AFRL-IF-RS-TR-2007-171
Final Technical Report
July 2007

DESIGN AND SIMULATION OF A PROGRAMMABLE MEMORY/MULTIPLIER ARRAY USING G4-FET TECHNOLOGY

University of Notre Dame

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. AD52

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Rome Research Site Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-IF-RS-TR-2007-171 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/

CHRISTOPHER FLYNN
Work Unit Manager

/s/

JAMES A. COLLINS, Deputy Chief
Advanced Computing Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
JUL 2007

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
Nov 06 – Feb 07

4. TITLE AND SUBTITLE

DESIGN AND SIMULATION OF A PROGRAMMABLE MEMORY/MULTIPLIER ARRAY USING G4-FET TECHNOLOGY

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8750-06-1-0234

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Jay B. Brockman and Peter M. Kogge

5d. PROJECT NUMBER
AD52

5e. TASK NUMBER
NT

5f. WORK UNIT NUMBER
RD

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Notre Dame
Office of Research
511 Main Building
Notre Dame IN 46556-5602

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington VA 22203-1714

AFRL/IFTC
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2007-171

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# AFRL-07-0048

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Field-programmable and mask-programmable gate arrays can greatly reduce the non-recurring costs of ASIC development by reusing both masks and physical design effort across many designs. The downside of gate arrays is that they result in suboptimal implementations in terms of area, speed, and power. In addition, there is very little flexibility in converting logic to memory or vice versa, a problem of increasing importance as memory-intensive applications gain in importance. To address these issues, we have investigated the design of a novel gate array structure based on G4-FET devices, which combine SOI CMOS and JFET technologies, and that can be biased to function as either a not-majority logic gate, a router/multiplexer, or as a DRAM cell. To demonstrate the potential of G4-FETs for gate arrays, we have designed a memory/multiplier array that consists of an array of configurable cells built from G4-FETs and a mask-configurable interconnect that may serve as either a multiply-accumulate circuit or as a memory array.

15. SUBJECT TERMS
FPGA, G4-FET

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Christopher Flynn
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UL	21	**19b. TELEPHONE NUMBER (*Include area code)***

Table of Contents

List of Figures

1 Executive Summary

Field-programmable and mask-programmable gate arrays can greatly reduce the non-recurring costs of ASIC development by reusing both masks and physical design effort across many designs. The downside of gate arrays is that they result in sub-optimal implementations, resulting in increased chip area and power, reduced clock rates, and increased recurring costs. In addition, there is very little flexibility in converting logic to memory or vice versa, a problem of increasing importance as memory-intensive applications gain in importance.

To address these issues, we have investigated the design of a novel gate array structure based on G4-FET technology. A G4-FET (4-Gate Field Effect Transistor) is an innovative 4-gate device that can be fabricated on a standard SOI CMOS (Silicon-on-Insulator, Metal-Oxide-Semiconductor) process that combines JFET (Junction Field-Effect Transistor) and MOSFET (Metal-Oxide-Semiconductor, Field-Effect Transistor) characteristics, and that can be biased to function as either a not-majority logic gate, a router/multiplexer, or as a DRAM cell. To demonstrate the potential of G4-FETs for gate arrays, we have designed a memory/multiplier array that consists of an array of configurable cells built from G4-FETs and a mask-configurable interconnect that may serve as either a multiply-accumulate circuit or as a memory array.

2 Background and Problem Definition

As the geometries of integrated circuits (ICs) continue to shrink below 90 nanometers, the non-recurring expense (NRE) of developing an IC rises sharply. One factor driving the NRE is mask cost, which is now approaching $4M for a complete set, as shown in Figure 1. Another factor, even more significant than the mask cost, is the cost of computer-aided design (CAD). One reason for the greater CAD expense is that the interactions between physical layers become far more complex as the distance between components decreases, and as a result, a much more sophisticated analysis must be performed when placing and routing components to ensure that both performance and signal integrity meets specifications. In addition, because the spacing of patterned elements is on a scale with the wavelength of the projected light, mask design must now take into consideration the neighborhood surrounding each polygon in determining its final shape. As a result, each unique, small region of a custom IC constitutes a complex physical design problem in itself. Further, as the density of integration and number of devices per IC increases, this physical design problem becomes much worse, to the point that full custom design is rapidly becoming intractable, and the number of ASIC starts is actually decreasing, also as shown in Figure 1.

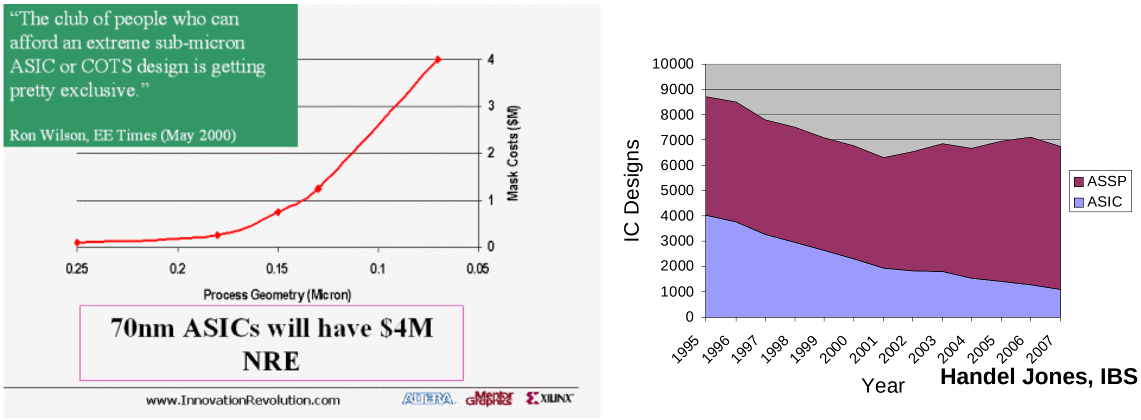


Figure 1: Increase in mask costs and decline in ASIC and ASSP (Application-Specific Signal Processor) starts

In response to high NRE costs of custom IC design, *programmable gate arrays* have been gaining popularity. A programmable gate array is a chip composed of a regular array of primitive cells, implemented using a base set of mask layers, which may then be personalized with additional (cheaper) back end of the line mask layers—or by post fabrication electrical programming in the field—to a particular design.

Gate arrays reduce NRE in two ways, both of which involve amortizing costs across many designs. The first area of savings is reuse of masks. Because gate arrays are composed of a set of base cells, the masks involved in defining the cells, especially the lower level ones most susceptible to the deep sub 90 nm effects, are common across all designs. For a mask-programmable gate array, a few additional metal or via masks are typically required to personalize a design. In a field-programmable gate array (FPGA), designs are personalized by either storing configuration data in memory cells or via fuses in the field after fabrication is complete, and thus all masks are reused by all designs. The second area of savings is the simplification of design verification and analysis that arises from the regularity of designs. Because the layout of devices and interconnect is restricted to a reduced set of regular patterns, both the final printed lithography and the delays through signal paths are more controlled and predictable than they would be in a free-form custom design. Thus by following a disciplined, structured design flow, it is possible to greatly simplify the complexity and hence cost of physical design.

