

Can we build Classical Control Circuits for Silicon Quantum Computers?

Mark Whitney, Yatish Patel, Nemanja Isailovic, John Kubitowicz
University of California at Berkeley
{whitney, yatish, nemanja, kubitron}@cs.berkeley.edu

ABSTRACT

Many who propose quantum computing technologies focus on the quantum *datapath* without addressing the complexity of the classical *control*. We investigate the complexity of control for a specific technology, namely the Kane silicon quantum computer. We show that the pulse sequences required to effect one of the simplest operations – two-bit swap – poses a significant challenge to scalable implementation. The reason for this is two-fold: first, extremely cold operating temperatures require use of something other than CMOS for control and, second, pulse-generation for a *single bit* in the datapath requires many classical transistors. The result suggests that architects must focus on a form of SIMD for quantum datapaths, sharing pulse-generation circuits between as many quantum bits as possible.

1. INTRODUCTION

Quantum computing has become a hot topic of late, spawning work on algorithms, complexity classes, secure communication, and esoteric physical systems. In an effort to realize the many theoretical advantages of a quantum computer, there have been a large number of experimental attempts and an even larger number of proposals for building such a computer. In fact, papers on quantum architecture have begun to appear in mainstream architecture conferences [1, 7].

While much work has been done on the quantum datapath for such computers, little if any attention has been paid to the classical control. This oversight is understandable, since solid-state physicists and chemists are still trying to produce stable quantum states for computation. Nonetheless, classical control will be a vital part of any quantum architecture. Thus, we explore the difficulty of providing classical control in this paper.

For any scalable technology, classical control must be integrated on the same substrate as the quantum datapath. This somewhat obvious statement leads to two immediate difficulties. First, many proposals for quantum computing involve temperatures close to absolute zero. As a result, control circuits must operate at the same

temperature, effectively ruling out CMOS transistors¹. Since *single-electron transistors* (SETs) work optimally at these low temperatures, we explore them here. Unfortunately, SET circuits may not produce sufficient drive current to generate the high-frequency pulses needed by quantum circuits.

Second, control circuitry must somehow match the pitch of the quantum datapath to allow scalable control — a problem touched upon in [7]. We illustrate this difficulty by targeting a simple “swap wire”, an element that transports quantum information by swapping it from bit to bit. In a swap wire, the drive transistors closest to the datapath place constraints on the pitch of the quantum bits. The complexity of control makes these constraints difficult to satisfy.

To be concrete, we explore these issues in the context of the second Kane solid-state NMR architecture [9]. We give some background in Section 2. Then, since operations on qubits require precise control of electrodes located at each qubit site, Section 3 derives the timing and characteristics of control pulses required to swap values. We then assume that control will be performed by integrated single-electron transistors (SETs) and calculate the necessary timing and layout details for these transistors in Section 4.

Although many of the exact details of electrical control for the Kane model are still a bit hazy, we can make basic estimates of required drive current and circuit complexity. Although we focus on a particular quantum computing technology, many of our conclusions should apply to other solid-state quantum computing technologies faced with low operating temperatures and tight pitch constraints. Our results suggest that the pulse-generation complexity (control transistors/quantum bit) is sufficient to pose a significant challenge to achieving scalable quantum computing.

2. KANE QC ARCHITECTURE

The quantum computer proposed by Skinner, Kane, and Davenport [9] is shown in Figure 1. Its use of Si as a substrate and planar configuration are an attempt to

¹The impurity carriers become “frozen out” near absolute zero.

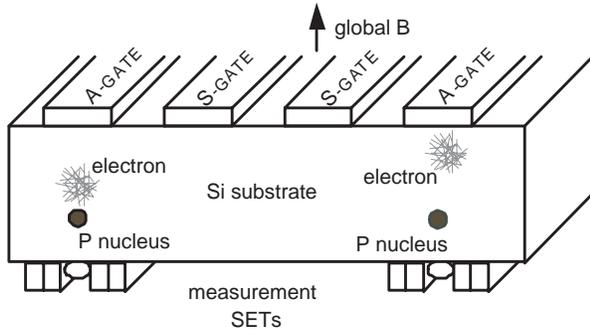


Figure 1: Basic layout of Kane solid state quantum computer. A P^+ ion and its donor electron comprise a qubit. The gates and control circuitry are above the qubits, while the measurement circuitry is below.

leverage the existing silicon processing industry for scalable fabrication. Each qubit consists of a phosphorus donor ion and its donor electron. A “zero” is represented by the spin singlet state and a “one” by the $S_z = 0$ spin triplet state.

The two basic physical gates, labeled “A” and “S” in Figure 1, control the electron/nucleus hyperfine interaction and inter-nucleus electron transport, respectively. These actions are controlled by the electric fields generated when the gate electrodes are charged. In addition to the gate electrodes, a global magnetic evolution on the electron and nucleus is affected at all times by an external field.

