A Scalable Reversible Computer in Silicon *

Michael Frank, Carlin Vieri, M. Josephine Ammer, Nicole Love,
Norman H. Margolus, and Thomas F. Knight, Jr.

MIT Artificial Intelligence Laboratory, Cambridge, USA

Abstract. The reversible and "adiabatic" transfer of charge in digital circuits has recently been a subject of interest in the low-power electronics community, but until now, no one had created a complete, fully reversible CPU using this technology. Fundamental physical scaling laws imply that a fully reversible processing element would permit unboundedly greater efficiency at some tasks, by several different metrics, than can be achieved with any possible irreversible computer. This paper describes the design of Flattop, a fully-adiabatic chip now in fabrication, which can serve as a general-purpose parallel processor when tiled in large arrays. Flattop implements the Billiard Ball Cellular Automaton, a universal and reversible model of computation. Flattop is implemented in a standard 0.5 μm CMOS silicon process using the *Split-Level Charge Recovery Logic* (SCRL) circuit family developed at our lab. Calculations indicate that our circuit can operate with about 2000 times the energy efficiency of an equivalent chip based on standard circuit techniques, modulo some unsolved power supply issues. Although Flattop is itself not a very practical architecture for performing arbitrary computations, it is an important proof-of-concept, demonstrating that physically reversible universal computers can actually be built using current technology.

1 Introduction

Microscopic physical dynamics is fundamentally invertible, or *reversible*, a given micro-state has only one path from the preceding and to the subsequent micro-state. This implies that the inner workings of a computer approaching the limits of physics must be reversible as well.

Conventional computers regularly perform "irreversible" operations that erase macro-state information. An operation as basic as switching the output of an inverter irreversibly erases the value previously stored on that node. But, since physics is reversible, any "irreversible" operation in a computer does not really destroy any micro-state information about the history of the system, it only transfers the information to some inaccessible, uncontrolled form, such as the thermal motions in a heat bath. Adding one bit of entropy to a thermal system having temperature T requires adding at least $k_B T \ln 2$ of energy into the system, where k_B is Boltzmann's constant, in order to increase the number

* This work was supported by ARPA contract DABT63-95-C-0130.

of available states [4]. This rule can even be considered to be a definition of temperature in terms of energy and entropy.

Conventional digital electronics dissipates far more energy than this, on average, whenever it erases a bit stored as the voltage of a circuit node; the energy loss using conventional methods is at least $\frac{1}{2}CV^2$, where C is the capacitance of the node, and V is the change in voltage needed to represent the new logic value. In a common CMOS technology, this energy is about 10^8 times higher than $kT \ln 2$.

1.1 Adiabatic circuits

The recent development of so-called "adiabatic" circuits (*e.g.*, see [3]) has shown that this large $\frac{1}{2}CV^2$ energy per bit-change is not strictly necessary. If circuit nodes are charged and discharged gradually, under the control of redundant information stored in other circuit nodes, then the circuit can change state in a quasistatic, *adiabatic* fashion. Here, as in the study of heat engines, an adiabatic process is one that takes place without any heat flow into or out of the system.

A few years ago, members of this group invented SCRL [9, 8], a particularly simple and elegant adiabatic circuit technique that allows construction of integrated, reversible, pipelined circuits using ordinary commercial CMOS fabrication processes. SCRL's operation is described in more detail in section 3.

Due to the non-zero resistance of real switches and wires, SCRL circuit operations will in fact still dissipate some energy, proportional to the speed at which charge is moved around. These operations are therefore not perfectly reversible in the physical sense. This relationship between speed and degree of reversibility holds for all physically plausible processes; it seems that in any system there will always be some friction-like effects that dissipate energy in proportion to speed. The closer a system approaches quasistatic behavior, the lower the energy dissipation.

An important parameter of any asymptotically reversible implementation technology is the exact value of the proportionality constant between the energy wasted per operation and the rate at which operations take place. This value has no known fundamental lower limit.

Another problem with current technology is that when the ratio of operating voltage to temperature is small, CMOS transistors leak small amounts of current even when they are turned off, which sets a lower limit on the energy per operation. However, this limit can be made exponentially small by operating with higher voltages or lower temperatures. Also, SCRL circuits may be designed so as to reduce this static leakage power at the cost of increased dynamic power. The dynamic power may then be reduced by slowing the speed of operation.

Problems due to resistance will diminish as technology improves. Circuits built from exotic but existing low-temperature superconducting switches, such as Josephson junctions [5], offer extremely low resistance to fast reversible changes of state and negligible leakage effects.

