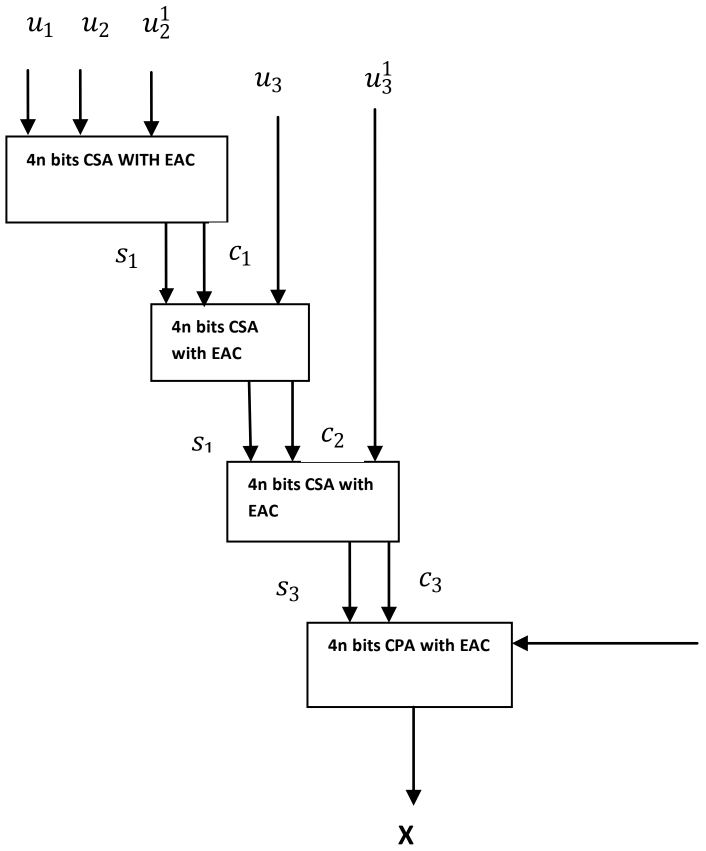


Fig. 2 Block diagram for proposed Reverse converter

HARDWARE REALISATION

Using the result of $r_1\ and\ r_3\ respectively$ that is equation (16) and (22) the hardware structure of the propose forward converter can be obtained (fig.1)

Fig. 1. Block diagram for the proposed forward converter

Hardware Design

Delay

2n -1 + 2n -1 and 2n + 1 + 2n +1

= 4n – 2 and 4n + 2

Delay = 4n + 2

Area

4n – 2 + 4n + 2

= 8n

Theorem 1: Given the following moduli set the following hold true $\{2^{2n} - 1, 2^{2n}, 2^{2n} + 1\}$

The proposed algorithm is described using the following theorems:-

Given that $m_1 = 2^{2n} - 1, m_2 = 2^{2n}\ and\ m_3 = 2^{2n} + 1$

$$m_3^{-1} = |m_1 m_2|_{m_3} = -2^{2n-1} \qquad (9)$$

$$m_2^{-1} = |m_1 m_3|_{m_2} = -1 \qquad (10)$$

]

$m_1^{-1} = |m_2 m_3|_{m_1} = 2^{2n-1}$ (11)

Proof: It can be easily shown below that equation (9), (10), and (11) which are $-2^{2n-1}, \quad -1, \ and \ 2^{2n-1}$ are the multiplicative inverse is of $m_1 m_2 m_3$

$$m_3^{-1} = |m_1 m_2|_{m_3} = -2^{2n-1}$$

$$|2^{2n} - 1\ x\ 2^{2n}|_{2^{2n}+1} = |2^{2n} + 1 - 1 - 1x\ 2^{2n} + 1 - 1|_{2^{2n}+1}$$

$$= |2|_{2^{2n}+1}$$

$$= |-2^{2n-1} x 2|_{2^{2n}+1} = 1$$

$$m_2^{-1} = |m_1 m_3|_{m_2} = -1$$

$$|2^{2n} - 1\ x 2^{2n} + 1|_{2^{2n}} = -1$$

$$m_1^{-1} = |m_2 m_3|_{m_1} = 2^{2n-1}$$

$$|2^{2n}\ x 2^{2n} + 1|_{2^{2n}-1} = |2^{2n} + 1 - 1\ x\ 2^{2n} - 1 + 1 + 1|_{2^{2n}-1}$$

$$|2\ x\ 2^{2n}|_{2^{2n}-1} = |2^{2n}|_{2^{2n}-1} = |1|_{2^{2n}-1}$$

Theorem 2:

The proposed algorithm is described using the following theorems:-

Given that $m_1 = 2^{2n} - 1, m_2 = 2^{2n}\ and\ m_3 = 2^{2n} + 1$ with respect to $r_1 r_2\ and\ r_3$ respectively, the decimal equivalent X is given as

$$X = m_2 \left\lfloor \frac{x}{m_2} \right\rfloor + x_2 \qquad (12)$$

Proof

It can be easily shown that

$$X = \left| \sum_{i=0}^{n} m_i \left| m_i^{-1} \right|_{m_i} x_i \right|_M \qquad (13)$$

Such that $M = \prod_{i=1}^{n} m_i$

$$M_i = \frac{M}{m_i}$$

M_i^{-1} is the multiplicative inverse of m_i

$$X = \left| \sum_{i=0}^{n} m_i \left| m_i^{-1} \right|_{m_i} x_i \right|_M \qquad (14)$$

$$X = |m_2 m_3 . -2^{2n-1} x_1 + m_1 m_3 . -1 . x_2 + m_1 m_2 2^{2n-1} . x_3|_{m_1 m_2 m_3} \qquad (15)$$

$$X = |m_2 m_3 . -2^{2n-1} x_1 + (m_2^2 - 1) . x_2 + m_1 m_2 2^{2n-1} . x_3|_{m_1 m_2 m_3} \qquad (16)$$

$$X = |m_2 m_3 . -2^{2n-1} x_1 + m_2^2 x_2 + x_2 + m_1 m_2 2^{2n-1} . x_3|_{m_1 m_2 m_3} \qquad (17)$$