Since a physical qubit is made up partly of a donor ion in the Si lattice, qubit communication is nontrivial. A local swap operation is an important quantum operator for short range communication. Later in the paper, we will define the steps necessary for the swapping of two physical qubits. This swap cell is the focus of this paper.

Once the details of a single swap cell are worked out, qubit communication via swap becomes possible. A series of swap cells joined end to end form a wire, and a qubit can be moved along the length of the wire by a series of successive swaps. Figure 2 shows the process of moving 4 qubits using three swap cells. Swap operations must be staggered so that adjoining qubit pairs are not simultaneously swapped. We note that one set of alternating qubits moves left and the other set moves right, depending on which set of staggered swaps are used first.

Using this conceptual architecture as a starting point, the next section delves into the details of the electrode control necessary to implement swap wires. The initial proposal for the architecture [9] makes the simplification that the interactions controlled by the electrodes can be made to be either “on” or “off,” using a sequence of precise electrical square pulses to the appropriate electrodes. We will use this basic concept, but will supplement with a discussion of pulse-height and slew-rate.

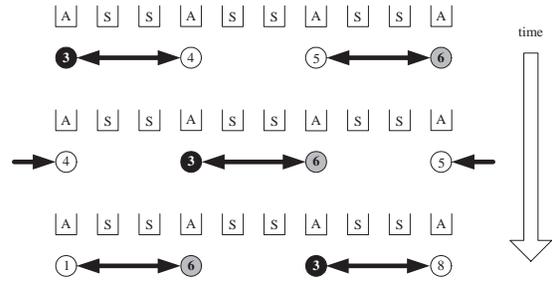


Figure 2: Multiple swap cells chained together to form a swap wire. Notice that odd qubits move right and even qubits move left.

3. SWAP WIRE SPECIFICATIONS

In this section, we derive pulse control sequences for Kane gates. We focus on one specific design goal, namely that of controlling a sequence of swap operations to produce a swap wire. Even for this restricted domain, we find that the complexity of the pulse sequences is non-trivial. After fully characterizing the control problem in this section, we will explore implementation issues in the next section. A consequence of the complexity of these pulse sequences will be that control circuitry for a qubit will dwarf the qubit itself.

3.1 Designing the Swap Datapath

Our first task will be to choose a datapath architecture for a swap wire. There are two different possibilities discussed in the literature, distinguished by placement of the control electrodes: *one-dimensional* (Figure 3) and *two-dimensional* (Figure 4). These two architectures have different requirements for control as we will see.

One-Dimensional Layout: The one-dimensional layout, proposed in [9], consists of A-gates separated by two S-gates each. Pulse sequences to swap between adjacent bits require six distinct signals to be controlled as shown in Figure 3. The sequence of steps necessary to perform a single swap operation is illustrated in Figure 5. Each time step is delineated by a dotted line with the corresponding necessary electrode voltage levels below.

The donor electron for the P ion at gate A_1 is labeled e_1 , while e_2 similarly corresponds to gate A_2 . In order to perform a swap, e_1 must execute a hyperfine interaction at gate A_2 , and e_2 must execute a hyperfine interaction at gate A_1 . The basic procedure is to pull e_1 “aside” (to gate S_1) while e_2 is moved to gate A_1 to perform its hyperfine interaction. Then e_2 is pulled “aside” (to gate S_4) while e_1 is moved to gate A_2 to perform its hyperfine interaction. Finally, both electrons are moved back to their corresponding A-gates, but the values at the P ions have been swapped.

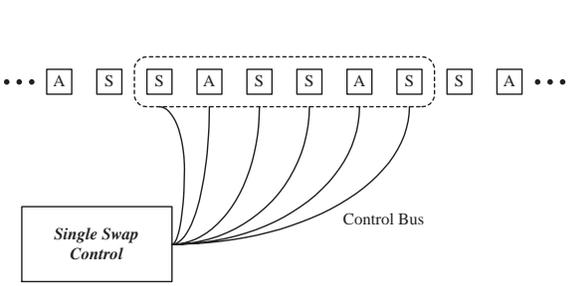


Figure 3: This is a top view of the 1-D swap cell layout replicated horizontally to form a swap wire.

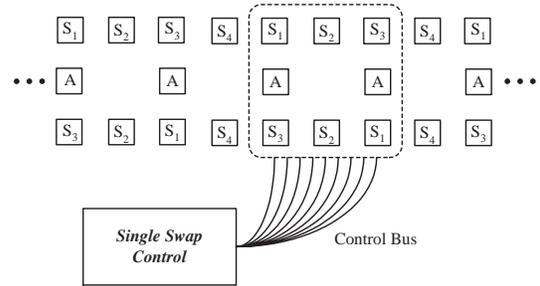


Figure 4: This is a top view of the 2-D swap cell layout replicated horizontally to form a swap wire.