Despite their tremendous advantages in reducing NRE, gate arrays also have their downside. Whereas custom or standard cell-based ASICs allow free selection and layout of components, gate array-based designs consist of "packing" a circuit into a predetermined set of resources, which consists of dedicated logic cells, memory, and interconnect, as shown in Figure 2. In particular, because the amount of memory and logic resources is fixed, it is difficult to convert these chips between logic and memory intensive applications.

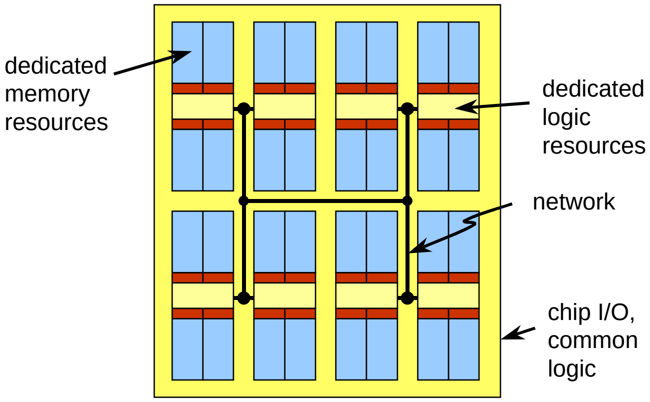


Figure 2: Dedicated resources on a gate array chip

Conventional CMOS mask and field-programmable gate arrays do provide a minimal amount of flexibility for converting logic resources to memory. Mask programmable circuits can sometimes be programmed to form a latch cell, at a density significantly less than pure SRAM and orders of magnitude less than 1-transistor DRAM. FPGAs already contain SRAM, but most of these bits are usually fixated into controlling the surrounding logic. Again, if memory is needed we are left with the unattractive proposition of loading SRAM bits to configure logic to implement a latch bit.

An innovative device, called the G4-FET, may revolutionize gate array design by providing unprecedented levels of flexibility for configuring circuit blocks. A G4-FET is a 4 gate transistor that combines both JFET and MOS characteristics in a single device that may be fabricated in a standard silicon-on-insulator (SOI) process. In doing so, it enables the conducting channel to be controlled vertically through MOS gates, as well as horizontally, through junction gates. In terms of its application to gate arrays, depending upon how it is biased, a single G4-FET can serve as either a *not-majority logic gate* or as a *charge storage-based memory cell*, similar to a DRAM cell. In this report, we provide a preliminary investigation of the feasibility of using G4-FET technology for implementing a mask-programmable gate array that can be configured either as logic or memory. Specifically, we demonstrate this by implementing both an array multiplier and a DRAM array on the same array of devices.

3 G4-FET Devices and Circuits

3.1 Structure and Operation of G4-FET Device

Figure 3 illustrates the structure of an n-channel G4-FET device. It is a majority carrier, buried channel, accumulation mode device where the source, drain, and body/channel regions are all n-type. It has two vertical MOS gates: a conventional polysilicon top gate, as well as the substrate, which can act as a bottom gate. In addition, it has two lateral JFET gates that form PN junctions with the channel region.

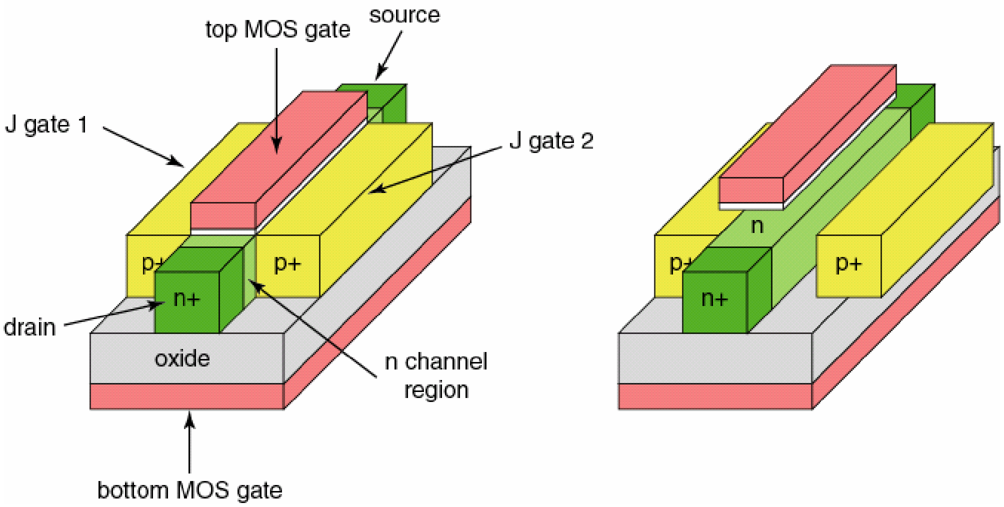
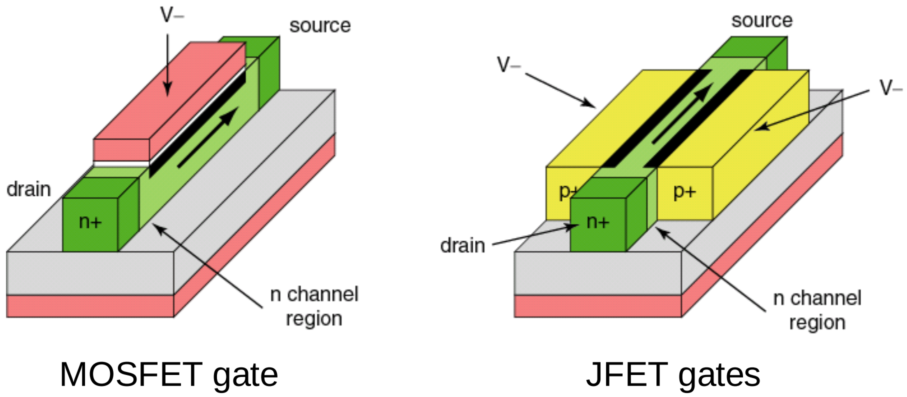


Figure 3: Structure of n-channel G4-FET device

Applying an appropriate bias to each of the four gates changes the shape of the channel, controlling the conduction path. Figure 4 illustrates the effects of the MOSFET top gate and the lateral JFET gates on the channel. For an N-type device, a negative voltage applied to the top gate depletes the channel region below the gate of carriers. Similarly, negative voltages applied to either of the JFET gates causes a widening of the PN-junction depletion region, further narrowing the channel.



MOSFET gate

JFET gates

Figure 4: G4-FET top MOS gate and side JFET gates control channel current

By using these bias voltages in combination, the device can be switched between linear, saturated or cutoff regions of operation. Notre Dame characterized the IV (Current & Voltage) characteristics of a set of N-type G4-FET devices fabricated by Honeywell and packaged by JPL [3]. In this experiment, the JFET gates were held at either 0 or –1 volt, while the top MOS gate was swept across the range of –3 to 3 volts. The results of the characterization are shown in Figure 5. When a positive voltage is applied to the top gate (show at 2V and 3V in the Figure), carriers accumulate under the top gate and channel conduction is high. The normal operation of the device for a digital circuit, however, is when the top MOS gate is at zero or negative voltage, shown as the bottom set of curves in the Figure. At –3V on the top MOS gate, the current is reduced substantially and with further reduction of the JFET gate voltages, it would cut off completely. In general, the

IV characteristics of G4-FET devices are very sensitive to the printed dimensions of the channel region, and a planned test chip will examine a range of device sizes to determine this sensitivity, as well as the "sweet spot" for the dimensions for a given process technology.

