Simulating Valleytronics Logic at Gate Level

A C++ Library and Example for Building Fully Electric Valleytronic Processors

T. Jacovich[1]

[1]Department of Physics
The George Washington University

ECE 6120: Advanced Microarchitecture Final Presentation

Valleytronics vs. CMOS

CMOS logic takes advantage of the Pauli Exclusion principle to create diodes and transistors. These allow easy voltage manipulation, but no additional constraints. Valleytronics offers a more precise view of the electron, in a manner similar to spintronics. Valleytronics has the following unique attributes.

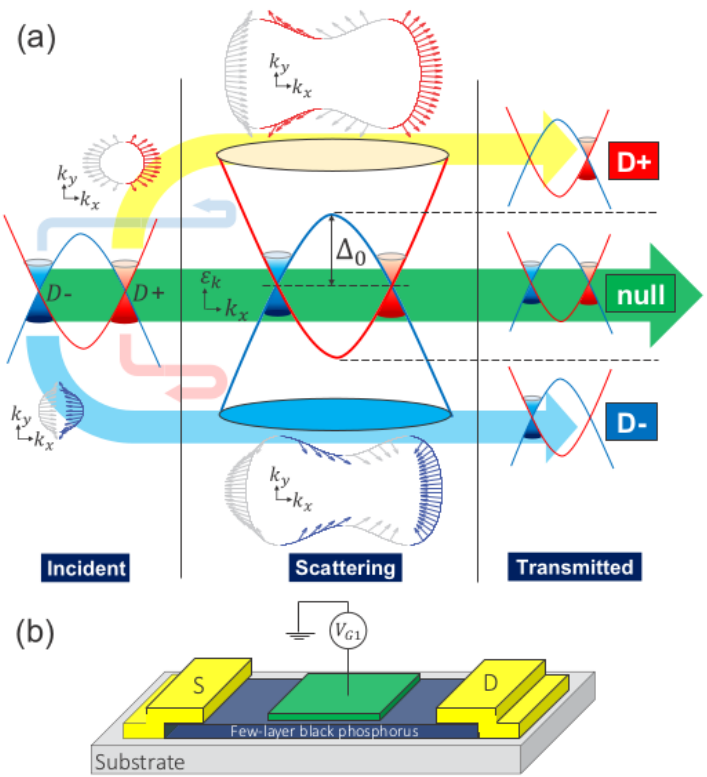
- Uses momentum states present in the metal (called valley states) to record additional information.
- Can be used in both classical and quantum processes depending on the implementation
- Incredibly new technology (Some of the required solid state metals were discovered in the last 5 years)

