Reversible Computing : Looking ahead of the curve

Saurabh Sahni, Author* Ankur Bakshi, Author*

** Institute of Engineering & Technology – DAVV, Indore, India*

Abstract

The new era mantra of success has been fabricated by computing revolution. Everyday we see IT penetrating into new application areas with the performance-per-unit-power-consumption of digital technology improving, as Moore had predicted. When the conventional approach will run out of steam, reversible computing will then be the only way to rapidly improve performance.

Reversible computing is the application of principles of recycling to computing. It means computing using a physical mechanism that is thermodynamically reversible and logically reversible as well. They are adiabatic systems that recycle their energy and emit very-little heat. When a computer performs a logical operation, the unwanted bits after the operation are thrown away with sudden changes of voltages from positive to negative, and dissipated as heat. For computer speeds to keep on increasing, we need to 'un-compute' the unwanted bits. On applying reversible computing we can move charges from one node to other with the help of oscillators and switches, hence saving both power and time.

This paper concentrates on reversible computing, its need and application in modern computing scenario. It covers design and implementation of reversible hardware logic as well as instruction sets. Various limits of the reversible computing are explained. Brief overview of reversible computing as a special case of quantum computing is also given. To realize reversible computing we have implemented a reversible full adder which can be used as the basis of reversible computer. Also a reversible programming logic been proposed.

Keywords: - *Reversible Computing, reversible digital logic, logic designs, reversible instruction sets*

1. Introduction

Ever wondered why Pentium processors become so hot on use. If they are cooled aggressively the performance improves. The processors throw away unwanted bits after computations in the form of heat. This heat dissipation increasingly becomes a limiting factor on performance.

Gordon Moore predicted that performance of integrated-circuit would continue to improve at an exponential rate with the performance per unit cost increasing by a factor of 2 every 18 months. But, there is a limit on how far we can keep doing this. If we want computer speeds to keep on increasing, we need to un-compute the unwanted bits.

Reversible computing means computing using a physical mechanism that is thermodynamically as well as logically reversible (ref.4). They are adiabatic systems that recycle their energy and thus emit very little heat.

2. The concept

Reversibility in computing implies that no information about the computational states can ever be lost, so we can recover any earlier stage by computing backwards or un-computing the results. This is termed as logical reversibility (ref.4). The benefits of logical reversibility can be gained only after employing physical reversibility. Physical reversibility is a process that dissipates no energy to heat. Absolutely perfect physical reversibility is practically unachievable.

Computing systems give off heat when voltage levels change from positive to negative: bits from zero to one (ref.3). Most of the energy needed to make that change is given off in the form of heat. Rather than changing voltages to new levels, reversible circuit elements will gradually move charge from one node to the next. This way, one can only expect to lose a minute amount of energy on each transition.

Reversible computing strongly affects digital logic designs. Reversible logic elements are needed to recover the state of inputs from the outputs. It will impact instruction sets and high-level programming languages as well. Eventually, these will also have to be reversible to provide optimal efficiency.

3. Need of reversible computing

High-performance chips releasing large amounts of heat impose practical limitation on how far can we improve the performance of the system (ref.1,4).

Reversible circuits that conserve information, by uncomputing bits instead of throwing them away, will soon offer the only physically possible way to keep improving performance.

Reversible computing will also lead to improvement in energy efficiency. Energy efficiency will fundamentally affect the speed of circuits such as nanocircuits—and therefore the speed of most computing applications.

To increase the portability of devices again reversible computing is required (ref.2). It will let circuit element sizes to reduce to atomic size limits and hence devices will become more portable.

Although the hardware design costs incurred in near future may be high but the power cost and performance being more dominant than logic hardware cost in today's computing era, the need of reversible computing cannot be ignored.

4. Design and Implementation

Reversible computing cannot be implemented on the existing logic designs which are irreversible in nature. The present day logic gates do not let us uncompute outputs to recover input values, except NOT gate which is the only reversible gate.

Reversible logic gates are circuits that have the same number of inputs and outputs and have one-to-one and onto mappings between inputs and outputs; thus, the input states can be always reconstructed from the output states.