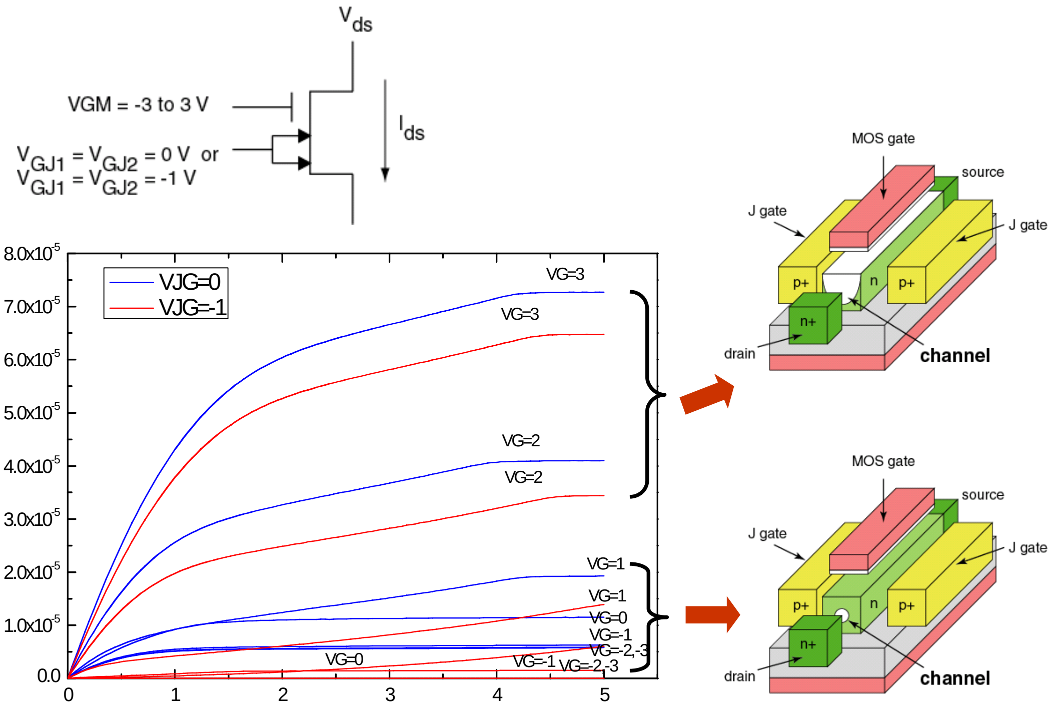


Figure 5: N-channel G4-FET IV characteristics

In order to simulate the behavior of G4-FET circuits, we developed Spice models from the characterized devices. We developed two types of models: the first a simple switch-resistor model, and the second, a more accurate non-linear analog simulation model. In the switch-resistor model, we approximate the conducting channel as a bar of resistive material, whose dimensions are determined by the terminal bias configuration, as illustrated in Figure 6. The length of the bar corresponds to the distance between the source and drain, while the height is a function of the MOS-gate voltage and the width is a function of JFET gate voltages. The resistivity of the bar was determined by curve-fitting the characterized devices.

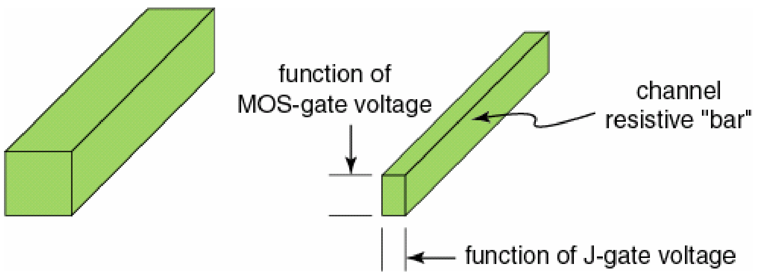


Figure 6: Resistive model of G4-FET channel

The detailed model consists of a hybrid of existing Spice MOS and JFET models. Using the model involves two passes of the Spice simulator: the first to determine the region of operation, and the second to invoke the proper configuration of the MOS or JFET model.

3.2 Majority Gate Logic

To date, only N-type G4-FET devices have been fabricated. In order to build combinational logic devices, the Honeywell test chip couples a G4-FET pull-down device with p-type CMOS resistive load (pseudo-NMOS) device, together with output level-shifters, as shown in Figure 7.

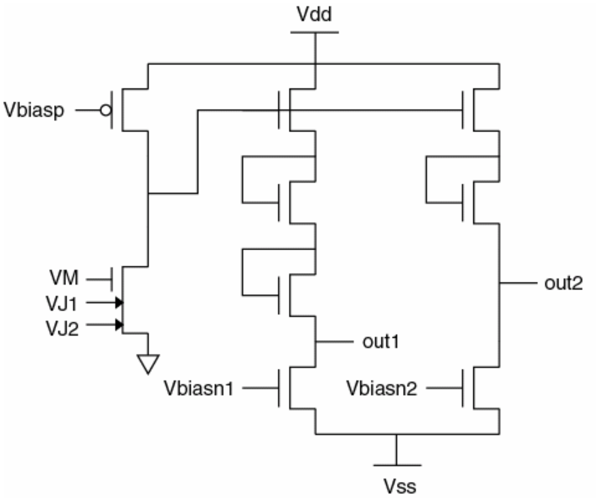
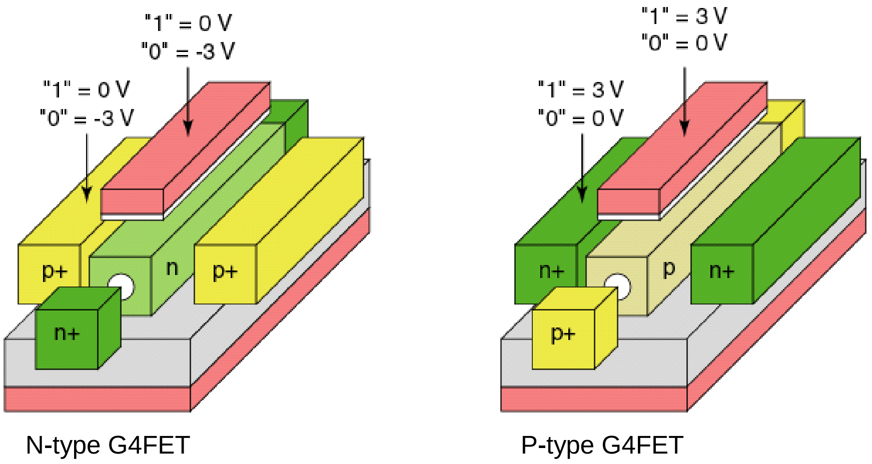


Figure 7: Honeywell/JPL G4-FET test circuit

In this configuration, the G4-FET device naturally forms an inverse majority gate. Based on the characterization of the devices on the Honeywell test chips, we assume that a logic 0 corresponds to a voltage of –3V and a logic 1 corresponds to a voltage of 0V. If at least two of three gates of the G4-FET—the MOS gate or the two JFET gates—have a logic 1 or high voltage asserted, then the device will have a conducting channel and the output will be pulled low. Conversely, if at least two of the three gates have a logic 0 or low voltage asserted, then the channel region will be fully depleted of carriers and the G4-FET will be turned off, so that the output will pull up to a high voltage with a logic 1 on the output.