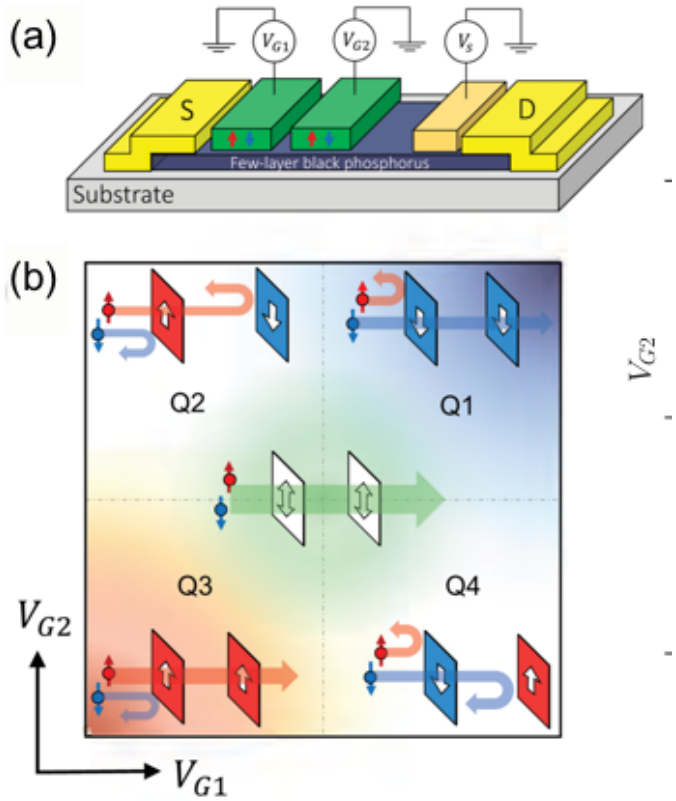
The Physics Behind All electronic gates

- Most realizations involving valleytronics have involved ultra-cold states to generate the effects.
- They have also required lasers or intense magnetic fields to create and modify
- The presented gates are based off of Ang et al. (Phys Rev. B 96, 245410 2017)
- These gates take advantage of *pseudospin-assisted valley-contrasted quantum-tunneling* (Yeah that is really what it is called.) Referred to as 2MDS in the paper.

The Physics Behind All-Electronic Gates

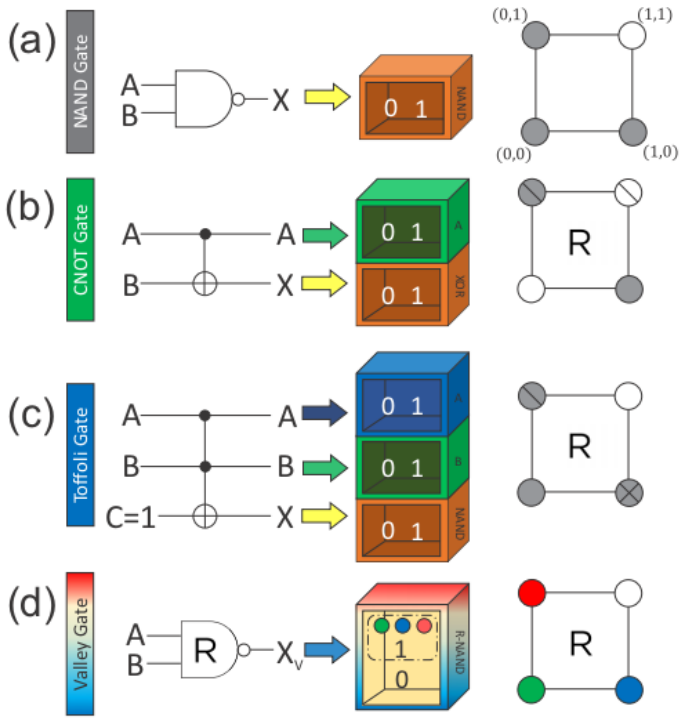
Schematics of the Gates





Reversible Logic
The Landauer Limit and Beyond

- One of the big advantages of this gate design is that it can be reversed with no additional logic.
- This makes them interesting for quantum/classical hybrid machines and for reducing heat output because of the Landauer Limit $k_B T ln\,[2]$ per lost bit.



The Plan

- The Original Plan was to take one of the readily available MIPS simulators and modify it to support valleytronic logic.
- The original plan was to implement the gates as close to physically accurate as possible.
- Because the MIPS simulator already simulated a processor, it seemed straightforward to compare valleytronic and traditional boolean logic
- But did this actually come to pass?

The Reality

- Trying to physically simulate the logic based on the Dirac Equation was Dumb (and slow, and ineffective).
- Only a few MIPS processor go down to individual gates and are available to be parsed at the level I needed, and they didn't like the 2 bits of information per single signal
- VHDL based simulators let me define some of the gates, but the valleytronic were not easily passed about.
- So I ended up starting from scratch in C++

valleytronics_logic

- A collection of libraries containing two classes `c_valley` (logic gates) `c_vlogic`(more complex circuits)
- c_valley contains most major logic gates to produces larger circuits as member objects. The rest can be constructed from these gates.
- c_vlogic contains several pieces of logic that operate reversibly.
- Unfortunately there are no CPU simulations yet because flip-flops and latches that preserve the valley polarization are not trivial.

Results and Conclusions

- All electric valleytronics could represent the next major logic technology
- As such, being able to readily simulate the logic is important for understanding where microarchitecure may be headed.
- Although there are currently some major hurdles, `valleytronics_logic` could conceivably serve as the basis for that next step.
- The big thing would be cycle logic, and that is not too far away, just further away than this presentation deadline.

References

1 Ang et al. Phys Rev B 96, 245410 (2017)

2 Isberg et al. Nature Materials 12 14/06/2013