Moreover, even with the high resistance and measurable leakage of conventional transistors at ordinary voltages and temperatures, SCRL is still capable of much greater energy-efficiencies than conventional CMOS circuits. For the commercial fabrication process used to make Flattop, it is estimated that at normal temperatures and voltages, SCRL circuits such as Flattop's can achieve on the order of 2000 times less energy per operation than normal circuits (that is, 2000 times more MIPS per Watt). To achieve this maximal energy efficiency the circuit must be run at relatively low clock speeds, around 100 kHz. At today's more typical CPU clock speeds on the order of 200 MHz, SCRL uses about the same amount of power as regular CMOS. These dissipation comparisons are for the on-chip circuits; certain power supply generation issues are still unsolved.

The low-speed but extremely low-energy circuits achievable with SCRL might have near-term applications in severely energy-limited environments, such as digital watches, portable or implanted medical monitoring devices, or any manner of other independently-powered digital devices.

1.2 Scaling issues

Perhaps more importantly, when maximum compactness of a machine is required and speed is limited by the achievable cooling capacity per unit of surface area, as is becoming the case, SCRL can actually be faster than standard CMOS, by allowing many more active circuits to be packed closely in three dimensions.

For example, with the device technology used for Flattop and logic gates spaced about 10 microns apart, if cooling is limited to 100 Watts per square centimeter of surface area (a low value, but reasonable for portable devices), the maximum clock speed for a surface densely covered with either conventional or SCRL circuits is limited by heat removal to a value of around 200 MHz. But if the allowed dissipation is only ten Watts per square centimeter, SCRL can still run at 72 MHz while the ordinary circuit is reduced to 20 MHz, less than a third as fast.

Alternatively, with again a dissipation of 100 Watts per square centimeter but a stack of 100 circuit boards on top of each other for a compact, massively parallel computation, the SCRL version can run at 20 MHz while the CMOS version slows to a crawl of 2 MHz.

In general, the maximum clock speed for a stack of SCRL circuits, when limited by cooling, decreases in proportion to only the square root of the number of layers being stacked, whereas in standard CMOS the speed decreases linearly. As the scale of a machine increases, SCRL is unboundedly faster than standard CMOS (or any possible irreversible technology) at performing reversible computations which call for a compact, three-dimensional network topology, such as volumetric simulations of reversible three-dimensional physical systems.

In general, 3-D arrays of reversible processing elements will likely offer an asymptotically most efficient physically possible non-quantum model

of computation, in that such arrays should be able to simulate any non-quantum computer model with at most a constant factor slowdown, dependent on technology but not on scale.

For further discussion of scaling issues see [1, 2].

1.3 Flattop

Despite interest in reversible and adiabatic computation, no one has yet designed a complete computer based on fully reversible, adiabatic technology. Such a design would demonstrate the physical realizability of universal reversible computation and provide practice analyzing and programming such a computer.

Thus this group has designed a very simple adiabatic universal computing element in SCRL circuits. The chip is currently being fabricated and when completed it will serve as a benchmark test chip for evaluating SCRL's power savings using a variety of power supplies.

To make the project more feasible, Flattop has the simplest parallel universal computer model the group is aware of: Margolus's Billiard Ball Model Cellular Automaton (BBMCA) [6]. The BBMCA is not the most convenient computer to program, but it is simple, universal, reversible, and scalable. Though the BBMCA model itself is only a two-dimensional cellular automaton, Flattop could in principle be wired in a 3-D mesh as well, for scalably executing reversible 3-D algorithms.

The following sections describe the BBMCA and SCRL, the Flattop circuit design at several levels, and a derivation of the power savings achieved.

Flattop is named after the local pool hall, Flattop Johnny's.

2 The BBMCA

In the billiard-ball model (BBM) of computation, logic values are modeled by the presence or absence of "billiard balls" moving along predetermined paths in a grid. These classical, elastic, hard spheres are constrained to travel at constant speed along straight lines. All the balls in the system are synchronized so that all collisions happen at well defined grid points. A logic gate is a point where two potential ball trajectories interact. Fixed walls may be introduced to bounce signals around. Figure 1 shows two possible BBM logic gates. The rectangles are fixed walls. The output is represented by the presence or absence of a ball at the designated output points of the gate. Any reversible logic circuit can be implemented in this model.

The BBM cellular automaton is a universal, reversible computing element based on these interactions of hard spheres. Each cell is a four input, four output logic block. Balls that enter the block alone continue along their original path, out the other side of the block. If exactly two balls enter opposite corners of the block, they "bounce" off each other and exit the block through the

Logic stages clocked on different phases are isolated from each other by pass gates. These pass gates are clocked by pairs of full-swing, ramping power clocks.