The greatest disadvantage of a resistive load device is that it dissipates static power when the G4-FET pull-down device is conducting. This power consumption would be a showstopper as compared to conventional CMOS, which only dissipates static power during a switching transition. In order to eliminate the static power dissipation, we propose the development of a p-type G4-FET device for building complementary circuits. One of the problems in doing so is the difference in operating voltage for the two devices. As shown in Figure 8, while the n-type device has high and low voltages of 0V and -3V, the p-type device would have high and low voltages of +3V and 0V.



N-type G4FET

P-type G4FET

Figure 8: N-type and P-type G4-FET devices

In order to make the devices compatible, it will be necessary to shift the terminal voltages. One way of doing this is by explicitly adding voltage-shifting devices as was done in the Honeywell n-type G4-FET test circuit. Another possibility is to adjust the bias on the bottom-gate/substrate terminal. As shown in Figure 9, for example, applying a sufficiently large positive voltage to the substrate contact of an n-type G4-FET would shift the values of logic levels upward. Additional evaluation will be required on a new test chip to characterize by how much these levels can be shifted. Assuming that we can use this technique to make the logic voltage levels of the n-type and p-type devices to be compatible, we could then construct a complementary G4-FET inverse majority gate, as shown in Figure 9, where Vbiasn and Vbiasp are the substrate bias voltages for shifting the logic levels.

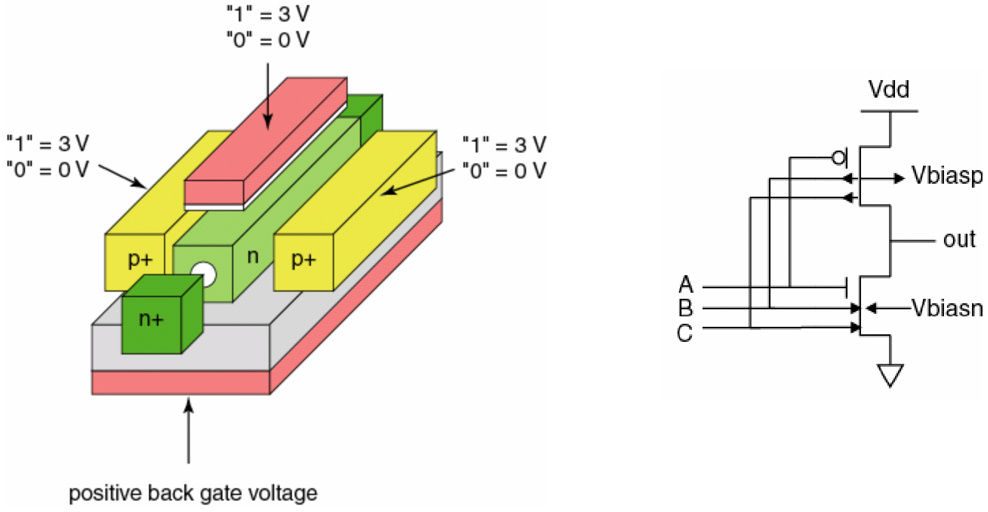


Figure 9: Using back bias to shift logic levels to create complementary G4-FET inverse majority gate.

The inverse majority gate is logically complete, and can be used to compose and Boolean function. Figure 10 shows the truth table for the inverse majority function and basic logic gates that can be formed from it. On its own, the inverse majority is the complement of the carry out function for a full adder. By setting two of the inputs to a 1 and 0, it forms an inverter, and by setting the third input to either a 1 or 0, it forms a two-input NAND or NOR gate.

a	*b*	*c*	output
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

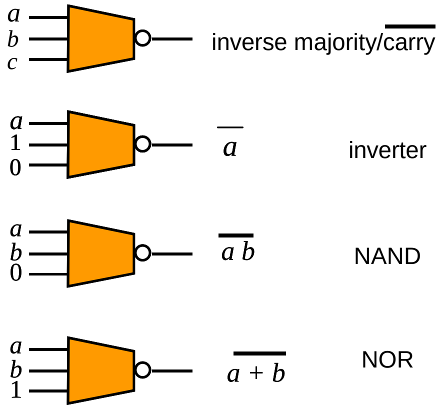


Figure 10: Inverse majority gate and derived logic functions

Figure 11 shows the implementation of a full adder, which requires only 3 inverse majority gates and 2 inverters. For comparison, the typical CMOS implementation requires 28 transistors. A complete power analysis of G4-FET logic has not yet been done, but the dynamic power dissipation should be comparable to that of CMOS and perhaps slightly less, based on the number of driven gates.

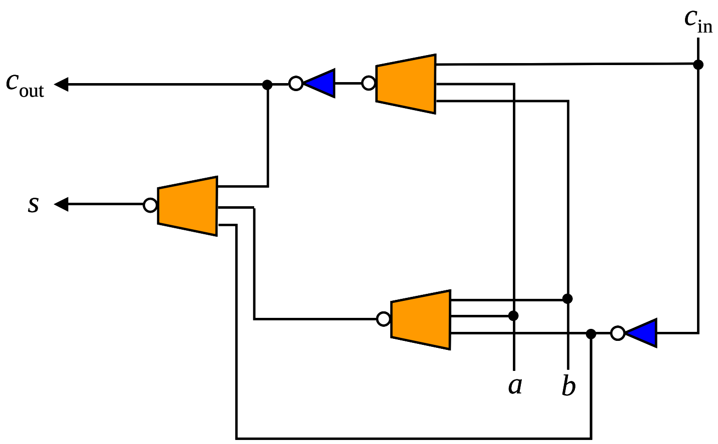


Figure 11: Full adder made from inverse majority gates

3.3 G4-FET Memory

In addition to functioning as a logic switch, a G4-FET device can also be biased to operate as charge-storage memory cell, similar to a DRAM cell. In this section, we first briefly review basic issues in classic memory design, and then introduce G4-FET memory design.

Figure 12 illustrates the organization of a conventional memory array. Individual memory cells are arranged into a grid of rows called wordlines and columns called bitlines. In order to read a memory cell, bits from the address cause exactly one row decoder to activate one wordline, at which point each of the cells on that wordline indicate whether they contain a 1 or a 0 on their corresponding bitline. Depending upon the memory technology, this information may be represented either by a change in bitline voltage or a current flow. Some memory technologies also contain a complementary bitline that will have the opposite logical information of the “true” bitline, which can speed up and simplify sensing. To write data to a cell, once a wordline is selected, a voltage is forced onto the bitlines and into the cell.