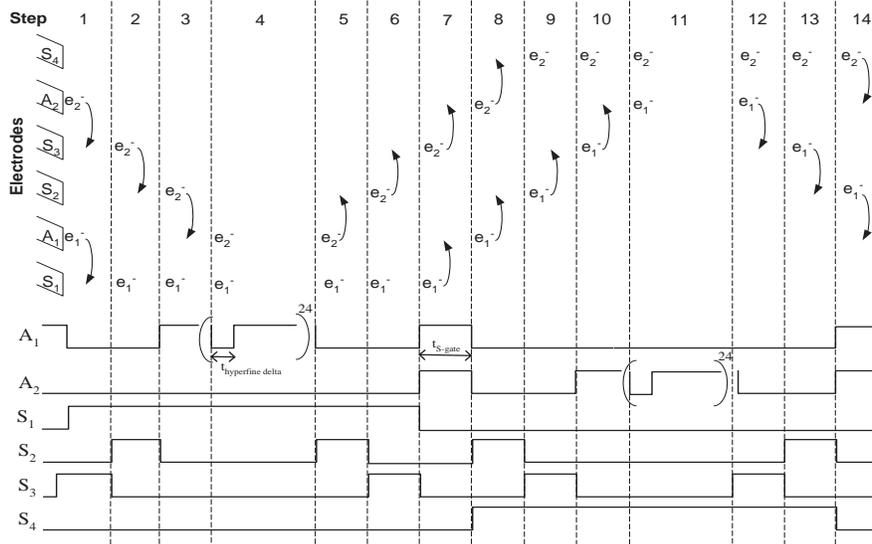


Figure 5: This is the time sequence of S- and A-gate operations necessary to swap two qubits in the 1-D layout. The parallelograms on the left represent A- and S-gates and each vertical line of electrons represents a time step in the swap. Below are the control signals necessary for each A- and S-gate to perform the electron transport above it.

The S-gate pulse sequences consist of square pulses at designated intervals. The A-gate pulse sequences consist of a few square pulses used for transport interleaved with a much more complicated pulse sequence used to apply the hyperfine interaction. We wish to obtain a “pure” hyperfine evolution, as described in [9], meaning the electron and ion interact without the influence of the global magnetic field. To do this, we must perform the hyperfine evolution in “deltas,” breaking the hyperfine period into 96 pieces. Each of the two-cycle deltas is followed by 254 cycles of pure magnetic evolution, allowing the magnetic interaction to come back to the beginning of the period when another delta is operated. There are only 24 repetitions of the on-off pulse in our sequence because the electron and nucleus need only interact for 1/4 of the period. Thus, in Figure 5, the parenthesis means that we apply two cycles low voltage followed by 254 cycles high voltage, and we repeat this 24 times.

The main problems with this design occur at steps 3 and 4 in Figure 5. At step 3 (S_1, A_1), e_1 is being “held”

by gate S_1 while e_2 is being pulled over by gate A_1 . Though it is more likely for e_2 to be attracted to gate A_1 than for e_1 , it is still possible for e_1 to escape from gate S_1 and move to gate A_1 in this step. Additionally, e_1 and e_2 are indistinguishable, so it is dangerous to have them in such close proximity to each other as this step requires. There is a risk they may spontaneously interchange.

The second and greater problem with this design occurs at step 4. At this step, e_1 should be “held” at gate S_1 while e_2 executes a hyperfine interaction at gate A_1 . In order for this to occur, gate A_1 must release e_2 . However, there is a chance of e_2 being pulled over to gate S_1 in this step instead of interacting with the P ion. Note that similar problems occur at steps 10 and 11.

Two-Dimensional Layout: In general, we would like to keep the donor electrons separated from each other as much as possible. For this reason, we consider the two-dimensional layout shown in Figure 6 (which was

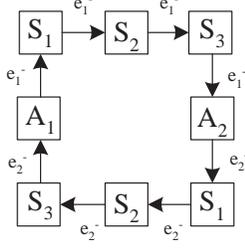


Figure 6: Schematic layout of the electrodes in the 2-D design of a single swap cell. The arrows denote the motion of the two donor electrons.

briefly suggested at the end of [3]). There are two distinct paths between each pair of A-gates. Thus, e_1 can be transported from gate A_1 to gate A_2 along the “top” path while e_2 is simultaneously being transported from gate A_2 to gate A_1 along the “bottom” path. A single swap is achieved by sequentially asserting S_1 , S_2 and S_3 to move the donor electrons to the opposite nuclei, then performing the hyperfine interaction at both A-gates, and finally returning the electrons to their initial locations.