Basic SCRL stages cannot compute non-inverting logic functions, but it is possible to compute non-inverting functions by providing an extra pair of "fast" rails that split before the main rails do, and re-combine after the main ones. (See [9, 8].) These rails can be used to drive an extra level of logic (such as an inverter) to drive the inputs of the main logic. In this way, a single SCRL stage can compute inverting or non-inverting logic functions. The alternative for the circuit designer is to propagate all signals in both positive and negative polarity. Due to the relative simplicity of Flattop's circuits, a single non-inverting stage was needed and was implemented using "fast" rails. Additionally, some signals were distributed in dual rails.

The Flattop array is implemented in so-called "3-phase SCRL" [8] because it is the simplest version of SCRL that doesn't depend on dynamic charge storage. The ability to run the clocks arbitrarily slowly or stop them altogether without worrying about losing stored logic values eases chip testing.

Figure 4 details the timing discipline for 3-phase SCRL with fast rails, next to an abstract illustration of the circuit components controlled by each timing signal. Space precludes giving a detailed account of the diagram here, but note that a complete cycle is 24 times as long as a single rising/falling transition. The shaded boxes denote the portion of each cycle during which the corresponding node connecting two stages will contain a valid logic value. One constraint of 3-phase SCRL is that all feedback loops must be a multiple of three stages long. a complete round of interaction from a cell to its neighbors and back in Flattop occurs in six stages, with three stages in each cell, and all cells identical. The 2-cell feedback loop could perhaps have been accomplished in three stages, but at a cost of more fast rails and asymmetric cells.

SCRL, like other adiabatic circuit families, requires a power supply that can generate these many resonant, swinging supply rails with low dissipation. If the power supplies are not efficient enough, power savings are limited. The development of a satisfactory power supply is an active research topic but is assumed to be available for the purposes of this paper.

4 Circuit Design

This section describes the design of the Flattop circuitry. Space constraints prohibit including all schematics, but examples of some of the more important components are given.

A fair amount of time was spent early in the project figuring out how to implement the BBMCA update rule in three SCRL stages using a minimum amount of logic. Initial concepts had 600 transistors, but the design finally was reduced to a 240-transistor design. However, this design did not incorporate array initialization, so extra logic was added to the design which increased the

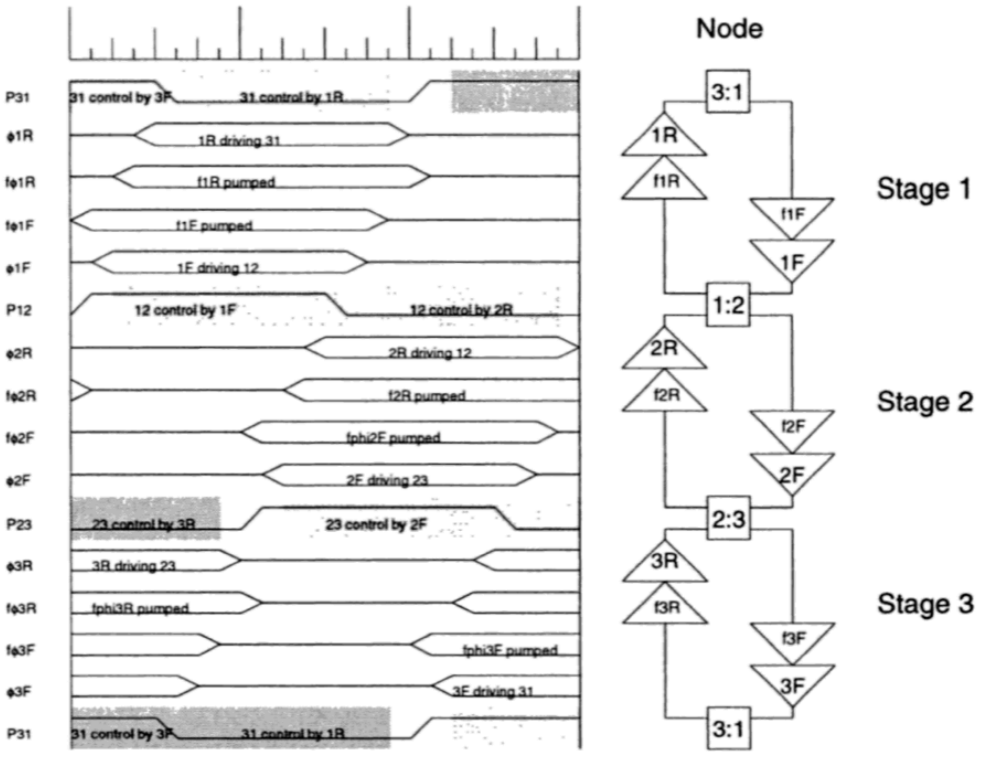

Fig. 4. Timing diagram for 3-phase SCRL clock discipline.

transistor count to 292. It is possible to reduce the size further, but the logic has not been re-minimized since adding array initialization.

