2010 23rd International Conference on VLSI Design

Inexact Decision Circuits: An Application to Hamming Weight Threshold Voting

Bharghava R, Abinesh R, Suresh Purini, Govindarajulu Regeti
Center for VLSI & Embedded System Technology
International Institute of Information Technology, Hyderabad
{ bharghava.rpg07, abinesh}@research.iiit.ac.in, {suresh.purini, gregeti)@iiit.ac.in

Abstract

In this work, the authors present the idea of inexact decision making, and its application to threshold voting of Hamming weights, used in bus coding schemes, N-Modular Redundancy (NMR), Median filtering, and other pattern matching applications. Decision circuits can be tweaked, to perform in an inexact manner, in order to optimize in terms of delay and power, but still maintaining high system accuracy. One such circuit identified by the authors is the threshold voting of Hamming weights. A majority voter is a special case of this family of circuits. The proposed inexact voter consumes up to 8 times less power than the exact voter, with negligible reduction in system accuracy and performance. The leakage power is also reduced by a factor of 3. The inexact voter allows for a higher frequency of operation, by reducing the critical path delay by a factor of up to 3.4. The results obtained validate the application of inexact decision making, with respect to threshold voters.

1. Introduction

In the era of mobile devices and green computing, low power system design is gaining more momentum than ever before. Increased integration and higher operating frequencies compound the problem of power dissipation in VLSI chips. Increased cost of cooling, battery design, and size of a device put restrictions on the power budget of such a system. Traditional low power design methods result in a drop in system performance, in terms of frequency of operation. This loss in system performance, as a trade off for power, is also not desired, but is accepted without choice in most cases. Thus, a lot of research, nowadays, goes into reducing power dissipation in unorthodox ways. One such methodology involves compromising the accuracy of data processing [10] in view of human perception capabilities. This is applicable primarily to arithmetic circuits. The state machine of the system is left untouched, since incorrect computation in this case may result in catastrophic effects. But some decision circuits exist which can be tweaked to operate inexactly, without affecting data accuracy, or data flow. This work involves the design and analysis of one such inexact decision making circuit – the majority voter, which is a special case of threshold voting of Hamming weights. The major areas of application of this type of voter are in bus coding techniques [1][2][3], N-Modular Redundancy (NMR) [4], median filtering [6], pattern matching [7], and Self-checking circuits. In case of bus coding, the measure of system efficiency is in terms of power reduction obtained. In other cases, system efficiency is subject to correctness of data. Extensive analysis has been done on the power, and delay efficiency of the inexact voter, and also its effects on overall system efficiency for bus coding, median filtering, and NMR. The following sections explain the concept of inexactness, and how its effect on system efficiency is to be measured.

2. Concept of Inexactness

Traditional design techniques assume exact operation of a circuit, according to its specifications. In case of digital system design, the truth table of the system has to be fully applicable to the circuit designed. Instead of designing an exact system, certain inexactness can be introduced if it does not lead to catastrophic errors. An existing method to design approximate logic circuits is proposed in [8], where certain 0-minterms are assumed as don't-cares to form 1-approximate or 0-approximate circuits. The algorithm proposed in designing these circuits is tailor-made to implement concurrent error detection. The same technique cannot be adopted in cases where the exact circuit is replaced by the approximate logic circuit, as the resulting system performance may not vary linearly with the inaccuracy. The technique proposed in this paper considers both addition and

1063-9667/10 $26.00 © 2010 IEEE
DOI 10.1109/VLSI.Design.2010.76

removal of minterms, unlike the previous work. For example, consider the function represented in the K-Map shown in Fig. 1a. The single '1' that represents the term $a\bar{b}\bar{c}\bar{d}$ involves multiple gates in implementation.

