International Journal of Pure and Applied Mathematics
Volume 120 No. 5 2018, 815-825
ISSN: 1314-3395 (on-line version)
url: http://www.acadpubl.eu/hub/
Special Issue

http://www.acadpubl.eu/hub/

Design and Implementation of Fault Tolerant Memory using Linear Block Codes

T Anil[1], N Ratna Deepthika[2], K.B. Madhavi[3], G Bhavana[4]

[1]Dept. of Electronics and Communication Engineering, Holy Mary Institute of Technology and Science, Hyderabad, India.
[2]Dept. of Electronics and Communication Engineering, Mallareddy Engineering College, Hyderabad, India.
[3]Dept. of Electronics and Communication Engineering, St' Peters Engineering College, Hyderabad, India.
[4]Dept. of Electronics and Communication Engineering, Institute of Aeronautical Engineering, Hyderabad, India
E-Mail: topaiahgari.anil@gmail.com[1], ratnanimmathota@gmail.com[2], mdhvmdhv@gmail.com[3]

Abstract: The recent developments in the semiconductor industry need headed in the improvement of All the more intricate parts What's more systems' architectures by permitting creation methods to put a higher number for transistors for every range of the silicon kick the bucket. Hence the manufacturing procedure needs aid less and less dependable. In we compelling reason to raise frameworks that will recognize the presence from claiming faults Furthermore fuse systems to endure these faults same time still delivering a adequate level about administration. We present another approach should outline flaw line tolerant memory utilizing straight square codes. Hamming codes would the sort about straight piece codes to lapse identification Also revision. Hamming codes belongs to the class of block code which are coded to work on a block of bits rather than individual bits of data. It takes a block of k input bits and produce n bits of code word. The encoder takes k input bits and produce k bits of code word. The encoder is designed through the usual generator matrix multiplication while in the decoder the code word is decoded using syndrome method. Proposed method is validated using the Verilog HDL test environment and the results showed that the reliability of the system increased with a little area overhead.

Keywords: Encode, Error correction, Fault detection, Serial one step MLD, Majority logic decoder/detector, Hamming code

1. Introduction

The problem of reliable computing is as old as the first computers, which employed relays, vacuum tubes, delay line and cathode-ray tube storage, and other relatively unreliable components. Semiconductor components and magnetic core storage elements with much greater reliability were introduced in the second computer generation, and the reliability problem was significantly alleviated. However, much greater reliability requirements were posed by the space program in the early 1960's, as well as by other real-time applications in which human lives could be jeopardized by computer failure. In addition to improvements in component reliability and in test methods, redundancy at various levels of system organization was investigated and employed to increase the probability of uninterrupted correct operation over specified time intervals. The concept of fault tolerance unifies various approaches to reliability assurance by means of testing, diagnosis, and redundancy in machine organization and operation. It emerged in the late 1960's and reached maturity with the formation of the IEEE Computer Society Technical Committee on Fault-Tolerant Computing in 1969 and the subsequent holding of the First International Symposium on Fault-Tolerant Computing in 1971. The Symposium has been held annually since then, and it has become the major

international forum for the discussion of current experience and new ideas in system design, redundancy techniques, system modelling and analysis, testing and diagnosis methods, and other related areas.The vast majority broadly acknowledged definition of a fault-tolerant registering framework is that it may be an arrangement which need those inherent proficiency (without outer assistance) should preserve the begun and Johnson had proceeded right execution about its projects and input/output (I/O) capacities in the vicinity of a sure situated from claiming operational faults. A operational flaw line may be a unspecified (failure-induced) transform in the esteem from claiming one alternately that's only the tip of the iceberg rationale variables in the fittings of the framework. It will be the quick outcome of a physical disappointment occasion.

That occasion might a chance to be An lasting part failure, An Brief alternately irregular part malfunction, or remotely beginning obstruction with those operation of the framework. "Correct execution" implies that the programs, those data, and the outcomes don't hold errors, and that the execution the long haul doesn't surpass a specified farthest point. An lapse will be those manifestation of a fault, i. E. , it is those progress On an direction book alternately a information statement which will be initiated Eventually Tom's perusing those vicinity of a flaw line In those perspective of the physical disappointment off chance. Those over definition of a fault-tolerant framework may be In light of those introduce that those framework will capacity effectively At operational faults are not present; that is, it postulates that configuration faults have been wiped out from the framework former to its use. That's only the tip of the iceberg general elucidation from claiming issue tolerance also incorporates the capacity on enduring thus-far-undetected configuration faults. Configuration faults happen both in equipment and over programming of the framework. They would the results about either:.An incomplete or incorrect specification.