The initial 240-transistor logic involved stages generating the following output signals. The inputs to stage one are A, B, C, and D.

- Stage 1: $\overline{A}, \overline{B}, \overline{C}, \overline{D}$, and $\overline{S} = \overline{(A+C)(B+D)}$.
- Stage 2: $A, B, C, D, S, \overline{S}$, and $\overline{A_{out}} = \overline{SA + \overline{S}\overline{A}(C+BD)}$, and similarly for B_{out}, C_{out}, and D_{out}.
- Stage 3: $A_{out}, B_{out}, C_{out}$, and D_{out}.

In this logic, stage one is mainly a buffer but also generates the S signal used in stage two. Stage two performs the real work of computing the update function, and stage three is another buffer. The reason stage two must produce S at its output is so that S will be available for use by the reverse half of stage two to compute the inverse update function.

4.1 Block diagram

Figure 5 shows a schematic block diagram of a single processing element. Note the three blocks, one for each of the three stages in the 3-phase SCRL pipeline. Each stage shown below contains both the forward and reverse SCRL circuitry.

The lollipop-shaped icon above each stage represents the set of swinging supply rails, in one of the three phases, which adiabatically drive the stage. The cell has four inputs: A, B, C, and D, which come from the four neighboring cells and has four corresponding outputs which go back out to those cells.

The SHIFT in and out signals are global signals shared by all cells; they tell the cells whether to operate in initialization mode or normal mode. In initialization mode the array of cells behaves as a shift register, and the array contents may be shifted in and out; in normal mode, the array obeys the BBMCA update rules.

The boxes attached to the input are for setting initial conditions during HSPICE simulation of the circuit.

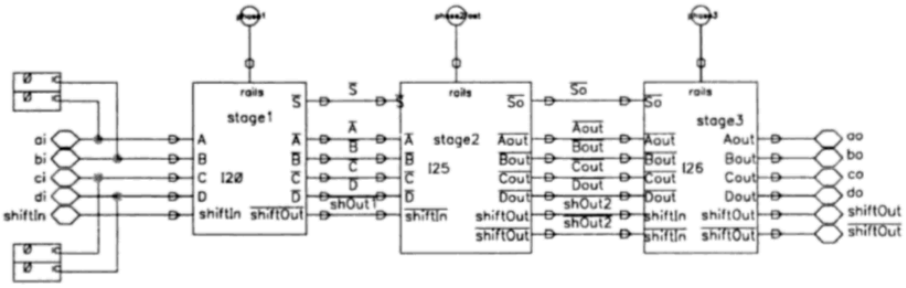

Fig. 5. Block diagram of PE cell.

Figure 6 shows the schematic icon that represents an entire processing element. The icon portrays the 2×2 block of BBMCA cells which the PE is updating, with the PE inputs and outputs placed in the appropriate cells. The cell grid is rotated 45 degrees from the representation in Figure 3 to show how the CA array is oriented with respect to the edges of the chip. With this orientation, the array of processing elements can communicate along pathways that run parallel to the chip edges, making layout easier.

Figure 7 shows one corner of an array of these processing elements. In the upper left corners are pathways used for initialization and reading out the whole array when used as a shift-register. Along the edges of the array are edge cells which provide connections to pins, allowing chip to chip communication during normal operation. There are not enough pins to allow communication everywhere along the edge, so in other places the wires at the edge just loop around to feed back into the array. Every PE receives the global shift signals, which run horizontally.

Figure 8 shows the entire 20×20 array of processing elements which we fabricated for testing purposes.

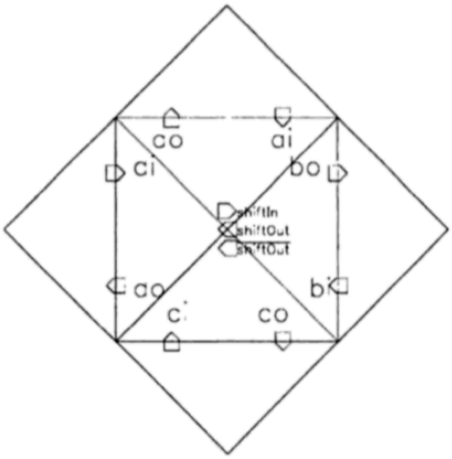

Fig. 6. Icon for a single cell.