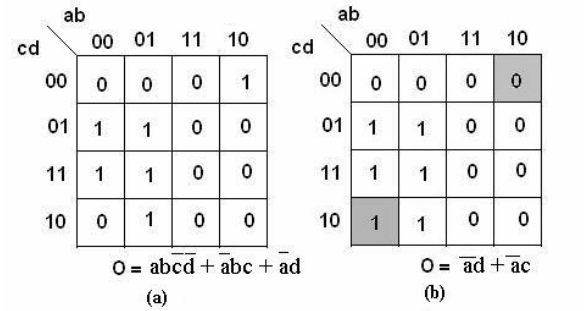


Figure 1: (a) Exact K-Map (b) Inexact K-Map

If it can be ascertained that removing that '1' does not lead to degrading results, the function can be implemented in an inexact manner as given in Fig. 1b. Also the single '1' that corresponds to $\bar{a}bc\bar{d}$ needs more gates as shown in Fig 1b. If an extra '1' can be added at $\bar{a}\bar{b}c\bar{d}$ the gate count to implement it is reduced. This is done to fill in the gaps perceived in the minterms placement. This leads to a drastic reduction in area, delay and power.

The error in the system depends on the number of input vectors whose output is altered, and the total number of input vectors. In the example functions considered, the inexact versions induce a 12.5% error (2 vectors in 16) in the system assuming all vectors occur equally likely. The circuit designed from inexact versions of the system will consume lesser power, occupy lesser area, and may also involve a lesser critical path delay, if the terms neglected were originally the only other term in its level of the critical path. Not all decision circuits can be designed in an inexact fashion. The extent of inexactness has to be quantified, and its effect on the system as a whole has to be analyzed extensively before such a step is taken. This is explained with the example of the majority voter.

3. Extent of Inexactness

The extent of inexactness that can be designed into a circuit depends on its application. The contribution of the input vectors neglected or added during design phase has to be weighed against the reduction in power, and/or area obtained, and this tradeoff has to be decided as per requirements. In the particular case where the purpose of the decision circuit is to decide on some action which eventually results in reduction in power dissipation, the quantification is simply the overall power saved using the inexact version of the circuit over the exact circuit. The inexact circuit may be more efficient in terms of power and speed than its exact version, but if the incorrectness in decision results in lesser power reduction, which offsets the initial advantage of inexactness, then application of inexactness to such a scenario is deemed invalid.

But in some cases, the system performance is measured in other quantities. For example, in the speculative execution of an architectural pipeline of a processor, the decision of whether or not to take a loop carries a large penalty for wrong decisions in terms of time, and power. Such a system cannot be quantified by power alone, because if the pipeline has to be flushed due to an incorrect decision, the total energy dissipated by execution of the entire program will be more, even though the power dissipated may be less. Also, time performance of the system is affected.

A generic error function could not be defined to quantify the inaccuracies arising out of the inexactness since the application can be linear or non-linear in nature. A linear error function can simply be the weighted average of the individual errors in the min-terms but it is not very useful for most systems. Designs like voters tend to be non-linear in nature thus designing a generic error metric a redundant one.

4. Hamming Threshold Voter

A binary hamming threshold voter, as in Fig. 2, is a circuit which compares the number of 1s with the number of 0s in its input, and outputs a 1 is the number of 1s is greater than a given threshold. If the number of 1s is lesser than the threshold, then it outputs a 0. If '*O*' is the output of the circuit, and $I = \{I_0, I_1, I_2, ..., I_{N-2}, I_{N-1}\}$ is the set of N input bits, then the majority function is described as follows:

O = 1 if No. of 1s in *I* > threshold
= 0 if No. of 1s in *I* <= threshold

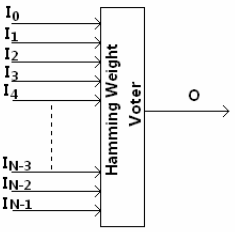


Figure 2: Hamming Weight Voter Block Diagram

A majority voter is a special case of Hamming weight threshold voters, wherein the threshold is set as half the width of the word to be voted on.Such a circuit is used as a decision making circuit in bus coding schemes, N-Modular redundant circuits, Self-testable circuits, median filtering, and pattern matching.