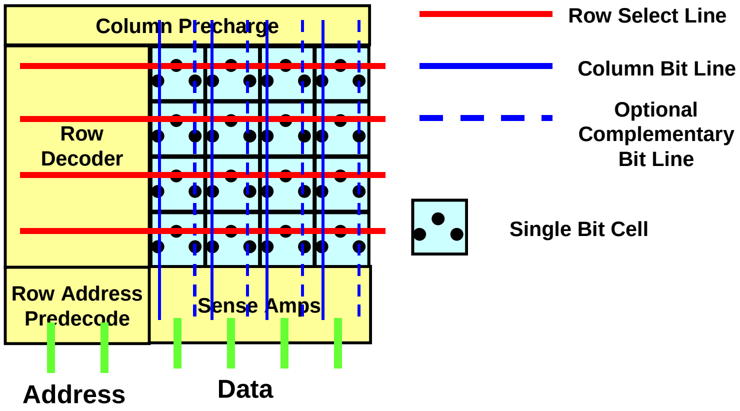


Figure 12: Conventional memory array organization.

There are two dominant CMOS random access memory technologies, static RAM (SRAM) and dynamic RAM (DRAM). As shown in Figure 13, an SRAM uses positive feedback to store a 1 or 0 between two cross-coupled inverters, and uses 6 transistors. A DRAM stores data as charge on a capacitor, and requires a single transistor and a capacitor. As also shown in Figure 13, the capacitor in a DRAM is a complex structure, most commonly formed as a trench into the silicon wafer.

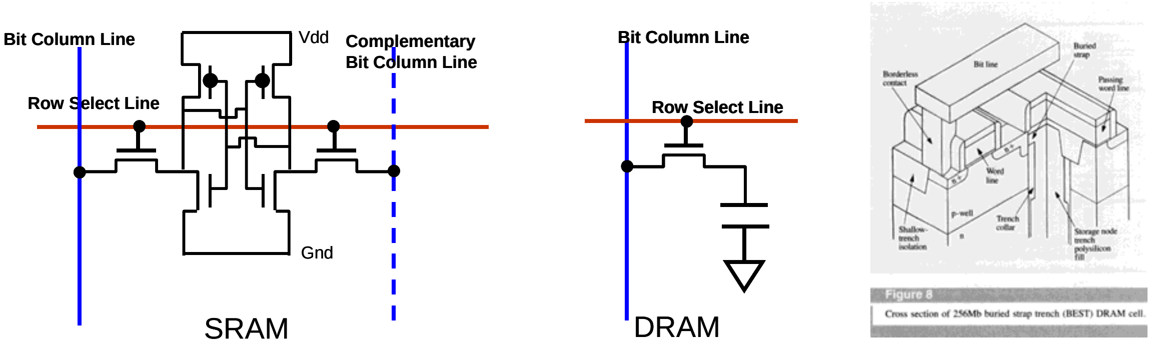


Figure 13: SRAM and DRAM cells.

Figure 14 illustrates the configuration and operation of a memory cell based on an n-type G4-FET device. Recall that when we used an n-type G4-FET as a logic device, the source and drain were the n-type regions on the "front" and "back" (as drawn in Figure 3) of the body, while the p-type regions on the sides served as the JFET gates. When using this device as a memory cell, we think of it as an SOI enhancement-mode PMOS device, where the p-type "sides" are the source and drain. A memory cell is "programmed" by storing charge across the depletion region of the source junction, which increases the depletion region width, cutting off the conducting channel formed by the body of the device and the two n-type regions at the "front" and "back" of the body. Unlike a DRAM organization, which uses a single wordline and single bitline for both reading and writing, the G4-FET memory uses separate wordline/bitline pairs for each of these operations. The top MOS gate is connected to the write wordline and the p-type drain is connected to the write bitline. The read wordline is connected to the n-type region at the back of the body, while the read bitline is connected to the n-type region at the front. The p-type source is left floating.

To program an n-type G4-FET memory cell, we assert a low voltage on the write wordline (MOS gate) and a high voltage on the write bitline (p-type drain). This causes charge to build up across the depletion region, shrinking the channel and increasing its resistance. To read a cell, a positive voltage is asserted on the read wordline. Depending upon whether the cell is programmed or not, the resulting current flow through the read bitline will either be low or high.

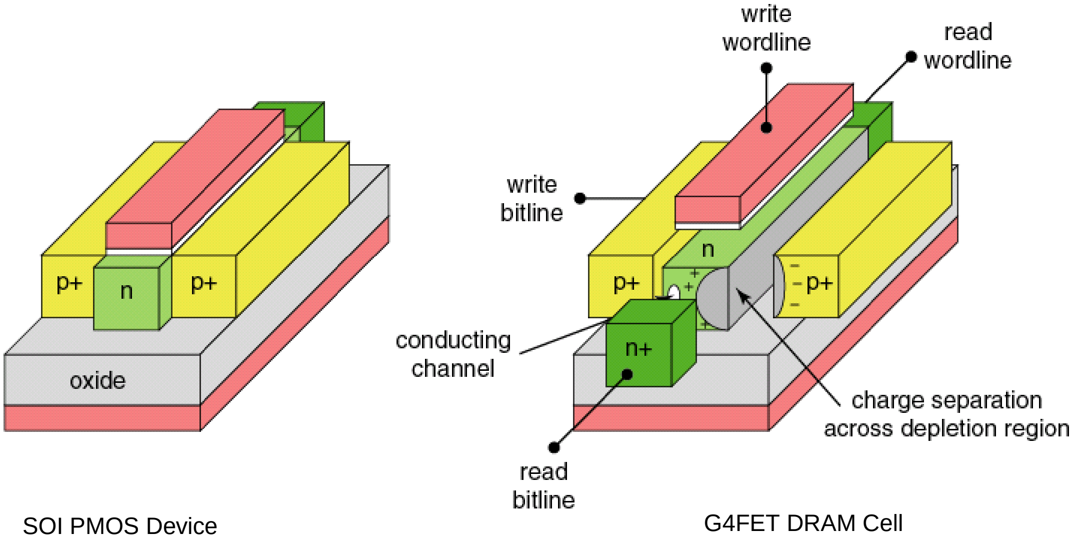


Figure 14: G4-FET DRAM cell.

3.4 G4-FET Layout

Figure 15 illustrates the layout of a G4-FET device. The actual body of the device, which lies underneath the MOS gate, is shown as the tiny red region in the center of the layout.

In order make it possible to connect to the device, the geometries of the other mask areas are much larger. The top gate itself, which is fabricated on the first-level polysilicon layer, is connected to a sideways H-shaped set of tabs that are fabricated on a second-level polysilicon layer.