4.1 Basic Reversible Gate

Consider the case of a simple two input XOR gate which is irreversible. On repeating one of the input to output makes the gate reversible. It can be realized as shown in Fig. 1(ref.6):

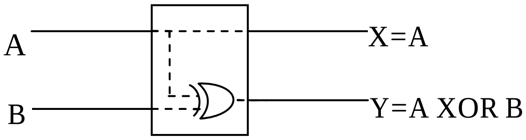


Fig.1 Reversible XOR Gate

Table 1 shows the truth table for the above circuit.

TABLE 1 Truth Table for Reversible XOR Gate

A	B	X(=A)	Y
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

4.2 Fredkin Reversible Gate

Fredkin proposed following (3,3) reversible gate with inputs X2, X1, X0 and outputs Y2, Y1, Y0 where

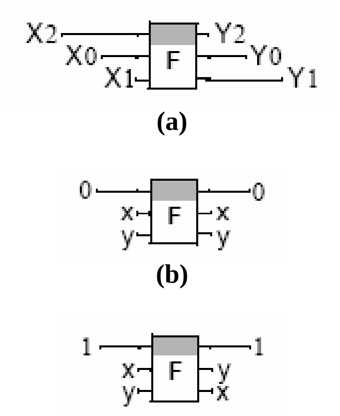
$$Y0 = \begin{cases} X0, & X2 = 0 \\ X1, & X2 = 1 \end{cases},$$

$$Y1 = \begin{cases} X1, & X2 = 0 \\ X0, & X2 = 1 \end{cases},$$

and

$$Y2 = X2$$

This gate is known as the Fredkin gate (FG). Fig. 2 shows the FG circuit symbol.



(a)

(b)

(c)

Fig. 2 (a) Basic Fredkin gate, (b) Operation when X2=0, (c) Operation when X2=1

This Fredkin gate is a reversible gate as at any moment we may move back to a previous state by just knowing the outputs at the current state.

Thus, truth table for Fredkin gate will be as Table 2

Table 2 Truth Table for Fredkin Gate.

Inputs			Outputs		
X2	X1	X0	Y2	Y1	Y0
0	x	y	0	x	y
1	x	y	1	y	x

Thus in the above gate no voltage level conversions are required, the output may be driven from the input voltages and hence energy will be saved as well as propagation delay will be reduced.

4.3 Reversible Logic Circuits

Using the above gate we may obtain all logic gates as shown in Fig.3.

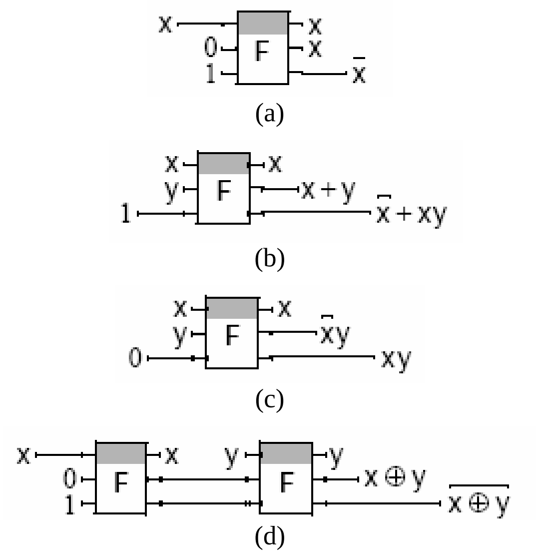


Fig. 3 Reversible Gate designs (a) NOT Gate (b) OR gate (c) AND Gate (d) XOR gate

Using these logic gates we may design different reversible and energy efficient logic circuits. The full adder, basic building block of an ALU, can be implemented using reversible logic. A possible design is shown next:

A conventional 4 bit irreversible adder circuit is shown in Fig. 4:

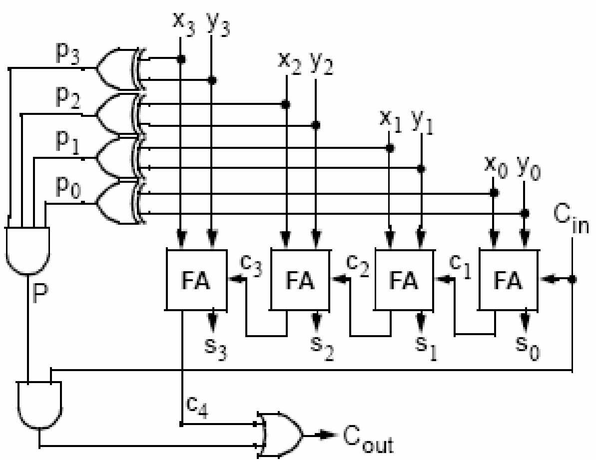


Fig. 4 Irreversible adder

The same adder can be implemented using reversible logic. The fig. 5 show one stage of a n bit full adder.