In this work, the authors propose an inexact design of a majority voter. Without loss of generality, other threshold voters can be design following a similar procedure. Such a circuit drastically reduces the circuit complexity and delay incurred. The voter, thus designed, is analyzed for performance variation from an exact majority vote.

The basic Majority Voter is implemented with a tree of full adders as in Fig. 3. The second part of the circuit can be replaced with a comparator to obtain a generalized hamming weight comparator.

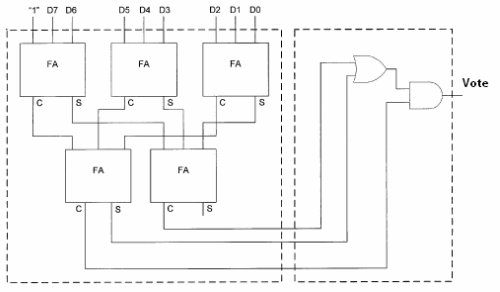
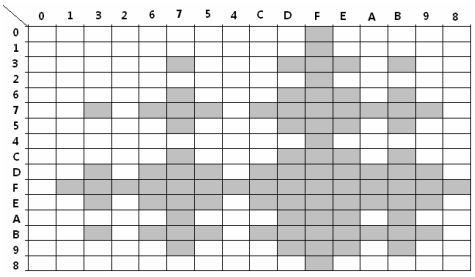


Figure 3: 8-bit Majority Voter Circuit (FA – Full Adder)

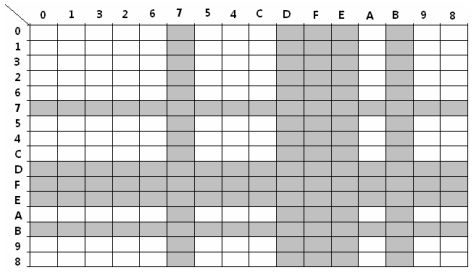
5. Inexactness in the Majority voter

The proposed majority voter will not produce the correct output for all input vectors. When such a circuit is used in decision making circuits in bus coding schemes, the accuracy of the overall system will not be affected. There will be some compromise in the reduction of power that could have been achieved, which is offset by the lower power consumption, and the lower delay incurred in the proposed circuit.

It was observed that the K-Map of an 8-bit voter (Fig. 4(a)) had intermittent 0s in rows and columns representing input combinations where 3 out of 4 LSBs or 4 MSBs were 1. Most of these *gaps* represent equal number of 1s and 0s and thus do not affect system performance. All these *gaps* were filled with 1s (as per the 2nd method), leading to inexact operation. Thus the rows and columns mentioned above are completely filled with 1s. This makes a 3-out-of-4 block the basic building block of the inexact voter. The output of this block is 1, if the number of 1s in the input is greater than 3. The inexact K-Map of an 8-bit voter is shown in Fig. 4(b). The rows and columns are numbered in hexadecimal. The 3-out-of-4 block can be implemented at gate level as in Fig. 5. In an FPGA, it takes only 1 LUT to implement the same. This is generalized to the entire family of voters. The input vector to the majority voter is divided into groups of 4, and each of this group is fed into a 3-out-of-4 block. The final majority vote is obtained by applying a similar vote algorithm to this N/4 signals.



(a)



(b)

Figure 4: (a) Exact K-Map of a Majority Voter (b) Inexact K-Map of a Majority Voter

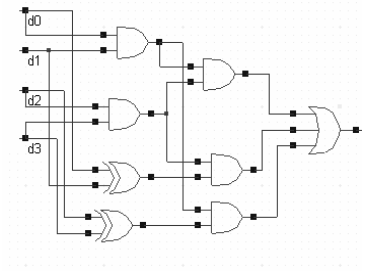


Figure 5: 3-out-of-4 block

For an 8-bit voter, the final vote is based on a 1-out-of-2 voting on the outputs of the two 4-bit exact voters as shown in Fig 6.