Figure 7 shows the necessary pulse sequences to accomplish a swap in this 2-D design. As a result of design symmetry, there is a single control signal for both S_1 -gates, a single one for both S_2 -gates, etc. The system starts with the A-gates going low while the S_1 -gates go high. This causes each electron to drift to the nearest S_1 -gate. A similar pulse transports the electrons to gate S_2 and then to gate S_3 .

The A-gates then go high, and at the end of that cycle, the electrons have swapped places. At this point, the A-gate pulse sequence shown in the parenthesis of Figure 7 must be applied 24 times ($1/4$ of the full period) at each A-gate in order to execute a hyperfine interaction with each P ion. We then move the electrons back using the same S-gate sequence, and the swap is complete.

In this design, the electrons are significantly farther apart than in the one-dimensional design (260nm in 2-D vs. 130nm in 1-D). Thus, we lessen the possibility of unintentional interchange. The 2-D design also has the advantage that the two hyperfine A-gate interactions happen simultaneously. In Section 4, we show that the hyperfine interaction consumes most of the time in the swap operation, so parallel A-gate operation almost halves the time needed to swap qubits. For these reasons, we decide to abandon the 1-D design and push forward with the 2-D design for our swap cell layout.

Two-Dimensional Wire: We now consider timing for a wire based on this 2-D swap cell. The goal is to move one set of donor electrons to the “right” and the other set to the “left,” as is shown in Figure 2. The timing diagram in Figure 7 (which applies to a single swap cell) must be extended somewhat for the swap wire.

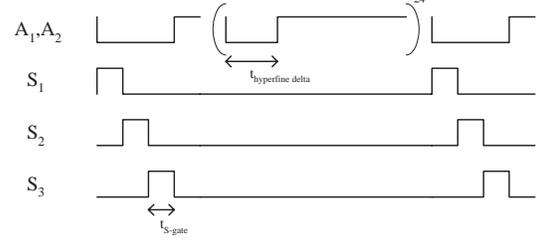


Figure 7: Control pulse sequence for 2-D layout. Note that there is a single pulse sequence for both S_1 gates. S_2 and S_3 gates each have their own signals as well.

From Figure 4, we see that there are five distinct pulse sequences that need to be generated, one each for the A-, S_1 -, S_2 -, S_3 - and S_4 -gates. (Note that there is an S_4 -gate which is not in the single swap cell in Figure 6.)

The pulse sequence for the A-gates is the same as for the basic swap. However, let us consider the S-gates in more detail. Suppose an electron begins at the leftmost A-gate in Figure 4. To reach the next A-gate to the right, we need to successively activate gate S_1 , then S_2 , then S_3 , then the A-gate again. Then to return to its original electrode, we activate gate S_1 , then S_2 , then S_3 , then the A-gate again. However, we then wish to swap in the other direction. This requires the sequence S_3 , S_4 , S_1 , A to get to the next A-gate to the left and the sequence S_3 , S_4 , S_1 , A to return. Thus, we wish to alternate between these two sequences. (Note also that we need to perform the hyperfine interaction upon arrival at the adjacent A-gate in both cases.) This sequence of operations, repeated ad infinitum, will result in alternating electrons traveling along the wire leftward and alternating electrons traveling along the wire rightward, as desired.

3.2 Pulse-Generation Parameters

From the Skinner-Kane QC proposal [9], we find that the period for the hyperfine interaction (A-gate) of a qubit is 8.5ns. The period of global magnetic interaction is 21.3ns under the suggested magnetic field strength of 1.6mT. The S-gate electron transport step details are ignored in the Skinner-Kane paper but since the interaction is not periodic, it will not affect the clock period. Later in this section and the next, we will make some quick estimates of the approximate timing of an S-gate operation, since this timing *will* still be necessary in the control logic we develop.

Clock Rate: The interaction timings determine the clock rate of the charging of the electrodes. The clock period must clearly be less than the smallest interaction period, which would be the hyperfine interaction delta. Thus, breaking up the hyperfine period into 96 pieces as mentioned earlier, we need a clock period of $8.5ns/96 = 88.5ps$ or a frequency of 11.3GHz.

The high clock rate required by the electron-nucleus hyperfine interaction introduces some interesting constraints on the exact positioning of the components in the Skinner-Kane architecture and will also impose constraints on the circuitry that drives them. We now discuss some effects of two circuit device parameters, voltage swing and drive current.

Pulse Height: The maximum electrode potential determines the time evolution of the donor electron wave function. As shown in Section 4, the transistors we have available for control produce limited voltages and currents. A lower potential will lead to a slower hyperfine turn off time because the force exerted on the electron by the electric field will be less. This effect requires closer study because the clock cycle of this quantum computer depends on the speed at which electrons can be manipulated in A-gate operations. In [9], it was assumed that the hyperfine interaction could go from “off” instantaneously to “on.” This assumption may not be valid with such short interaction times and low electrode voltages. Since the focus of this paper is on control circuits and not refinement of the physical model, we will continue on the assumption that the A-gates “just work,” but this issue clearly merits future work.