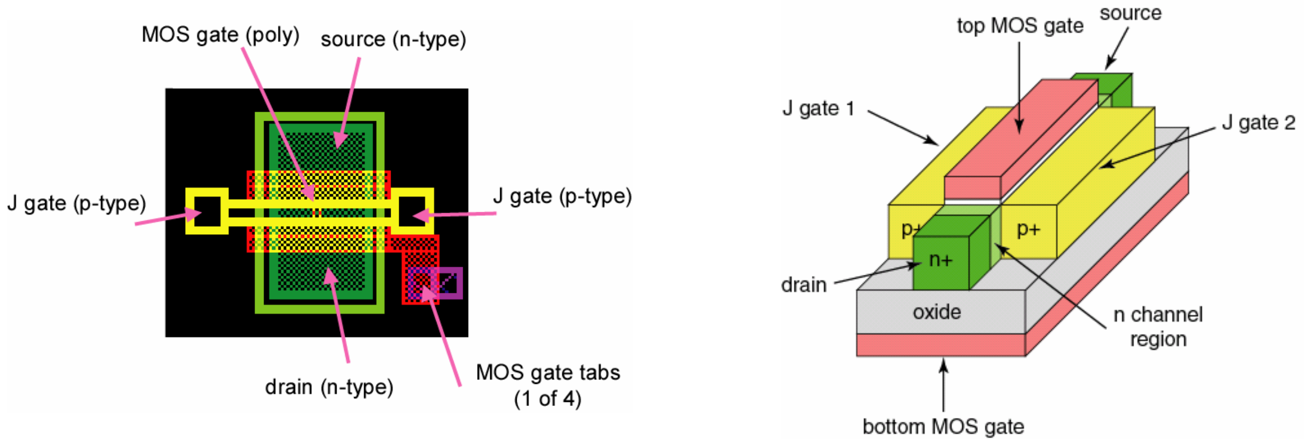


Figure 15: Mask layout of G4-FET device

Given this basic device topology, we designed a layout for a full-adder cell, which is the main building block of an array multiplier. Figure 16 shows the layout of a complementary G4-FET adder cell, based on the circuit design from Figure 11, as compared to a conventional CMOS full adder, using the same design rules for line widths and spacings. Both layout were hand-optimized to minimize layout area. While the G4-FET adder only has 6 G4-FETs and 4 MOSFETs, as compared to 28 MOSFETs in the CMOS adder, its overall area is not substantially less because of the necessary interconnect. On the other hand, the G4-FET layout is much more regular, which is a tremendous advantage for lithography and manufacturability. All of the polysilicon lines have the same orientation, and the cell architecture itself is modular. If we imposed the same constraints on the direction of the polysilicon lines on the CMOS adder layout, its area would increase substantially.

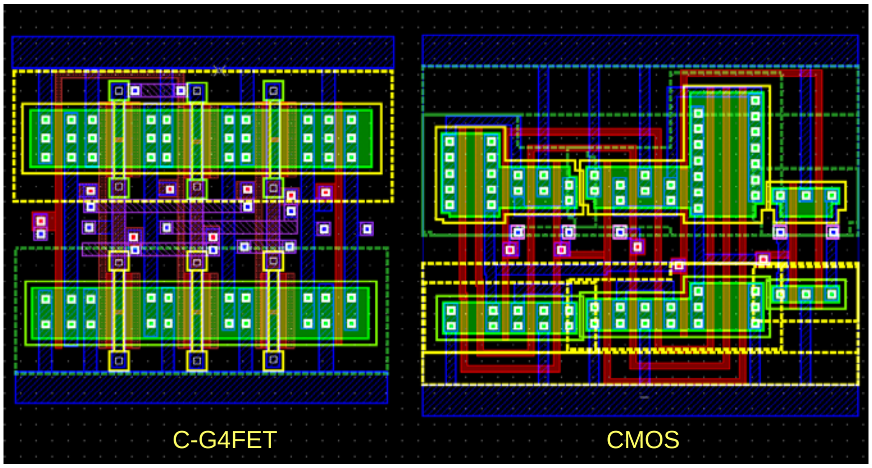


Figure 16: Optimized layouts of full adder using G4-FET and conventional CMOS technologies

4 Mask Programmable G4-FET Gate Array

4.1 Field Programmable versus Mask Programmable Gate Arrays

A field programmable gate array (FPGA) consists of an array of basic cells, with fixed wiring channels interconnected by programmable switchboxes. As a result of these constraints, resources in gate arrays are often poorly utilized, leading to excessive chip area and slower clock rates. For example, a typical FPGA cell contains one or more lookup tables (LUTs), a flip-flop and some multiplexers, with 60 or more transistors per cell. If all that is needed is a single 2-input NAND gate, however, this could be implemented as custom CMOS cell with only 4 transistors. Most of the inefficiency of FPGAs stems from the fact that configuration data is stored in SRAM cells, which greatly increase the transistor count, and hence the area and leakage power.

Much of this can be overcome by using mask-programmable gate array technology. Figure 17 shows a comparison of a lookup table implemented using both SRAM cells or a via mask to implement the configuration. Whereas the SRAM based implementation has more than 50 transistors, the via-programmable version has only 14 transistors. Because it requires a custom mask for each design, however, the via-programmable LUT has greater non-recurring engineering (NRE) than the field-programmable version. Thus, the choice of mask- versus field-programmable gate arrays represents a tradeoff between recurring and non-recurring costs. Both types of gate arrays have their place, depending upon the volume of the product and other factors.

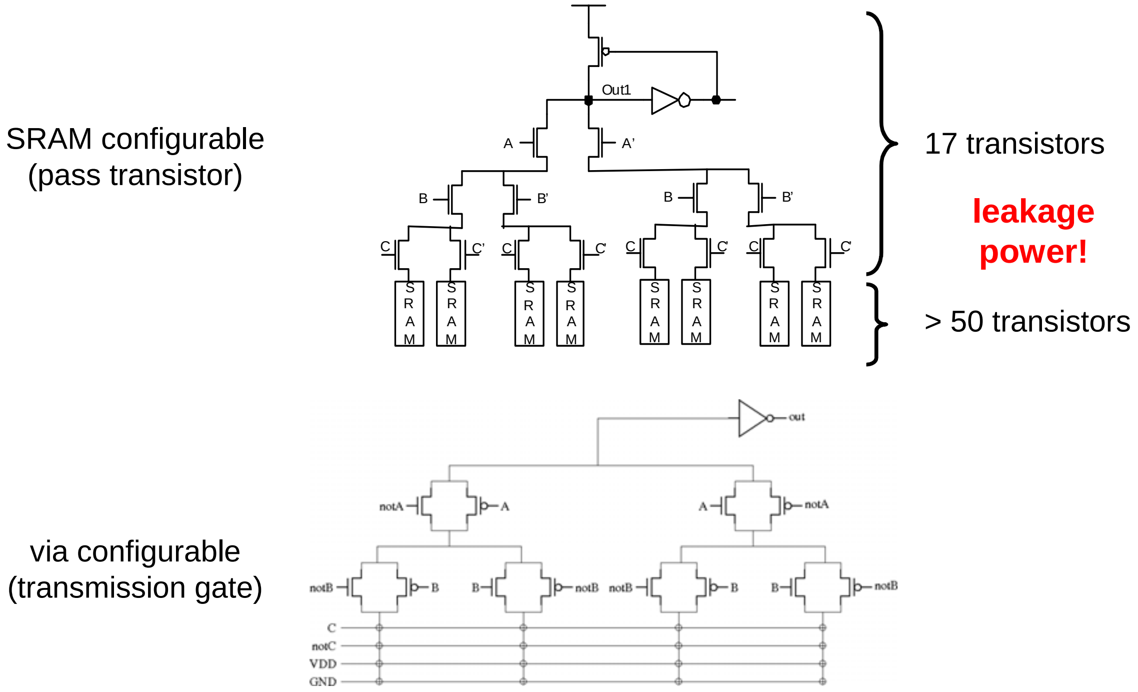
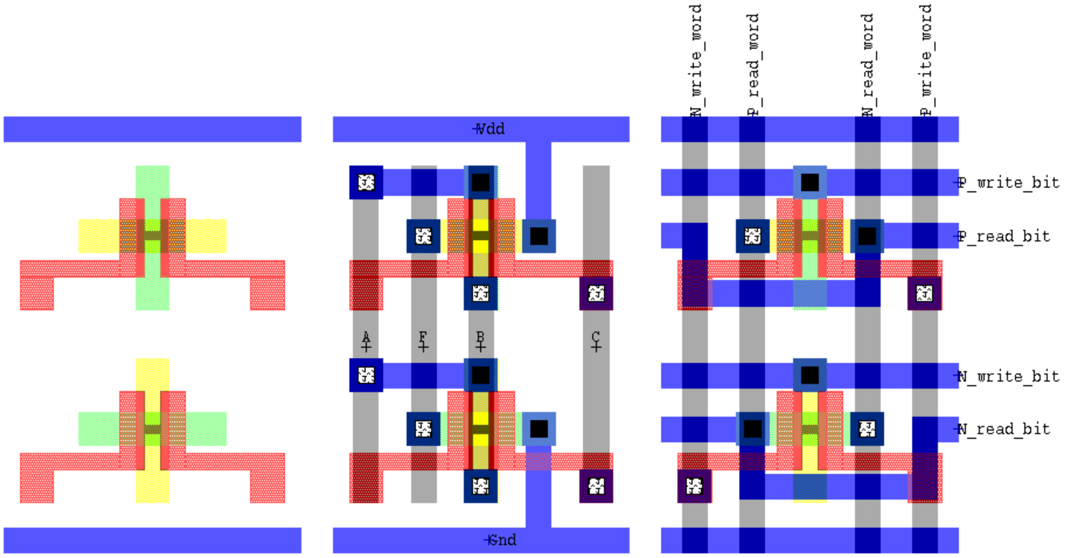