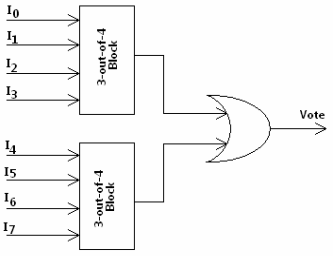


Figure 6: 8-bit Inexact Voter

For a 16 bit voter, it is a 2-out-of-4 decision, as shown in Fig 7, while the 32 bit voter will have a 5-out-of-8 circuit. Other Hamming weight threshold voters can be designed in a similar manner, depending on the threshold value.

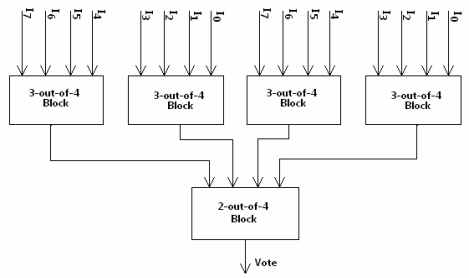


Figure 7: 16-bit Inexact Voter

6. Application to Bus Coding

Numerous bus coding techniques have been proposed in the past in order to reduce the effect of self-capacitance and coupling-capacitance of buses on the power dissipation of an integrated circuit. Wire shaping, buffer insertion, and several bus coding schemes have been used to reduce power dissipation due to coupling-capacitance between adjacent wires. Mitigating the effect of self-capacitance involves the reduction of data transitions on the bus. Some decision making circuit is required to ascertain whether the data has to be coded or not. Most techniques like this involve a majority voter, or a similar circuit.

The first major initiative in bus coding schemes was the Bus-invert coding scheme [1], wherein if data B is to be transmitted after data A, the hamming vector between them is calculated, using XOR gates. The hamming vector is passed through a majority voter. An intermediate output of the voter is the hamming distance. If the hamming distance is found to be greater than half the bus width, data B is inverted and transmitted. If the above condition is not satisfied, data B is sent as it is. A number of bus coding techniques that followed were modifications of the Bus-invert technique [2][3]. The major bottleneck in implementing such schemes is the Majority Voter. It can be shown that the number of full adders required to implement an N-bit Majority Voter is (N-2). The height of the tree increases logarithmically with the number of bits. There have been analog implementations of a majority voter [4], which is more efficient than a completely digital one. But in the case of an FPGA, analog implementations are not possible. Also, where completely digital circuits are required, an analog voter cannot be used. The efficiency of the inexact circuit for the case of bus-invert was compared with the same system using an exact majority voter. The efficiency is taken in terms of the reduction in transitions. The inexact majority voter will flag certain input vectors in a wrong manner. The results comparing them are shown in Figure 8.

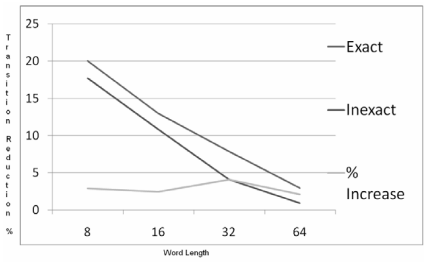


Figure 8: Comparison of inexact and exact voter for bus invert

The transition reduction obtained on using an inexact voter is 2.5% less than the transition reduction produced by an exact majority voter. For a supply voltage of 3.3V, an operating frequency of 100MHz, and a load capacitance of 3pF, the decrease in transition reduction amounts to a power dissipation of 20.42uW. If the difference in power dissipation between the exact and inexact voter in greater than this figure, then using an inexact voter is feasible.