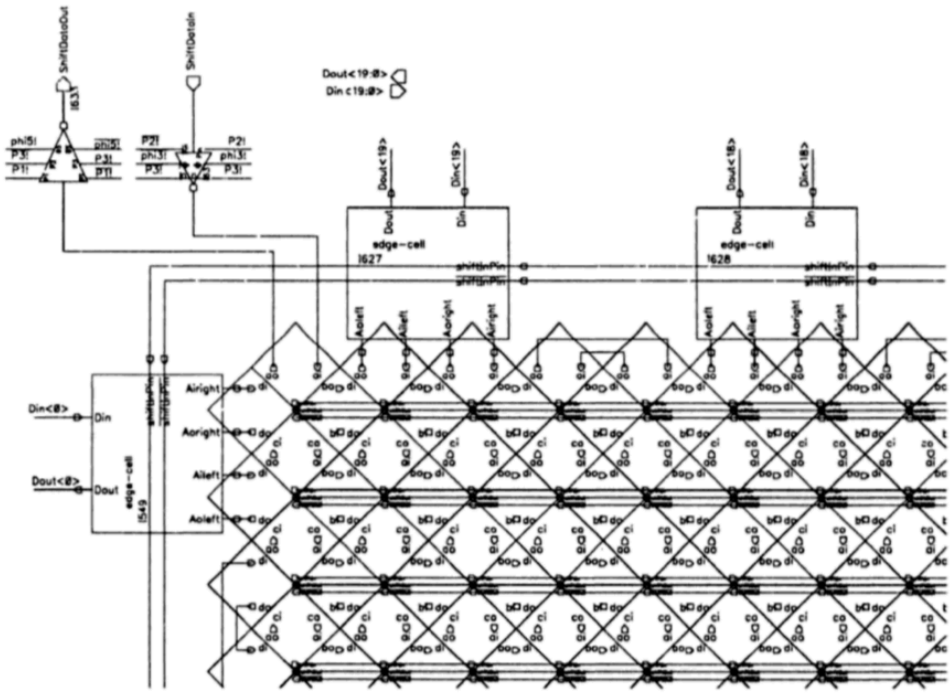

Fig. 7. One corner of a large array of processing elements.