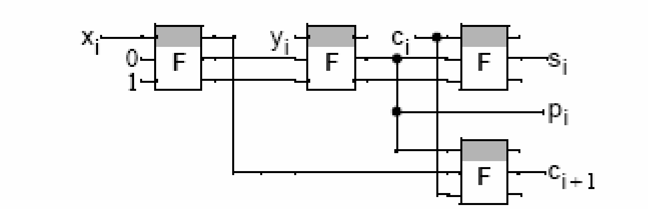


Fig. 5 Reversible n bit Logic Adder

On cascading n stages of the circuit shown in fig.5 we may obtain n-bit reversible full adder.
Similarly, other complex hardware circuits can also be designed.

5. Reversible Instruction Sets

In order to achieve reversibility to its maximum extent, the instruction set also needs to be reversible. A reversible instruction is one that, given the result and the operation can be "undone" (ref.8).
We need to design new instructions for implementing reversibility at software level. Only then reversible computing can be realized fully.

Below is a simple reversible arithmetic logic:

```
a = 10 ;
a=a+5;
a=a*4;
```

To undo the above sequence of instructions following instructions must be executed:

```
a=a/4;
a=a-5;
//Here we get back the initial value of a (=10)
```

The reversible instruction set should provide an inherent support for such operations.
While un-computing the instructions, addition will become subtraction, multiplication will become division and vice versa. At the same time when we will un-compute conditions the control will enter the loop body if the condition is false.

Reversible loop statements can have following structure:
In reversible instruction set we will replace a simple while loop with the 2 while loops at start as well as end of the loop. During forward traversal condition must be true while in reverse traversal condition must be false to stay in the loop.

Lets consider the case of a program printing 0 to 9.

```
LL=0;           //Lower Limit
UL=10;          //Upper Limit
i=0;
while(LL==i)
{
print(i);
i++;
}
while(i!=UL);
```

Fig. 6. Reversible program loop.

These statements in fig. 6 print values of i from 0 to 9 while computing forward. During un-computing they print 9 to 0.

Similarly, complex algorithms can also be implemented using the reversible instruction set.

6. Reversible Computing and its contemporaries

Reversible computing is similar to quantum computing, and shares some technological requirements with it. However, the motivations for them are different, as are some of the requirements. Quantum computing uses quibits, not bits, as its basic unit of information. Quibits maintain two states, enabling significant performance increases to occur (ref.5). A classical reversible operation is therefore a special case of a quantum operation, in which classical input states are always mapped to classical output states. The quantum viewpoint therefore expands the range of possible operations. An input state can be mapped to a state that is "in between" classical states, in the sense that the vector has non-zero components in more than one classical state. It can be established that, every quantum computer is a reversible computer, but not every reversible computer is a quantum computer

Reversible computing is also closely related to nano-computing and bio molecular computing.

The design of a nano-computer will specify every atom and covalent bond present in the device. We can reasonably expect the switches, gates, or other embodiment of the logical elements to be on the order of a nanometer in size. At this order of magnitude reversible computing becomes indispensable as heat dissipation by erasing large number of bits will hamper the system performance.

In bio molecular computation (ref.7) presence of reversible logic can also be observed. Information can be encoded in DNA and DNA amplification can be used to implement logically reversible gates that comprises a complete set of operations capable of universal computation.

7. Applications:

Reversible computing may have applications in computer security and transaction processing, but the main long-term benefit will be felt very well in those areas which require high energy efficiency, speed and performance.

The potential application areas of reversible computing include the following (ref.1, 3, 4) :

- **Laptop/handheld/wearable computers** – Obviously, here there are other power dissipation sources such as displays, disks and transmitters to consider. But in any products where computation is the dominant component of total power (but not total weight), the application of adiabatics will be beneficial. Adiabatic techniques should even allow conversion to battery less (fully solar or RF powered) operation in small wearables.