Figure 17: Field-programmable and mask-programmable implementations of a LUT

G4-FET technology opens new possibilities for the design of gate arrays. Rather than using a LUT as a basic building block, with G4-FETs a natural choice is to look at building logic from majority gates. Zhang et. al. [6] show that synthesis using majority gates directly results in up to a 68 percent reduction in gate count, with an average of over 20 percent, across the MCNC (Microelectronics Center of North Carolina) benchmark suite versus the conventional approach of converting 2-input logic gates to majority gates. In the Zhang paper, they considered each of the 256 possible 3-input logic functions and found the most efficient mappings to majority gates. We believe that with G4-FET logic, even greater efficiencies might be achieved because of the ability to use series and parallel connections between G4-FET devices to create additional logic gates, as is done with conventional CMOS devices.

4.2 Primitive Cell Design and Configuration

Using the cell architecture of the G4-FET adder as a guide, we developed metal mask-programmable G4-FET gate array that, depending upon the wiring overlay, could serve either as logic or as a memory array. Figure 18 illustrates the layout of the primitive cell template, as well as the cell configured as a majority gate and as 2 memory cells. The template contains the mask patterns for the critical device layers. Configuration as a majority gate or memory cell uses only metal 1 and metal 2. The memory cell contains 2 bits from the n-type and p-type devices, with separate pairs of wordlines and bitlines for each. In this arrangement, the wordlines are vertical wires on metal 2, while the bitlines are horizontal wires on metal 1. When the memory cells are tiled, they form two interleaved arrays, with alternating rows (bitlines) of n-type and p-type cells.



template majority gate memory (2 bits)

Figure 18: G4-FET gate array primitive tile configured as inverse majority gate and memory cell

4.3 Configuration of Gate Array for Multiplier or Memory

As illustrated in Figure 19, a multiplier is a 2-dimensional array of single-bit multiply-add cells, each of which contains a full-adder and a single-bit multiplier, which is simply an AND gate.

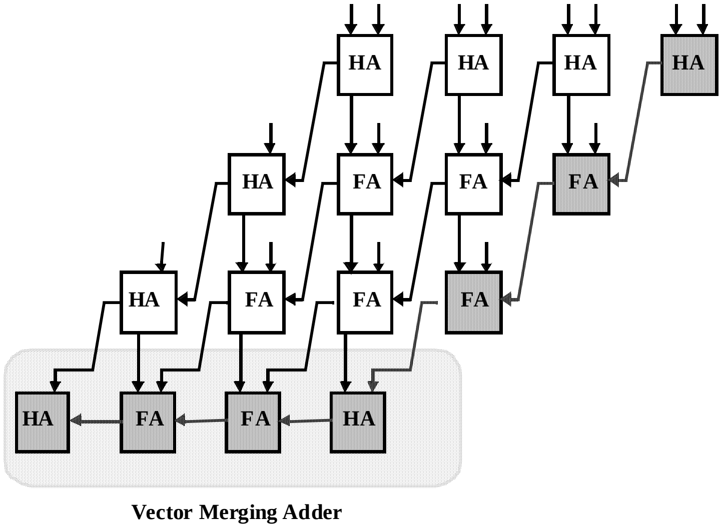


Figure 19: Array multiplier

As shown earlier in Figure 11, a full-adder can be implemented with 3 majority gates and two inverters. Figure 20 illustrates the layout of a multiply-add cell. It is composed of 6 minimal logic tiles, each configured as either a majority gate, and gate, or inverter using metal 1 and metal 2 interconnect. Alternating rows of tiles are mirrored vertically, so that they can share common Vdd and ground busses. The configured tiles are then wired together to form the multiply-add cell using metal 3 and metal 4 (and some metal 2 where convenient to simply run vertical wires). We carefully placed the ports on the perimeter of the multiply-add cell so that all interconnections to form the complete multiplier can be made by abutment, with no need of a separate custom wiring overlay. Figure 20 shows the placement of these ports, which consists of the multiplier and multiplicand signals (X and Y), the carry propagation signals (C), and the sum propagations signals (S).