1) An incorrect translation of the specification into the final design-the assemblies of components or the sequences of machine language instructions and sets of data words.

In place should endure left-over configuration faults, those framework necessities procurements will distinguish their vicinity Furthermore to deactivate the influenced equipment or product same time system execution precedes utilizing the remaining parts of the framework.

2. Motivation

Two distinct approaches have been used in order to reach the goal of reliable computing. These approaches are applicable to all parts of the computing system, including its hardware elements, micro-programs, system programs, data bases, and user programs.

In the conventional methodology those unwavering quality of registering is guaranteed Eventually Tom's perusing An from the earlier disposal of the reason for faults. This disposal takes spot in front of standard utilize begins, and the assets that need aid allocated should accomplish dependability are used looking into perfecting those framework former to its field utilization. Excess may be not employed, and the sum parts of the framework must work effectively in the least times. Since in practice it has not been possible to assure the complete a priori elimination of all causes of unreliability, the goal of fault intolerance is to reduce the unreliability or the unavailability of the system to an acceptably low value. To supplement this approach, manual maintenance procedures are devised which return the system to an operating condition after a failure. The cost of providing maintenance personnel and the cost of the disruption and delay of computing also are parts of the overall cost of using the fault-intolerance approach. The procedures which have led to the attainment of reliable systems using this approach are acquisition of the most reliable components within the given cost and performance constraints, use of thoroughly refined techniques for the interconnection of components and assembly of subsystems,

packaging of the hardware to screen out expected forms of interference and 1305 carrying out of comprehensive testing to eliminate hardware and software design faults.

Once the design has been completed, a quantitative prediction of system reliability is made using known or predicted failure rates for the components and interconnections. In a “purely" fault-intolerant (i.e., no redundant) design, the probability of fault-free hardware operation is equated to the probability of correct program execution. Such a design is characterized by the decision to invest all the reliability resources into high-reliability components and refinement of assembly, packaging, and testing techniques. Occasional system failures are accepted as a necessary evil, and manual maintenance is provided for their correction. To facilitate maintenance, some built-in error detection, diagnosis, and retry techniques are provided. This is the current practice in computer system design the trend is toward an increasing number of built-in aids for the maintenance engineer.

3. Problem of Design Faults

In order to attain the goal of general fault-tolerance it is necessary that one of two conditions should be satisfied with respect to design faults:

1) The hardware and software should be free of design faults prior to the start of operational use.

2) The system should contain complete provisions to detect and to circumvent the effects of hardware and software design faults during operational use.

The principal tools for the a priori elimination of design faults are design verification programs for hardware and an extensive repertoire of validation and verification tools for software. These methods correspond to the fault-intolerance approach of perfecting hardware components in order to avoid operational faults.

4. Literature Survey

4.1 Error Detection and Correction

Lapse identification Furthermore revision codes strategy may be generally used should relieve absolute occasion upset on a incorporated information preparing What's more require additional equipment. All things considered this system provides for determinedly deficiency scope. Lapse identification What's more revision codes might make actualized in two ways which relies on transmission information. Whether transmission happens clinched alongside special case bearing will beat error, the slip control framework meets expectations out through the ahead lapse revision. Therefore, this strategy will be functional for a satellite transmission Furthermore it will be represented to us.

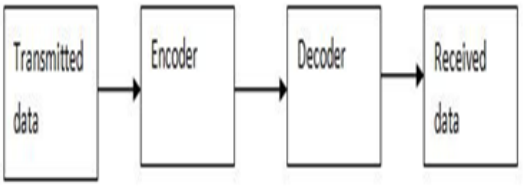

Fig 2.1:Scheme of a forward error correction.