Error rates in A-gate operations are partially dependent on the differential between the potential at the P ion directly under the electrode and the potential at a P ion adjacent from the same electrode. The question is essentially whether the qubit directly under the electrode will evolve significantly faster than the adjacent qubit. This error constraint will surely require greater qubit separation so the neighboring qubit observes a further decayed potential.

The results in [8] imply that qubit nuclei separation of 20nm would not work with such a low voltage. In fact, there is an interesting tradeoff in errors here. The closer the qubits are spaced, the larger the probability of error from an A-gate affecting an adjacent qubit site. The further away the qubits are spaced, the larger the probability of error from decoherence of the fragile electron spin state as it is transported from one qubit site to another. From [4], we see that spin coherent electron transport is possible at up to $100\mu m$. In our analysis, we choose a somewhat arbitrary electrode spacing of 100nm. We chose this number because electrostatic simulations show that the electron directly under the gate feels an electric field about 2 orders of magnitude stronger than the electron at an adjacent site.

Slew Rate: Another concern is whether the gate electrodes above the donor sites can be charged in the necessary 88.5ps. Electrode charge time depends on both the voltage swing and maximum current available.

If we look at the A-gate in Figure 8, the electrode above the donor site is 30nm in length on a side and 130nm between qubit sites (electrodes are 100nm apart

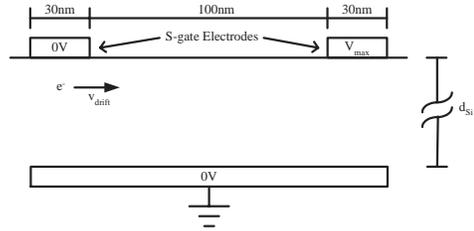


Figure 8: Setting gate capacitance: Constraints on drive current and maximum slew-rate set a maximum gate capacitance. One option for minimizing this capacitance is to adjust the height relative to the back gate (see text).

and qubits are centered underneath). If we model the electrode with a back gate as a parallel plate capacitor, we find the electrode’s capacitance to be $C_{electrode} \approx \epsilon_0 * \kappa_{Si} * 900nm^2cm/d_{Si}$, where d_{Si} is the thickness of the Si between the electrode and the metallic back gate. We can calculate the minimum d_{Si} given the constraint that the capacitor must be small enough to be charged in the above mentioned time, with the given maximum current. If we approximate the electrode/capacitor as charging at a constant rate, the approximate minimum spacing needed is:

$$d_{Si} = \frac{V_{electrode} * \epsilon_0 * \kappa_{Si} * 900nm^2}{\tau_{hyperfine} * i_{max}}$$

From [8], we see that the electrode to back gate distance (d_{Si}) affects the probability of erroneous A-gate operation. The closer the back gate is to the electrode, the larger the voltage differential between the active qubit site and the adjacent site, thus a smaller probability of error. This back gate distance would ideally feed into an error analysis which is a function of qubit separation and d_{Si} . This calculation is beyond the scope of this paper but it will be important future work to produce a physical model of the full A-gate and S-gate operation and determine the total error rates and more precise timings of these elementary operations.

Section 4.2 will take the geometry formula above and the parameters of our chosen circuit technology to derive exact parameters of the swap cell layout with the necessary control. First, we design the logic for the swap cell controlling state machine.

4. DRIVE AND CONTROL CIRCUITRY

Now that we’ve settled on the two-dimensional swap layout (Figure 4) and determined the pulse sequences that we need to implement, we can design the control circuitry necessary to produce these pulses. The extremely low temperatures (1 Kelvin) required for proper operation of a solid state QC architecture render standard CMOS transistors useless. Fortunately, a good deal of research has been conducted in modeling and building single-electron transistors (SETs) [10, 5, 13], which ac-