- **Spacecraft** – The high cost-per-weight of launching computing-related power supplies, solar panels, and cooling systems into orbit imposes a demand for adiabatic power reduction in spacecraft in which these components weigh a significant fraction of total spacecraft weight (and the computer itself does not).

- **Implanted medical devices** – Such as automatic drug-delivery systems or pacemakers. These may need to do significant computation continuously to execute decision-making algorithms to determine whether to deliver a drug or electrical stimulation at a given moment based on sensor inputs. However, one obviously wishes to minimize their power requirements, since the patient will wish to be able to move around freely for long periods of time without replacing batteries or recharging the device. The device might even be powered by a chemical engine using the body's own resources.

- **Wallet "smart cards" –** An example of a wearable computer for personal & financial data, digital cash, etc.

- **"Smart tags" on inventory –** Used for locating objects. Smart tags may sit on boxes in warehouses for months, listening to the airwaves and doing low-power signal processing. If they hear a location request that includes their own id number or a template matching their description, they locate themselves using GPS and broadcast a signal giving their coordinates. Obviously these devices need very low power in order that they can sit around for a long time and not go dead. With adiabatics, they could even be powered by ambient EM via their receiving antennas.

8. Reversible Computation Limits:

1. Computation Rate Limits

Reversible computing also has a limit in terms of rate (ref.4). There is a maximum rate at which transitions (such as bit-flips) between distinguishable states can take place. One form of this upper bound depends only on the total energy E in the system, and is given by 4E/h, where $h=2\pi\mathbf{h}$ is the unreduced Planck's constant. But this bound is quite high and we may achieve a maximum rate of about 1 PHz (10^{15} Hz) per device.

2. Ideal Switch and Oscillator

For realizing reversible computing to the best extent the switches and oscillators used must be close to ideal state (ref.3). They must not loose energy in transactions occurring for flipping of bits. But designing ideal switches and oscillators is not possible so they too impose a limit on reversible computing.

3. Synchronization

Appropriate synchronization is required inside the reversible processor (ref.3). In order to make rapid forward progress through the computation, the machine state needs to evolve nearly ballistically (that is, dominated by its forward momentum). A certain ballistic asynchronous reversible processor would be disastrous if small misalignments in the arrival times of different ballistically-propagating signals occur. The reversible systems must not suffer from such instabilities which may impose a limit on its performance.

9. Conclusion

The recycling of real life entities like bottles, cans, paper, etc was unknown some decades back. But they are quite prominent today. The same applies to the computing world also. Computers today are terribly wasteful devices. They throw away millions of bits, billions of times every second. They are based on irreversible logic devices, which have been recognized as being fundamentally energy-inefficient for several decades.

Truly, the only way we might ever get around this limit is by using reversible computing, which uncomputes bits that are no longer needed, rather than overwriting them. Un-computing bits allows their energy to be recovered and recycled for use in later operations. As the demand for performance becomes more prominent the need for reversible computing devices will become stronger. Before the computer industry reaches the fundamental brick wall of performance and energy constraints of computing devices, reversible computing needs to be fully developed.

10. References

[1] Michael P. Frank, Reversibility for Efficient Computing, , Ph. D. Thesis , May 1999. http://www.cise.ufl.edu/~mpf/rc/thesis/phdthesis.html

[2] Carlin Vieri, Reversible Computing for Energy Efficient and Trustable Computation, April 1998 http://www.ai.mit.edu/~cvieri/reversible.html

[3] Network Magazine: Get Mean, Go Green, January 2004, http://www.networkmagazine.com

[4] Michael P. Frank Reversible Computing Page, http://www.cise.ufl.edu/~mpf

[5] Reversible Computing: Quantum Computing's Practical Cousin, Simons Conference on Quantum and Reversible Computation

[6] Pawel Kerntopf, Marek A. Perkowski, Mozammel H. A. Khan, On Universality of Ternary Reversible Logic Gates

[7] Joshua P. Klein, Thomas H. Leete, Harvey Rubin, University of Pennsylvania, Biomolecular Implementation of Logically Reversible Computation with Minimal Energy Dissipation

[8] Andrew Dickinson, An Introduction to Reversible Computing using phPISA, December 2001, http://www.cise.ufl.edu/research/