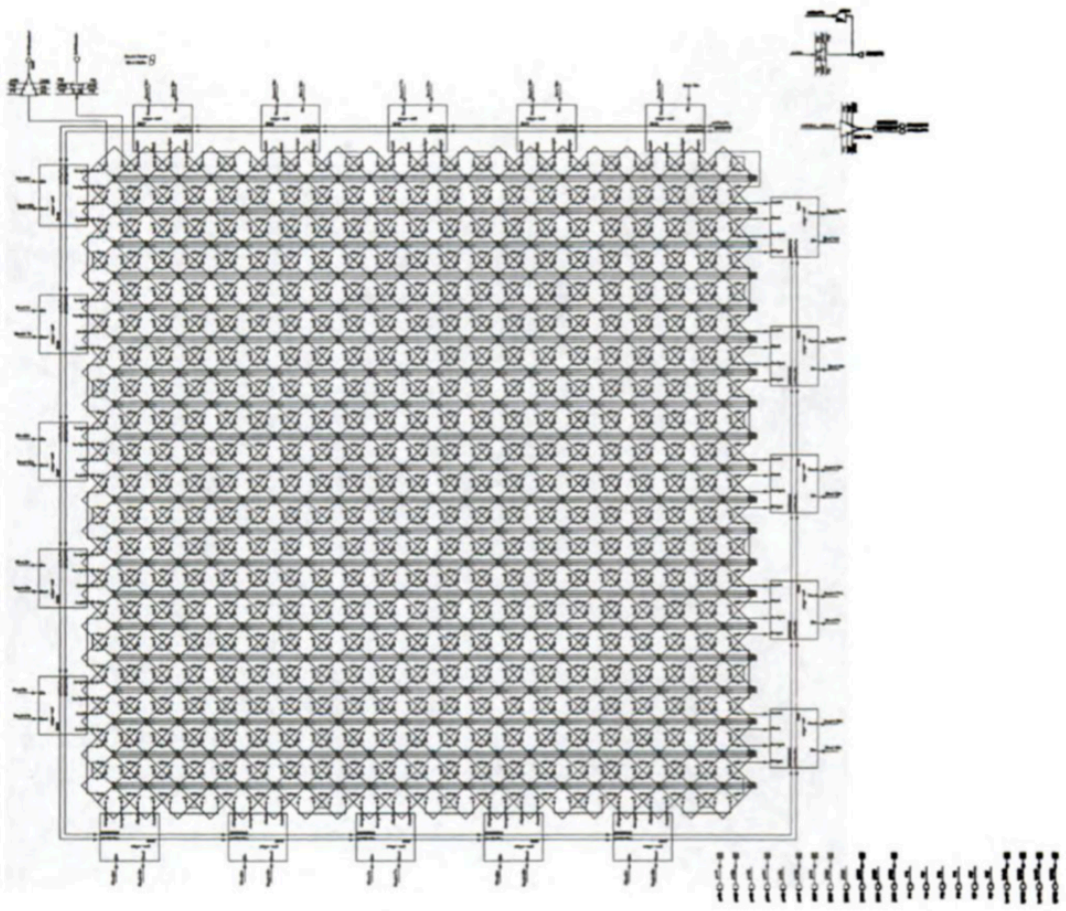

Fig. 8. A 20×20 array of processing elements.

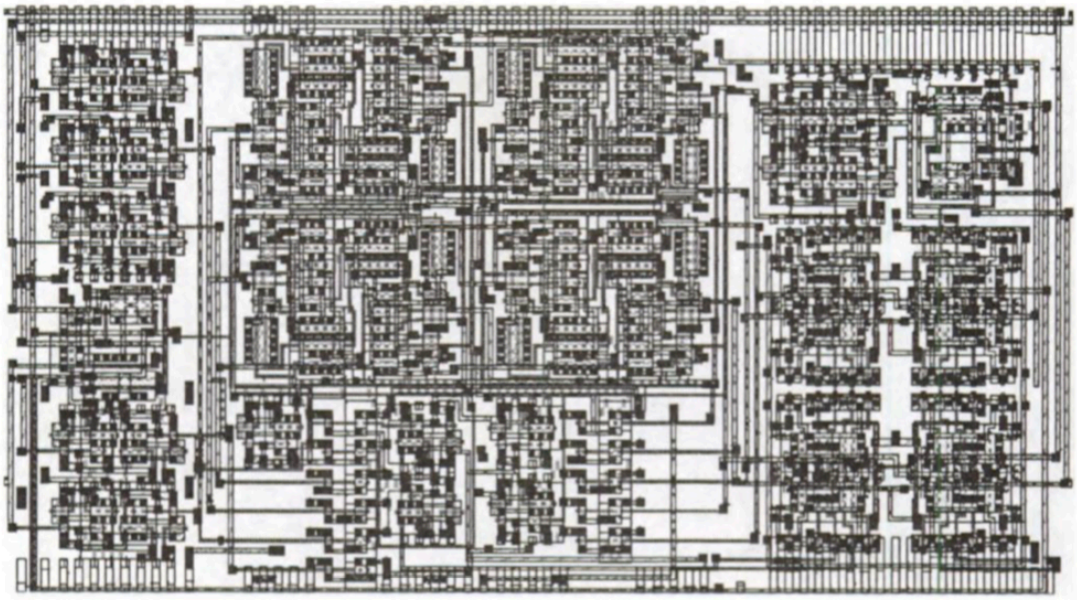

Fig. 11. Complete layout of a single cell.

pacitance C_L per load line was determined for an earlier version of our circuit, and was found to be ~170 fF.

NMOS		PMOS	
Var	Value	Var	Value
ϕ_0	1.1V	ϕ_0	0.8993
m_j	0.726	m_j	0.4905
m_{jsw}	0.2451	m_{jsw}	0.2451
C_j	$4.67\text{x}10^{-4}$ F/m^2	C_j	$8.76\text{x}10^{-4}$ F/m^2
C_{jsw}	$3.20\text{x}10^{-10}$ F/m	C_{jsw}	$2.13\text{x}10^{-10}$ F/m
t_{ox}	9nm	t_{ox}	9nm
μ_n	978.1 cm^2/V^2-s	μ_p	228.5 cm^2/V^2-s
C_{ox}	$3.89\text{x}10^{-15}$ F/μm^2	C_{ox}	$3.89\text{x}10^{-15}$ F/μm^2
k'_n	$3.80\text{x}10^{-4}$ A/V^2	k'_p	$8.889\text{x}10^{-5}$ A/V^2

Table 1. Device parameters for the HP CMOS 14 process, from HSPICE models.