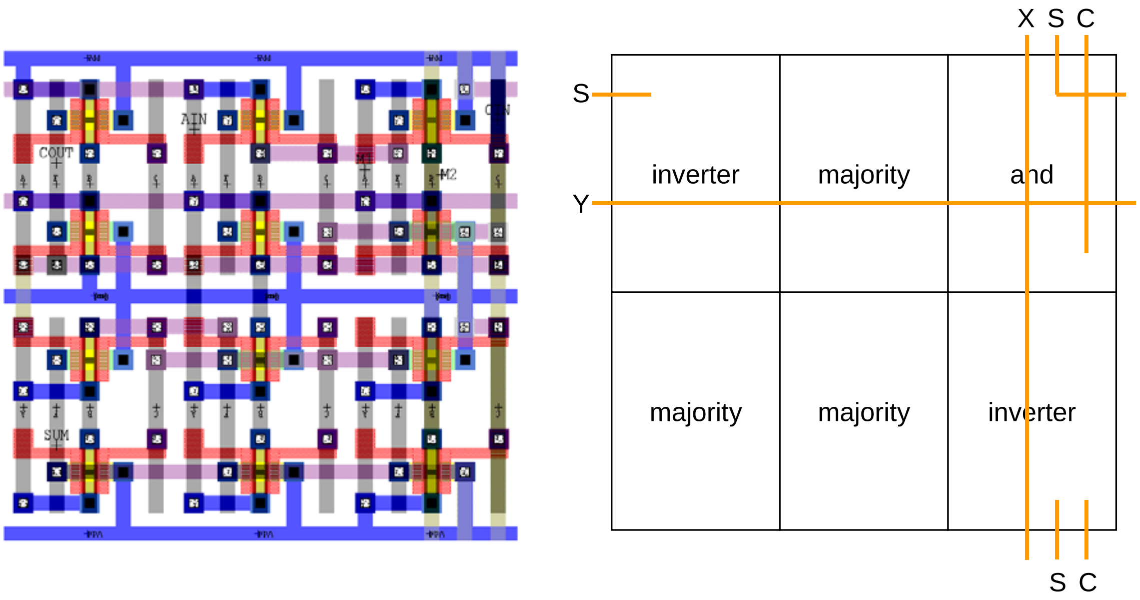


Figure 20: Full adder configured from G4-FET primitive tiles

Figure 21 shows a complete 4x4 multiplier array next to a memory array with the same dimensions. The memory contains 8 rows and 12 columns of 2 bit cells, for a total of 192 bits.

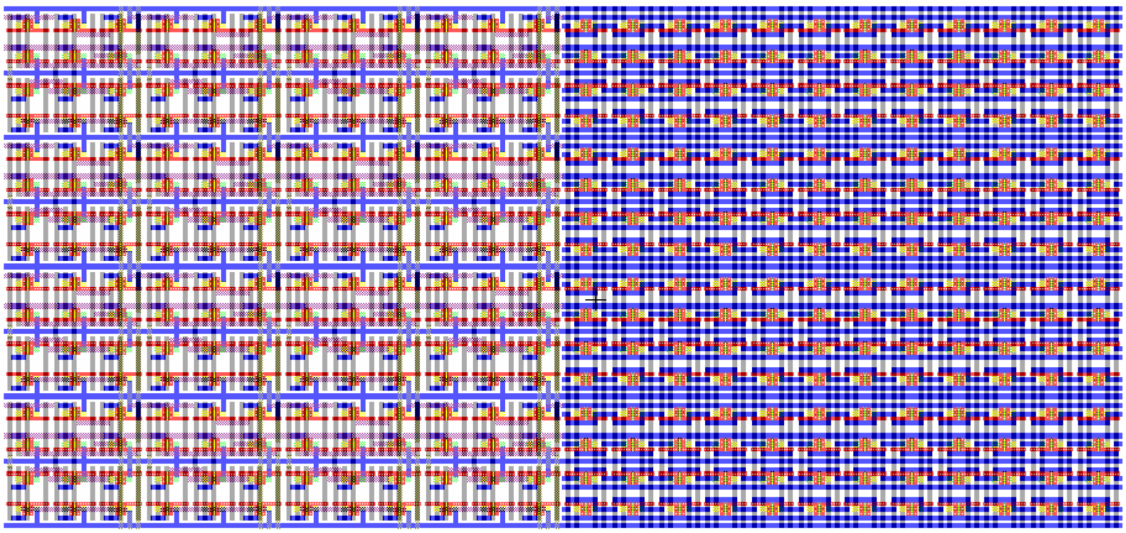


Figure 21: Comparison of 4x4 multiplier (left) and 192-bit memory array (right) configured from G4-FET gate array tiles.

Table 1 shows the comparative cell areas for different memory technologies, normalized to the MPU (Microprocessor Unit) printed gate length for 2007 from the ITRS (International Technology Roadmap for Semiconductors) [4]. Based on our layout from Figure 21, G4-FET RAM implemented from configurable cells has a greater density than SRAM, although it is still considerably less dense than DRAM.

Memory Technology	Cell Area (gate length)^2
SRAM	255.1
G4-FET RAM	**210.0**
FeRAM	170.1
MRAM	90.7
PCRAM (NMOS)	30.6
DRAM	18.1
PCRAM (BJT)	15.3
FLASH	7.4

Table 1: Comparative memory cell areas (ITRS 2007 year values)

5 Conclusions and Future Work

In this project, we implemented a mask-programmable gate-array structure using G4-FET devices, and showed that it could be configured efficiently as either a multiplier array or as memory. Not only did we find that G4-FETs are well suited for gate arrays, we also believe that a gate array is probably the preferred implementation technology for G4-FETs. We base this conclusion on several observations:

- Because a G4-FET packs more logical capability into a single device than does a MOSFET, logic circuits implemented in G4-FET technology contain fewer devices than do their MOS counterparts. Thus, the irregular device layout that one often finds within CMOS standard cell libraries to save area would have much less benefit for G4-FET designs.
- Because G4-FET devices have a greater fan-in than CMOS devices (3 logic input gates rather than 1), the layouts are even more dominated by interconnect that are CMOS layouts. As a result, the routing architecture must be carefully managed, and a structured grid greatly simplifies this process.
- Dense memories require a regular grid structure with wordlines and bitlines. If G4-FETs are to be configurable as either logic or memory, they *must* be arranged in an array pattern.

As we noted earlier in the report, thus far only n-type G4-FETs have been fabricated. In order to develop a circuit technology that will be competitive with CMOS in terms of area and—more importantly—power, it will be necessary to develop a complementary p-type device. As a result of this project, we provided input to JPL for a test chip that is being fabricated by Honeywell that contains both p-type and n-type of varying sizes that will enable us to determine if this is viable. In particular, we need to determine if it is possible to shift the logic levels of the devices by biasing the back "4th gate" of the G4-FETs so that both the p-type and n-type devices can use the same voltages for a logic 0 and 1. This test chip is expected to be out of fab in July, 2007.

6 References

1. K. Akarvardar, B. Blalock, S. Chen, S. Cristoloveanu, P. Gentil, M. M. Mojarradi, "Digital Circuits Using SOI Four-Gate Transistor,"

2. S. Cristoloveanu , B. Blalock, F. Allibert , B. M. Dufrene, and M. M. Mojarradi, "The Four-Gate Transistor," Proc. of the 2002 European Solid-State Device Research Conf, pp. 323-326, Firenze, Italy, September 2002.

3. B. Dufrene, K. Akarvardar, S. Cristoloveanu, B. J. Blalock, P. Gentil, E. Kolawa, and M. M. Mojarradi, "Investigation of the Four-Gate Action in G –FETs," IEEE Trans. On Electron Devices, Vol. 51, No. 11, pp. 1931-1935, Nov. 2004.

4. International Technology Roadmap for Semiconductors. http://www.itrs.net. 2006

5. Terauchi, M., "A logic-process-compatible SOI DRAM gain cell operating at 0.5 volt," IEEE International SOI Conference, 7-10 Oct. 2002 Page(s):86 – 87.

6. Zhang, R., Gupta, P., and Jha, N. "Synthesis of Majority and Minority Networks and Its Applications to QCA, TPL and SET Based Nanotechnologies," Proceedings of the 18th Conference on VLSI Design, pp 229-234, 2005.