4.2 Residue Code

Those equipment usage of the 32-bit shortcoming tolerant math rationale Unit, have utilized the deposit code and duplication from claiming fittings component so as on attain an less fittings overhead. On deposit codes, information parts Furthermore weigh parts would differentiate to have the capacity will recognize the errors. Likewise to Boolean operations of the ALU duplication for equipment with examination need been used to recognize those slip. Since the deposit code camwood best identify the error, they must devise

different extra ALU should settle on lapse revision workable. On such technique, though person lapse need happened to ALU, then it ought replaceability the unique ALU with the save.

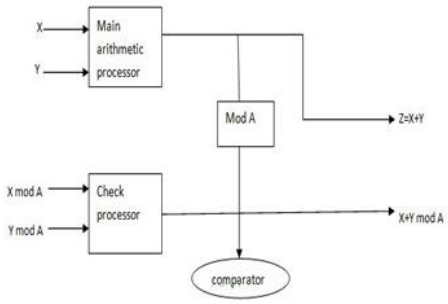

Fig 2.2:Arithmetic processor with residue checker

5. Design Methodology

5.1 Fault Tolerant

In the outline for fault-tolerant systems, the creator must Think as of the could be allowed event for a few diverse sorts about faults for example, such that transient faults, irregular faults, permanency faults, legitimate faults, and uncertain faults. Transient faults, regularly brought about by outer disturbances, exist to a limited period for the long run furthermore is nonrecurring. Irregular faults happen occasionally furthermore regularly aftereffect from flimsy gadget operation. Changeless faults are culprit and might a chance to be initiated Toward physical harm or outline errors. Legitimate faults happen when inputs or outputs about rationale entryways need aid stuck-at-0 or stuck-at-1. Uncertain faults happen The point when inputs alternately outputs of rationale entryways coast the middle of rationale 0 Furthermore rationale 1.

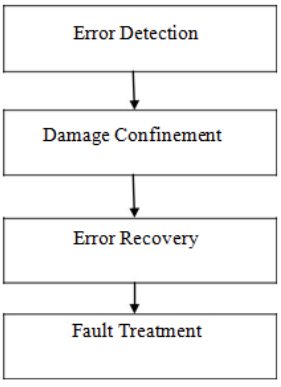

Fig 3.1:General fault tolerant procedure

An arrangement might work effectively in the vicinity of the previously stated faults though that suitable type from claiming excess is consolidated under the framework. Two major fault-tolerant configuration methodologies are static what more element excess is. Static excess is the utilization about excess segments something like that that faults might be veiled. Progressive excess may be the revamp of a framework so that the capacities of a faun unit would exchange to different utilitarian units. Four particular sorts for excess would majority of the data redundancy, the long run redundancy, programming

redundancy, and fittings excess. Data excess is the utilization of lapse identifying or slip correcting codes to majority of the data representational..

5.2 Hamming Codes

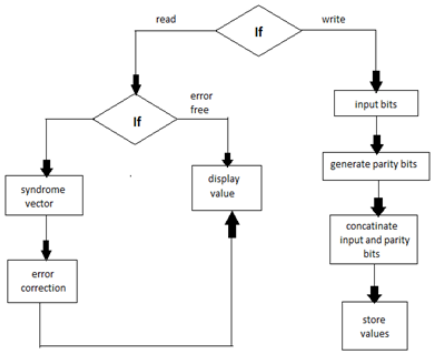

Fig 3.2: Algorithm for hamming codes

Coding is a powerful technique which helps us avoid unwanted information changes during data storage. Hamming codes are an important family of linear codes. The hamming codes consist of hamming matrix H which is used for encoding and decoding. During write operation, the parity bits are generated for a particular input data bits, the code word is generated for input bits and stored in the particular location of RAM. During read operation, the data is read from a particular location of RAM.

The syndrome vector is computed; therefore if syndrome is zero there no errors. The error position is found by nonzero syndrome vector and the particular bit location is corrected and the correct stored value is displayed.

Hamming codes are an important family of linear codes. They are named after Richard Hamming, who developed the first single-error correcting Hamming code and its extended version, single-error correcting double-error detecting Hamming code, in the early 1950s. These codes remain important even today.