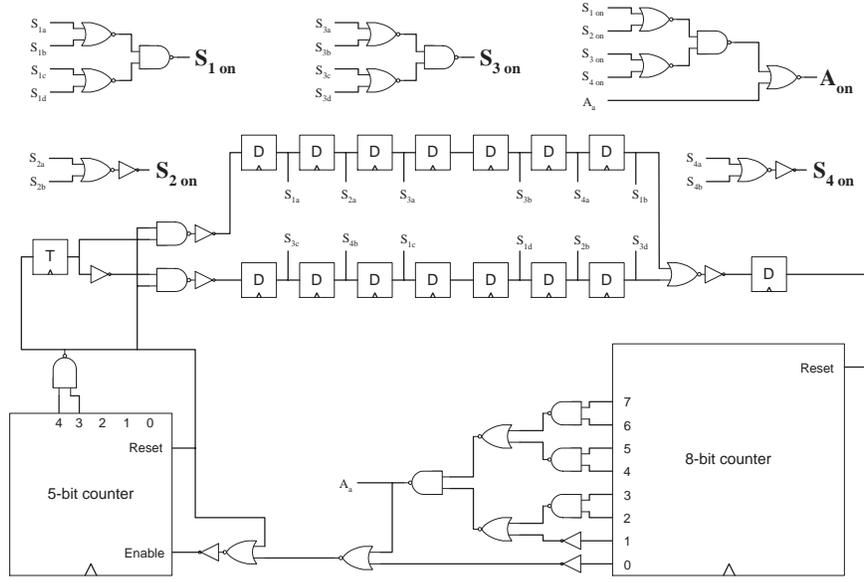


Figure 9: This control mechanism for a swap wire produces the pulse sequences previously discussed. The outputs produced are four signals for the four types of S-gates and one signal for the A-gates.

tually have more favorable characteristics at lower temperatures. This makes SETs an ideal choice for implementing the drive circuitry and control logic in our architecture. We will now describe an SET-based control circuit for our swap wire.

4.1 Control Circuit Design

We shall design a single global control mechanism which produces the five distinct pulse sequences that need to be generated, one each for the S_1 -, S_2 -, S_3 -, S_4 - and A -gates discussed in Section 3. The signals are then distributed to many A- and S-gates.

Figure 9 shows the control mechanism. The bulk of the control is used to generate the complicated A-gate pulse sequence. The goal for the hyperfine interaction is to turn the A-gate off for 2 cycles out of every 256, and to do this 24 times. The 8-bit counter is used to count up to 256 and turn the A-gate off for 2 of those cycles. The 5-bit counter is used to count up to 24 to determine the end of the hyperfine interaction.

The S-gate control is implemented as a series of D flip-flops. There is one series of flip-flops for each of the two sequences we wish to activate. The T flip-flop is toggled to switch between these two sequences. The D flip-flops are used to keep the appropriate signal high for a cycle. At the end of this section, we will show that one cycle is sufficient time to complete the electron transport. When either sequence is completed, we set the A-gate high and begin the hyperfine interaction sequence again.

In order to determine the feasibility of such a design, we shall approximate its area using modern technology constraints. SET islands of size 30nm x 30nm have been

successfully fabricated [5], so we use this as our minimum feature size. We also assume that the electrode size is the minimum feature size of 30nm to a side.

Figure 10 shows the layout of an inverter, a NAND gate and an SR-latch using SETs. The squares labeled “T” represent 30nm x 30nm islands. The thicker black lines represent tunnel junctions at either end of each island. The grey areas are physically located above the rest of the layout. They act as the gate of each SET by creating capacitance to an island. (Note that the grey areas are drawn thinner than 30nm for clarity, but they are actually just as thick as the other areas.) The dark black squares are connectors between the lower layout and the grey areas above. In the inverter and the NAND gate, the grey areas are the inputs to the gate.

Using the fact that each unit square in Figure 10 is 30nm x 30nm, we can calculate approximate areas for these three basic gates. The inverter is 60nm x 150nm, the NAND gate is 120nm x 210nm (the same as a NOR gate), and the SR-latch is 210nm x 210nm. Additionally, an XOR gate in SET technology only uses two SETs, so it uses approximately the same amount of room as an inverter. Now we can attempt to approximate the total area for the control mechanism.

D flip-flops and T flip-flops each consist of 2 SR-latches, 4 NAND gates and an inverter. The control mechanism contains 16 flip-flops, 24 NAND or NOR gates and 9 inverters, not including the two counters. An n-bit counter requires n T flip-flops for the counting mechanism. However, an additional 3 NAND gates, 1 inverter and 1 XOR gate are needed per bit if we wish to be able to RESET. So the 8-bit and 5-bit counters to-

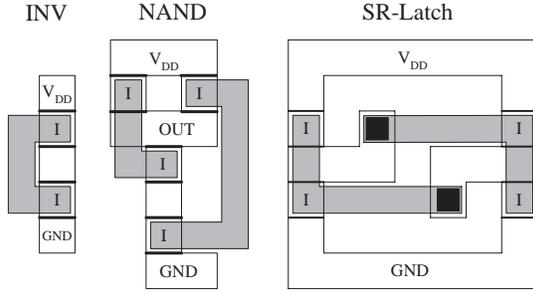


Figure 10: SET layouts for three basic components: an inverter, a NAND gate, and an SR-latch. Squares labelled “I” are the SET islands. Grey areas are above the SETs, connected to the lower level at the black junctions.

gether use 13 T flip-flops, 39 NAND gates, 13 inverters and 13 XOR gates.

This gives us a grand total of 58 SR latches, 179 NAND or NOR gates, 51 inverters and 13 XOR gates. This means that the gates take up a total area of $\sim 7.6\mu\text{m}^2$. Without doing a full optimized layout, we can’t know the actual area of the mechanism. So for the purposes of discussion, we will assume that the space for routing between the gates is approximately 50% of the area of the gates, giving us a total area of $\sim 11.5\mu\text{m}^2$ for the entire control mechanism.