In general, the standard equations that we use in our analysis can be obtained from any modern VLSI textbook, such as [7].

$$C_{db} = K_{eq_j} * C_j * AD_n + K_{eq_{jsw}} * C_{jsw} * PD_n \tag{1}$$

$$K_{eq} = \frac{-\phi_0{}^m \left[(\phi_0 - V_{high})^{1-m} - (\phi_0 - V_{low})^{1-m}\right]}{(V_{high} - V_{low})(1-m)} \tag{2}$$

$$C_g = C_{ox} W L \tag{3}$$

The value of C_L is used to determine an estimate of the maximum current being drawn from the rails. In an adiabatic circuit, determining the characteristics of the circuit during the transient is extremely difficult because both the source and the drain are changing. In adiabatic circuits, switching the rails slowly produces a smooth transition because the capacitive load has enough time to charge up with the rail. With a slow changing rail, the drain follows the source with a small lag. This lag is V_{DS}, and it will be approximately constant during most of the switching time. This analysis assumes the rails are switching slowly enough to produce a fairly constant V_{DS}, labeled as V_{DSest}. An approximation of the current shown in equation 4 is used, based on the voltage change at the output. With the assumption that the rail is changing slowly, this approximation of the current should be close to the average current during the transition. In calculating V_{DSest}, V_{GS} is approximated as its value at the halfway mark of the transition, which seems suitable for a first order approximation. Table 2 shows the results of calculating I_{est} and V_{DSest} for a variety of rise/fall times.

$$I_{est} = C_L \frac{\Delta V_{out}}{t_r} \tag{4}$$

$$I_{est} = k'_p \frac{W}{L} \left[\left(\overline{V}_{GS} - V_T \right) V_{DSest} - \frac{{V_{DS}}^2}{2} \right] \tag{5}$$

$$V_{DSest} = \overline{V}_{GS} - \sqrt{{\overline{V}_{GS}}^2 - 2\frac{I_{est}}{k_p}} \tag{6}$$

t_{rf}	I_{est}	V_{DS}	E_{sw}	P_{sw}	E_{leak}
1ns	280.5μA	0.22V	2.71pJ	113μW	9.5aJ
10ns	28.0μA	0.021V	264fJ	1.1μW	95aJ
100ns	2.80μA	2mV	26fJ	10.8nW	950aJ
1μs	280nA	207μV	2.6fJ	108pW	9.5fJ
10μs	28.0nA	20μV	260aJ	1.08pW	95fJ
100μs	2.80nA	2μV	26aJ	10.8fW	950fJ
1ms	0.280nA	200nV	2.6aJ	0.108fW	9.5pJ

Table 2. Adiabatic E_{dis} vs t_{rf}.

The equation (7), used for energy dissipation due to switching of the adiabatic circuit, was determined by integrating the power $P = IV$ over the transition. Using eq. 7 the energy per signal per cycle was calculated, and is shown in Table 2. The energy dissipated due to leakage was calculated using

$I_{\text{leak}} = 10^{-11}$A as the base leakage current through a transistor (value taken from the HP CMOS 14 HSPICE model).

The table reflects the fact that the energy due to switching is, at slow speeds, proportional to the inverse of t_{rf}, and the energy due to the leakage is proportional to t_{rf}. Equation 9 shows the total energy dissipation per operation E_{tot} as a function of t_{rf} with the proportionality constants K_{sw} and K_{leak} as needed for the switching and leakage energy terms. Using this equation the optimum t_{rf} can be found which minimizes the energy dissipated by the circuit.

$$E_{\text{sw}} = 2I_{\text{est}}V_{\text{DSest}}t_{\text{rf}} \tag{7}$$

$$E_{\text{tot}} = E_{\text{sw}} + E_{\text{leak}} \tag{8}$$

$$E_{\text{tot}} = \frac{K_{\text{sw}}}{t_{\text{rf}}} + K_{\text{leak}}t_{\text{rf}} \tag{9}$$