Data			Codeword						
d_0	d_1	d_2	c_0	c_1	c_2	c_3	c_4	c_5	c_6
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	0	1	1
0	1	0	0	1	0	1	1	0	1
0	1	1	0	1	1	0	1	1	0
1	0	0	1	0	0	1	1	1	0
1	0	1	1	0	1	0	1	0	1
1	1	0	1	1	0	0	0	1	1
1	1	1	1	1	1	1	0	0	0

Table 3.1: linear code for (7, 3)

An (n, k)linear code is a Hamming code if its parity check matrix contains 2n−k−1 columns representing all possible nonzero binary vectors of length n − k. Here is an example of the parity check matrix of a (7, 4) Hamming code:

The information rate of hamming codes is (k/n). Since the parity check matrix of a Hamming code has (2n−k – 1) columns and the dimension of a parity check matrix is given a
(n − k)× n, the data length k and the code length n of a Hamming code are related as follows

$n = 2n-k-1$ (3.1)

Hamming codes exist only for those pairs (n, k) which satisfy Eq. 3.1. For any $r > 2$, Eq. 3.1 has a solution for $k = 2r-r-1$ and $n = 2r-1$. Table 3.1 shows examples of data lengths for which a Hamming code exists. It also shows the information rate of the resulting code. The information rate of Hamming codes is the highest possible for codes with code distance 3 and code length $2r-1$.

6. Implementation

Random-Access Memory (RAM) is a form of computer data storage which stores data and machine code currently being used. A Random-Access Memory device allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory. In today's technology, random-access memory takes the form of circuits. Therefore RAM's are fabricated using semiconductor devices mainly transistors.

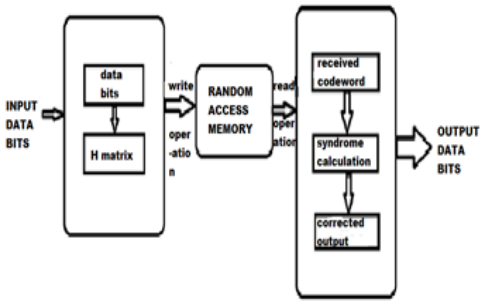

Fig 4.1: Block Diagram

Hamming codes are widely used for error correction in Dynamic random access memory (DRAMs). As we mentioned before, DRAMs are susceptible to random bit flips caused by alpha particles from the device packaging, background radiation, etc. A typical fault rate is one single-bit flip per day per gigabyte of DRAM. A memory protected by a Hamming code looks similar to the one in Fig. 5.3 except that more than one parity check bit is generated for each data word. Encoding is performed on complete data words, rather than individual bytes. The check bits are computed by parity generators which are designed according to the generator matrix of the code.

For instance, for a (7, 4) Hamming code with the generator matrix, three parity generators implement the equations $p0 = d0 \oplus d2 \oplus d3$, $p1 = d0 \oplus d2 \oplus d3$ and $p2 = d1 \oplus d2 \oplus d3$. A parity generator is usually realized as a tree of XOR gates.

6.1 Write Operation

Step1: input data bits are given. Let it be min which is of four bits.

Step2: parity bits are generated according to the given input data bits. Let the parity bits q of three bits. The parity bits are generated by the xor operation of input data bits.

q [1] = min[1]^min[2]^min[3]
q [2] = min[1]^min[2]^min[4]
q [3] = min[1]^min[3]^min[4]

Step3: the code word is formed by the concatenation of input data bits and parity bits. The code word formed is of 7 bits.

C={min,q}

Step4: the code word is stored in certain memory location of RAM.

6.2 Read Operation

Step1: the data (d) is read from a particular memory location of RAM which is of 7 bits.

Step2: the syndrome vector is calculated. The syndrome vector is the xor operation of data bits read from RAM and it is 3 bits.

s[1]= d[1]^d[2]^d[3]^d[5]
s[2]= d[2]^d[3]^d[4]^d[6]
s[3]= d[1]^d[2]^d[4]^d[7]

Step3: the syndrome vector is zero than there is no error. Go to step7.

Step4: if the syndrome is not zero and the error position is found go to step5.

Step5: the error vector which of 7 bits with "1" in error bit position and remaining other bits are zero. Ex: if error is in first bit e=1000000.