Divide all through by m_2 and compute their floor value we have

$$\left\lfloor \frac{x}{m_2} \right\rfloor = |-2^{2n-1} m_3 x_1 - m_2 x_2 + 2^{2n-1} m_1 x_3|_{m_1 m_3} \qquad (18)$$

From

$$X = |m_2 m_3 |m_1^{-1}|_{m_1} x_1 + m_1\ m_3\ |m_2^{-1}|_{m_2}\ x_2 + m_1 m_2\ |m_3^{-1}|_{m_1 m_2 m_3} \qquad (19)$$

$$X = |m_2 m_3 . -2^{2n-1} x_1 + m_2^2 x_2 + x_2 + m_1 m_2 2^{2n-1} . x_3|_{m_1 m_2 m_3} \qquad (20)$$

$$\left\lfloor \frac{x}{m_2} \right\rfloor = |-2^{2n-1} m_3 x_1 - m_2 x_2 + 2^{2n-1} m_1 x_3|_{m_1 m_3} \qquad (21)$$

$$X = m_2 \left\lfloor \frac{x}{m_2} \right\rfloor + x_2 \qquad (22)$$

$$\left\lfloor \frac{x}{m_2} \right\rfloor = |2^{2n-1} m_3 x_1 - m_2 x_2 - 2^{2n-1} m_1 x_3|_{m_1 m_3} \qquad (23)$$

Example 1. Given that X = 22 and take n =1 using the moduli sets $\{m_1 = 2^{2n} - 1, m_2 = 2^{2n}\ and\ m_3 = 2^{2n} + 1\}$ from equation (23) $\left\lfloor \frac{x}{m_2} \right\rfloor = |2.5.1 - 4.2 - 2.3.2|_{15} = |110|_{15} = 5$ and from (22) $X = 5.4 + 2 = 22$

III HARDWARE REALISATION

The hardware realisation of the forward conversion for the moduli set $\{2^{2n} - 1, 2^{2n}, 2^{2n} + 1\}$ is based on (16) and (22) is achieved by using simple fast adders like the Carry Save Adder (CSA) for three bits addition and Carry Propagate Adder (CPA) for two bits addition as shown in the diagram in fig. 1 above, the reserve conversion was also achieved with fast adder like carry save adder (CSA) and carry propagation adder as shown in fig. 2 above.

IV PERFORMANCE EVALUATION

The performances of the proposed converters are evaluated in terms of area cost and conversion delay, the forward and reverse converter is evaluated theoretically in term of conversion time and area cost. The hardware utilization of the proposed converter is computed using the full adder and is compared with equivalent best known state of the art forward and reverse converter, the proposed converter required a delay of 4n + 2 and 8n +3 for both forward and reverse converter respectively, while the area cost is 8n and 4n + 3. Total delay for forward and reserve conversion is 12n + 5 and the total cost for both forward and reverse conversion is 12n + 3.

V CONCLUSION

Typically data encryption is the process of encoding information from sender site to the receiver site by decoding the received encrypted data/information. In this paper, three moduli set $\{2^{2n} - 1, 2^{2n}, 2^{2n} + 1\}$ are used to design an effective forward conversion for the selected moduli set with 4n + 2 and 8n as delay and Area respectively for the information encryption and reverse converter for the same moduli set with 4n + 3 and 8n + 3 as area and Delay respectively, our proposed scheme out perform with the state of the art in terms of both security and computational efficiency

REFERENCES

[1] Wang, Y., Song, X., Aboulhamid, M., & Shen, H. (2002). Adder based residue to binary number converters for (2/sup n/-1, 2/sup n/, 2/sup n/+ 1). *IEEE Transactions on Signal Processing*, *50*(7), 1772-1779.

[2] Wang, Y. (2000). Residue-to-binary converters based on new Chinese remainder theorems. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *47*(3), 197-205.

[3] Gbolagade, K. A., & Cotofana, S. D. (2009, August). Residue-to-decimal converters for moduli set with common factors. In *Circuits and Systems, 2009. MWSCAS'09. 52nd IEEE International Midwest Symposium on* (pp. 624-627). IEEE.

[4] Gbolagade, K. A., Voicu, G. R., & Cotofana, S. D. (2011). An Efficient FPGA Design of Residue-to-Binary Converter for the Moduli Set {2n+ 1, 2n, 2n-1\} $. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *19*(8), 1500-1503.

[5]Siewobr, H., & Gbolagade, K. A. (2014). Modulo Operation Free Reverse Conversion in the {2 (2n+ 1)-1, 2n, 22n-1} Moduli Set. *International Journal of Computer Applications*, *85*(18).

[6] Avizienis, A., & Laprie, J. C. (2000). LAAS-CNRS 7, avenue du Colonel Roche 31077 Toulouse (France) Brian Randell Dept. of Computing Science.

[7] Raicu, I., Foster, I. T., & Beckman, P. (2011, June). Making a case for distributed file systems at exascale. In *Proceedings of the third international workshop on Large-scale system and application performance* (pp. 11-18). ACM.

[8] Mathur, P., & Nishchal, N. (2010, October). Cloud computing: New challenge to the entire computer industry. In *Parallel Distributed and Grid Computing (PDGC), 2010 1st International Conference on* (pp. 223-228). IEEE.