7. Application to NMR

In an N-modular redundant system, the circuit which is to be fault tolerant is replicated N times and the output of all these N circuits is fed to a voter circuit. The voters used are majority voter, plurality voter, median voter etc. [4]. In this case, the decision of the voter will decide whether the fault incurred in the system is tolerable or not. A special case of this is the popular Tri-Modular-Redundancy (TMR).

Using an inexact voter in this case may not be justified, as redundancy is applied to get more reliability. This is especially true in critical applications, as that of control circuits in space applications, and nuclear reactors. But in the case of redundancy of sensors, the output of each sensor is subject to process variation, and may not render the correct output as required by it. Each sensor may give an output slightly deviated from the exact value. Thus, even usage of an exact majority voter may not produce the expected output. So, inexact voting solutions may not hinder correctness of the system output. This was tested by generating multiple copies of an input system with varying noise parameters. They were fed to a NMR system that votes on the individual bits to form a data word. The voting was done with both exact and

inexact circuits. The plots of the correlation of the 2 outputs with the input are shown in Figure 9 & 10.

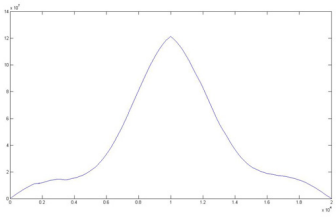


Figure 9: Correlation plot between the input and output of the exact system

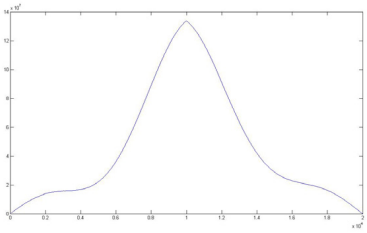


Figure 10: Correlation plot between the input and output of the inexact system

The difference between the two plots is very minimal with some difference in the maxima alone. Also the number of wrong decisions was 3% which is insignificant compared to the vastly reduced power, size and timing requirements.

8. Application to Median filtering

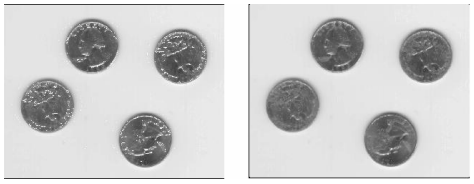
Median filters are typically used in image processing systems for a variety of purposes. The median filter algorithm involves selecting the middle value from a sorted list, which involves complex circuitry. So generally order statistic methods are used to determine the middle value, which involves voting on each bit positions of the list of data [6]. The rank order filter works by first voting on the MSB. Then the vote bit is checked with the MSB data bits. Those data whose MSB is different from the vote bit have their left bits changed to their MSB. This is done on all the bit positions.

(a) (b)

Figure 11: (a) Original image (b) Exact Median Rank Filtered Image

The inexact majority voter was used for implementing the rank order filter and the results are shown in figures 11 & 12.

The median filter with the inexact voter shows a small number of artifacts in filtering. These artifacts happen mainly because the filter saturates due to the nature of rank order filter. So a hybrid system was designed which checks the output of the inexact system saturates and if so will calculate that part with an exact system. This makes use of power gating to switch on the exact system only when needed. This will lead to an increase in area but will still lead to an enormous decrease in power consumed since the number of such instances will be very less (<5 %).



(a) (b)

Figure 12: (a) Inexact voter rank order filtered image (b) Hybrid voter rank order filtered image

9. Results

The proposed circuit and the exact voter circuit were implemented using Synopsys synthesis tools in 180nm process technology. The circuits were designed to operate at a frequency of 100MHz, with a supply voltage of 3.3V, and a wire capacitance of 3 pF.