Step6: the error vector and data read from RAM location are xored. Therefore the corrected code word (cc) is obtained.

cc=e^d

Step7: the data bits are displayed.

7. Simulation Results

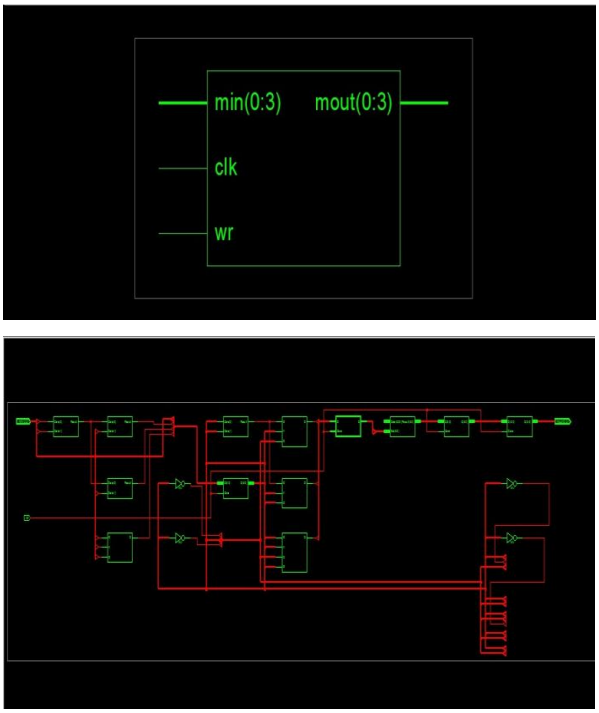

Fig 5.1.1:RTL Schematic of memory with fault tolerance

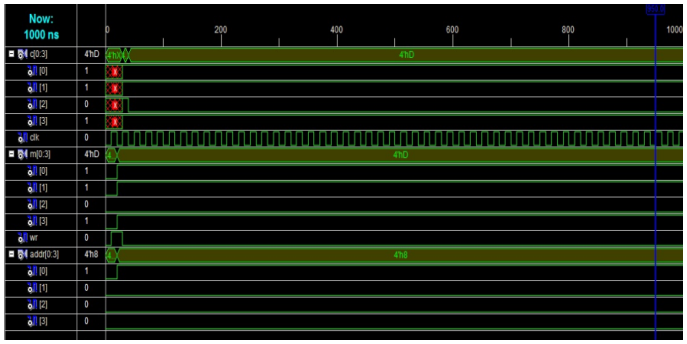

Fig 5.1.2: Simulation Results of memory with fault tolerance

invoked error, out=1111010
syndrome vector,{s[0],s[1],s[2]}=101
correctedoutput[x]=1101010
errorless message[mout]=1101
din=1101010
ram[0]=1101010

7.1 Final Results

RTL Top Level Output File Name	: memory.ngr
Top Level Output File Name	: memory
Output Format	: NGC
Optimization Goal	: Speed
Keep Hierarchy	: YES
Target Technology	: Automotive 9500XL
Macro Preserve	: YES
XOR Preserve	: YES
Clock Enable	: YES

7.2 Design Statistics

# IOs	: 10
Cell Usage	
# BELS	: 68
# AND2	: 15
# AND3	: 1
# INV	: 20
# OR2	: 19
# XOR2	: 13
# Flip-flops/Latches	: 18
# LD	: 18
# IO Buffers	: 9
# IBUF	: 5
# OBUF	: 4

7.3 Applications

There are a lot of people provisions about coding principle in the up to date world. In workstation science, the place coding hypothesis originated, capable slip identification and revision codes need aid utilized within the transmission for advanced information. DRAM (dynamic random access memory) employments hamming codes to slip revision purposes. However, hamming codes have a least separation which empowers them with right special case awful spot for every code expression. Concerning illustration Pcs need advanced from 8-bit machines will 16-bits, 32-bits alternately significantly 64-bits, the capacity on right just An absolute bit lapse introduces the expanding plausibility for information debasement. In the vicinity of ever expanding information throughput Indeed developed hamming codes appear to be should miss the mark of the obliged slip revision.