4.2 Drive Circuit Calculations

The A-gate and S-gate electrodes need to be charged and discharged according to the delicate timing requirements mentioned in Section 3.2. In our design, we use SET buffers to charge and discharge the electrodes. A schematic of the gate electrode with charging and discharging SETs is shown in Figure 11. We will now use some parameters from the literature on SET physical implementation to derive some of the constraints outlined earlier.

The introduction of SET buffers as electrode drive circuits leads to a few problems, the primary ones being voltage swing and drive current. All physically implemented SETs so far are capable of handling voltage swings of up to only 40mV [12]. This is due to the fact that the gating operation of SETs relies on Coulomb blockade in the quantum dot island of the SET [11]. Source-drain voltages above 40mV essentially overpower the effect of Coulomb blockade to prevent electrons from moving through the quantum dot. This device constraint means that the A- and S-gate electrodes can only be charged to at most 40mV. This relatively low voltage constraint effects the operating speed of the electron wave function deforming operations, as well as the quantum operation error rates as mentioned in 3.2.

Since *single-electron* transistors work on the order of single-electron flow from source to drain, they certainly do not have high drive current either. In fact, in current

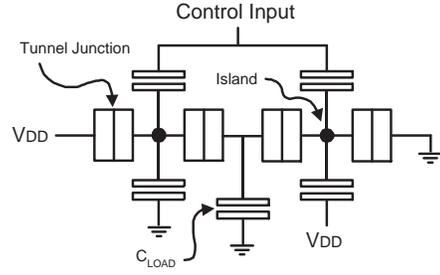


Figure 11: Driving a quantum control gate with SETs. This drive inverter should be placed close to the quantum gates. Control from the pulse-generator enters at the top. C_{LOAD} is the Kane control gate.

physical specimens, the maximum current observed is around 5nA [5]. Thus, our a concern is how our electrode must be situated so it can be charged on the order of our clock period, 88.5ps.

Thus, we have the following parameters: $V_{electrode} = 0.04V$, $\tau_{hyperfine} = 88.5ps$, and $i_{max} = 5nA$. If we plug these into the formula derived in Section 3.2, we obtain $d_{si} = 0.82\mu\text{m}$.

With the electrode distances, voltages, and currents established, we are also in a position to calculate S-gate timing. Placing d_{si} into Figure 8 and using an electrostatic simulator, we calculate the electric field to be approximately 20V/cm at the target electron. We then use the experimental data in [2] to compute the electron drift velocity for that field strength. Using this we find the approximate time for an electron to move from one electrode to the next (a distance of 130nm) to be $t_{shuttling} = 6ps$. This justifies our assumption that the electron transport time is significantly less than the hyperfine interaction time.

So as a summary of timing, the expected time for an electron to move between two adjacent electrodes is $\sim 6ps$. Our clock period was 88.5ps, so we can expect that each electron will easily reach the next gate within one clock cycle. For this reason, to perform electron transport, the S_1 gates need only remain high for one clock cycle. This is likewise true for the S_2 and S_3 gates. As mentioned before, this provides further support for the 2-D swap cell layout because the time spent on the hyperfine interaction dominates. The 1-D layout requires serial interaction while the 2-D layout can interact in parallel.

5. DISCUSSION

From Section 4, we see that regardless of the specific layout of SET circuits, the qubit control circuitry for a swap wire is extensive. An issue we must therefore consider is the placement of this control. One method of integrating the swap cells and control circuitry is to place all the required control in a layer above or in the plane

alongside the swap cells. The control circuitry consumes too much space to allow for a unique control block dedicated to each swap cell. This results in the need for multiple swap cells to share a single control block. The sharing of a control block is entirely possible, however we must factor in the area requirements of such a design.

As described in Section 4, the area of the control circuitry is estimated to be about $11.5\mu\text{m}^2$. If we assume the logic can be laid out with the same width as the swap cells, 250nm, the minimum wire length allowed will be $46\mu\text{m}$ in order to fit all the necessary control circuitry, naively assuming that we can lay out this circuit to have such a narrow width. If a swap cell cluster is to be used as our wire to connect quantum logic components which cannot be adjoined without routing, we will be forced to design very sparse quantum circuits.

Another issue we must consider is the ability to manufacture the proposed design. Many people have successfully fabricated single-electron device circuits including inverters, NAND gates, NOR gates, and XOR gates [10, 5, 12]. The fairly complicated fabrication techniques required to make the small islands and tunnel junctions limit the number and orientation of SETs that can be created. Unfortunately, the circuits we have proposed require large SET circuits with very small feature sizes which up to this date have not been experimentally demonstrated. However, research is currently being conducted to turn SET circuits into reality [6].