Table 1: Comparison of power consumption for Inexact and exact voter

	Dynamic Power (uW)		Cell Leakage Power (nW)	
Word Length	Exact Voter	Inexact Voter	Exact Voter	Inexact Voter
8	143.66	26.43	2.46	.76
16	182.18	32.32	4.43	1.64
32	240.94	30.08	9.57	3.42

SPEC2000 benchmark files were used as stimulus for both the circuits. The dynamic power and cell leakage power consumption results are shown in Table 1. The efficiency of the proposed system can be gauged by comparing the actual power reduction that can be obtained in a typical bus. The net power reduction, after accounting for the overhead circuitry,

is shown is Table 2. The huge reduction can be easily understood as following diminishing returns.

Table 2: Comparison of overall power reduction for inexact and exact voter

	Overall Power Reduction(uW)	
Word Length	Exact Voter	Inexact Voter
8	20	117
16	-77	56
32	-177	3

The proposed system shows that the technique can work with lesser accuracy. It can be said that it is always better to use inexact circuitry in low power bus coding system since exact systems can never save more power than they consume. Also, the leakage power of an inexact voter is very less, making it more advantageous as technology is further scaled down.

Table 3: Area/Delay Comparison for Inexact and Exact voter

	Critical Path Delay(ns)		Area(um^2)	
Word Length	Exact Voter	Inexact Voter	Exact Voter	Inexact Voter
8	1.97	1.32	987	362
16	3.02	1.60	2163	811
32	3.59	1.61	4840	1682

Delay analysis for the two circuits, which determines the maximum frequency of operation, is presented in Table 3. The reduced critical path delay of an inexact voter allows for higher frequency of operation, leading to higher bandwidth. The inexact voter also consumes lesser chip area than its exact counterpart as shown in Table 3.

10. Conclusion

The utility of inexactness in decision circuits was presented with specific attention to majority voters, a class of Hamming Threshold voters. The tradeoff between system accuracy and the power consumed was analyzed using the majority voter for multiple applications. It was shown that the reduction in accuracy was acceptable compared to the significant reduction in hardware requirements for Bus coding, Median filters, and N-Modular redundancy. The inexact voter also enables higher frequencies of operation and occupies lesser chip area. These results validate the use of inexact decision making in the case presented. The proposed technique can be extended to design any circuitry, but the tradeoff should be analyzed depending on the effect of the inaccuracy on final system performance.

11. References

[1] M. R. Stan and W. P. Burleson : “Bus-invert coding for low-power I/O”, IEEE Transactions on VLSI Systems, vol. 3, pp. 49–58, Mar. 1995.

[2] Ki-Wook Kim, Kwang-Hyun Baek, Naresh Shanbhag, C. L. Liu, Sung-Mo Kang, "Coupling-Driven Signal Encoding Scheme for Low-Power Interface Design," iccad, pp.318, 2000 International Conference on Computer-Aided Design (ICCAD '00), 2000

[3] Youngsoo Shin, Soo-Ik Chae, and Kiyoung Choi : “Partial Bus-invert Coding for Power Optimization of Application-Specific Systems”, IEEE Transactions on VLSI Systems, VOL. 9, NO. 2, pp 377-383, APRIL 2001.

[4] Parhami, B : “Voting Algorithms”, IEEE Transactions on Reliability, 1994, 43, (4), pp. 617-629

[5] Wei-Chung Cheng and Pedram, M. : “Memory bus encoding for low power: a tutorial”, 2001 International Symposium on Quality Electronic Design, pp 199-204

[6] Barun k. Kar and Dhiraj K. Pradhan “A new algorithm for order statistic and sorting”, IEEE Transactions on signal processing, Vol 41, No 8, August 1993

[7] Barral, C., Coron, J.-S., and Naccache, D. “Externalised fingerprint matching”, Springer LNCS 2004, 3072, pp. 309–315

[8] Mihir R. Choudhury and Kartik Mohanram . “Approximate logic circuits for low overhead, non-intrusive concurrent error detection”, DATE 2008, pp.903-908.

[9] Krishna V. Palem: “Energy Aware Computing through Probabilistic Switching: A Study of Limits”. IEEE Trans. Computers 54(9): 1123-1137 (2005)