Conclusion

Shortcoming tolerant Also trustworthy memory may be intended successfully, similarly as there will be reliable system, exact Furthermore flaw line spare systems; straight piece codes approach is used to outline a dependable framework. Memory is outlined in such an approach that errors need aid distinguished also remedied. Assuming that no errors are display those crazy is same. The circlet planned works once low force also giving phenomenal execution. The suggested framework being dependable What's more hosting deficiency tolerant proficiencies needs best couple Plant watts about control for operation.

Future Scope

In the recommended change those excess odds are appended In the wind of information odds. This dispenses with those overhead from claiming interspersing the excess odds during that sender conclusion Furthermore their evacuation during the collector wind following checking to single-bit lapse Also ensuing correction, if whatever. Further those exert required to recognizing those qualities of the excess odds may be easier in the suggested novel strategy. Hamming code is typically utilized for transmission from claiming 7-bit information thing. Scaling it to bigger information lengths brings about a considerable measure about overhead because of interspersing those excess odds and their evacuation after the fact. To contrast, those suggested technique may be exceedingly versatile without such overhead. We perceive that there may be main 7 spot overhead to a 56-bit information stream, which may be a significant part less contrasted with 4 touch overhead for a 7-bit information. Due to this characteristic this new system will be suitableness to transmission of vast measure information bit-streams likewise long Similarly as there is lapse.

References

1. V. S. Veeravalli, Fault Tolerance for Arithmetic and Logic Unit, IEEE Southeast on, GA, Atlanta (2009) pp. 329-334.
2. K. Matsumoto, M. Uehara and H. Mori, Evaluating the Fault Tolerance of Stateful TMR, in13th International Conference on Network-based Information Systems, Takayama (2010) pp. 332-336.
3. J.Vial, A. Virazel, A. Bosio, P. Girard, C. Landrault, S. Pravossoudovitch, Is triple modular redundancy suitable for yield improvement?, IET Computer Digital Technology .
4. N. M. Huu, B. Robisson, M. Agoyan and N. Drach, Low-cost fault tolerance on the ALU in simple pipelined processors, in IEEE 13th International Symposium on Design and Diagnostics of Electronics Circuits and Systems, Austria (2010) pp. 28-31.
5. S. Subharani and R. Palaniappan, Fault tolerance for ALU - evaluated module redundancy technique, in Proceedings of Singapore International Conference on Intelligent Control and Instrumentation, India (1992), pp. 92-96.
6. R. E. Lyons and W. Vanderkulk, The use of triple-modular redundancy to improve Computer reliability, IBM journal of research and development.

7. M. Hamamatsu, T. Tsuchiya and T. Kikuno" On the reliability of cascaded TMR Systems, in 16th Pacific Rim International Symposium on Dependable Computing, Tokyo (2010) pp. 184-190.
8. P. Yin, Y. Chen, C. Lu, S. Shyu et al., A multi-state fault-tolerant multiplier with tripl module redundancy (TMR) technique, in IEEE 4th International Conference on Intelligent Systems Modelling and Simulation, Bangkok (2013) pp. 636-641
9. V. Khorasani, B. V. Vahdat and M. Mortazavi, Analyzing area penalty of 32-bit fault tolerant ALU using BCH code, in 14th Euromicro Conference on Digital System Design.
10. D. Shinghal and D. Chandra, "Design and analysis of a fault tolerant microprocessor based on triple modular redundancy using VHDL,"International Journal of Advances in Engineering and Technology, vol. 1, no. 1, pp. 21-27. 2011
11. M. Dangeti, S. N. Singh, Minimization of transistor count and power in an embedded system using GDI technique, Universal Journal of Applied Computer Science and Technology, 2 (3) (2012) 308-313.
12. Prof. R.V.Kshrisagar, Sanjeev Sharma "An Algorithm for fault tolerance in FPGA"
13. Carl Carmichael "Triple Module Redundancy Design Techniques for Virtex FPGAs" XAPP197 (v1.0.1) July 6, 2006, xilinx .com.
14. Jonathan Johnson Michael Wirthlin "Voter Insertion Techniques for Fault Tolerant FPGA Design" NSF Center for High Performance Reconfigurable Computing (CHREC) Dept. of Elec. & Comp. Engineering Brigham Young University Provo, UT 84604.