$$t_{\text{rf}_{\text{opt}}} = 523ns \tag{10}$$

$$t_{\text{cycle}_{\text{opt}}} = 24t_{\text{rf}} = 12.5\mu s \tag{11}$$

In comparison, implementing the same circuit using conventional CMOSrequired four stages corresponding to the three stages and the fast stage of the adiabatic circuit. The energy dissipated by the CMOS circuit is given by equation 12. The first term is the energy dissipated due to switching and the second is due to short circuit current. Equation 12 assumes worst-case switching activity (i.e. $\alpha = 1$). In conventional CMOS the reversible stages are not needed, therefore the load capacitance of the circuit will be approximately half the capacitance of the adiabatic circuit. The total energy dissipated per cycle for the conventional CMOS circuit is 21.4pJ/cycle. The propagation delay of the conventional CMOS circuit using equation 13 is 59.5ps/stage therefore the total propagation delay for the four stages is 238ps. The propagation delay for the adiabatic circuit using the same equation is 260ps/edge and there are 24 edges used to propagate through the three stages, therefore the total propagation delay is 6240ps. Table 3 is a summary of the comparison of the conventional CMOS vs. an adiabatic version. The conventional CMOS circuit has a propagation delay which is more than 26 times less than the propagation delay of the adiabatic circuit. The energy savings of $\sim$ 2000x by the adiabatic circuit over the conventional CMOS circuit is enormous. The worst-case energy dissipation for the adiabatic circuit is assuming that the power supplies are switching effectively instantaneously, in which case the same equation used for energy dissipation due to switching in CMOS can be used as an approximation for the adiabatic circuit. The worst-case energy dissipation is 10.38pJ/cycle.

$$E_{\text{tot}}/\text{cycle} = \frac{1}{2}C_{\text{L}}{V_{\text{dd}}}^2 + t_{\text{rf}}I_{\text{peak}}V_{\text{dd}} \tag{12}$$

$$t_{\text{d}} = \frac{C_{\text{L}}\left(V_2 - V_1\right)}{I_{\text{av}}} \tag{13}$$

Table 3. Performance of standard CMOS vs. Adiabatic implementations of Flattop unit cell.

Var	CMOS	Adi.	Adi./ CMOS
Min. energy (pJ/cyc)	21.4	$9.9\text{x}10^{-3}$	1/2164
Max. energy (pJ/cyc)	21.4	10.38	.5
Min. prop. delay (ps)	238	6240	26

6 Conclusion

This paper describes the world's first complete circuit- and layout-level design for a fully adiabatic and reversible universal computer. The architecture is parallel and scalable to arbitrarily large arrays, assuming power supply inputs are repeated periodically. Global timing skew is not an issue since all data interconnections are local. This paper therefore provides the first concrete example of a piece of hardware that can be programmed to perform arbitrary computations using arbitrarily little energy per operation (ignoring leakage and power supply issues). Even when actual leakage factors are taken into account, the circuit can still operate with less than one thousandth of the energy per operation of a traditional circuit implementing the same computation model. Assuming adequate power supplies can be built, our design and analysis illustrate the enormous power benefits that can be gained by computing adiabatically.

References

1. Michael P. Frank and Thomas F. Knight, Jr. Ultimate theoretical models of nanocomputers. *Nanotechnology*, 1997. To be presented at the Fifth Foresight Conference on Molecular Nanotechnology, Palo Alto, California, Nov. 1997. `http://www.ai.mit.edu/~mpf/Nano97/paper.html`.
2. Michael P. Frank, Thomas F. Knight, Jr., and Norman H. Margolus. Reversibility in optimally scalable computer architectures. In *proc. of the First International Conference on Unconventional Models of Computation (UMC-98)*, Auckland, New Zealand, January 1998. `http://www.ai.mit.edu/~mpf/rc/scaling_paper/scaling.html`.
3. J. G. Koller and W. C. Athas. Adiabatic switching, low energy computing, and the physics of storing and erasing information. In *Physics of Computation Workshop*, 1992.
4. R. Landauer. Irreversibility and heat generation in the computing process. *IBM J. Research and Development*, 5:183–191, 1961.
5. K. K. Likharev. Classical and quantum limitations on energy consumption in computation. *Int'l J. Theoretical Physics*, 21(3/4):311–326, 1982.
6. N. H. Margolus. *Physics and Computation.* PhD thesis, MIT Artificial Intelligence Laboratory, 1988.
7. J.M. Rabaey. *Digital Integrated Circuits: A Design Perspective.* Prentice-Hall, 1996. `http://infopad.EECS.Berkeley.EDU/~icdesign`.

8. S. G. Younis. *Asymptotically Zero Energy Computing Using Split-Level Charge Recovery Logic.* PhD thesis, MIT Artificial Intelligence Laboratory, 1994.
9. Saed G. Younis and Thomas F. Knight, Jr. Asymptotically zero energy split-level charge recovery logic. In *International Workshop on Low Power Design*, pages 177–182, 1994.