The design complexity we have uncovered in this paper suggests that quantum computer architects will have to consider reusing pulse-generation circuits. This suggests a SIMD model of qubit control, where many qubits receive signals simultaneously. In the case of the swap wire, this is straightforward: qubits are transported identically between sets of functional units along sections of swap wire connected to the same control block. Taking this design a step further, the swap wires could be analogous to a standard microprocessor and its clock distribution network. If we place a global instance of the control circuitry on the chip, we can distribute the control signals to all the swap wires. This scheme, coupled with a few SETs above the swap cells to enable and disable the control signals, will allow wires of arbitrary length.

Of course, with such low drive currents and high clock rates, the issues of long distance clock distribution and high fanout will need careful study. These control signals will need substantial buffering, which will consume more area than our current estimates. In short, the classical control associated with quantum computing provides a significant design challenge. We have only scratched the surface with this preliminary study.

6. CONCLUSION

In this paper, we explored some of the issues present in integrating classical control with a quantum datapath. For concreteness, we picked a specific technol-

ogy (Kane quantum computing) and specific datapath functionality (swap wire). However, our results are generally applicable: they highlight the fact that each quantum bit requires complex pulse generation to operate correctly. This, in turn, creates fabrication problems, since quantum bits must be close enough together to interact without decoherence. An additional problem is the extremely low operating temperatures which force use of low-drive control elements such as single-electron transistors. Our results sound a general cautionary note: the pulse-generation complexity (control transistors/quantum bit) is sufficiently complex to pose a barrier to achieving large-scale quantum computing. This suggests that architects must attempt to reuse control circuitry as much as possible, perhaps adopting a form of SIMD design for quantum datapaths.

7. REFERENCES

- [1] D. Copley, M. Oskin, T. Metodiev, F. T. Chong, and I. Chuang. The Effect of Communication Costs in Solid-State Quantum Computing Architectures. In *Symp. on Parallel Alg. and Arch (SPAA)*, 2003.
- [2] C. Jacobini, C. Canali, G. Ottaviani, and A. Quaranta. A Review of Some Charge Transport Properties of Silicon. *Solid-State Electronics*, 20:77–89, 1977.
- [3] B. E. Kane. Silicon-based Quantum Computation. <http://xxx.lanl.gov/quant-ph/0003031>, 2000.
- [4] J. M. Kikkawa and D. D. Awschalom. Lateral Drag of Spin Coherence in Gallium Arsenide. *Nature*, 397:139–141, 1999.
- [5] D. H. Kim, S. Sung, K. R. Kim, B. H. Choi, S. W. Hwang, D. Ahn, J. D. Lee, and B. Park. Si Single-Electron Transistors with Sidewall Depletion Gates and their Application to Dynamic Single-Electron Transfer Logic. In *IEEE Int'l Electron Devices Mtg.*, 2001.
- [6] Y. Ono, Y. Takahashi, K. Yamazaki, M. Nagase, H. Namatsu, K. Kurihara, and K. Murase. Fabrication Method for IC-Oriented Si Single-Electron Transistors. *IEEE Trans. on Electron Devices*, 47:147–153, 2000.
- [7] M. Oskin, F. T. Chong, I. Chuang, and J. Kubiatiowicz. Building Quantum Wires: The Long and Short of it. In *Int'l Symp. on Comp. Arch. (ISCA)*, 2003.
- [8] C. Pakes, C. Wellard, D. Jamieson, L. Hollenberg, S. Praver, A. Dzurak, A. Hamilton, and R. Clark. Modeling of Electrostatic Gate Operations in the Kane Solid State Quantum Computer. *Proc. of SPIE*, 4591:70–77, 2001.
- [9] A. J. Skinner, M. E. Davenport, and B. E. Kane. Hydrogenic Spin Quantum Computing in Silicon: a Digital Approach. <http://xxx.lanl.gov/quant-ph/0206159>, 2003.
- [10] Y. Takahashi, A. Fujiwara, Y. Ono, and K. Murase. Silicon Single-Electron Devices and Their Applications. In *IEEE Int'l Symp on Multiple-Valued Logic*, 2000.
- [11] Y. Takahashi, Y. Ono, A. Fujiwara, and H. Inokawa. Silicon Single-Electron Devices. *Journal of Physics: Condensed Matter*, 14:R995–R1033, 2002.
- [12] K. Uchida, J. Koga, R. Ohba, and A. Toriumi. Programmable Single-Electron Transistor Logic for Low-Power Intelligent Si LSI. In *IEEE Int'l Solid-State Circuits Conf.*, 2002.
- [13] C. Wasshuber. Single-Electronics - How It Works. How It's Used. How It's Simulated. In *IEEE Int'l Symp. on Quality Electronic Design*, 2002.