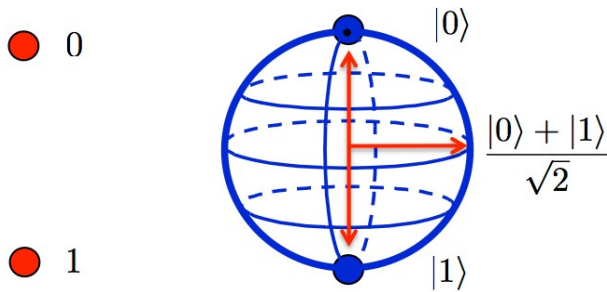


# Modeling of silicon two qubit gates

Yann-Michel Niquet, INAC/MEM/L\_Sim  
([yniquet@cea.fr](mailto:yniquet@cea.fr))



**Classical Bit**

**Qubit**

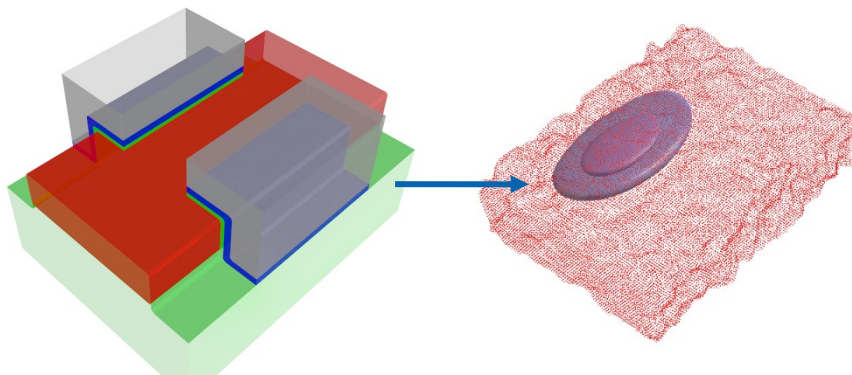
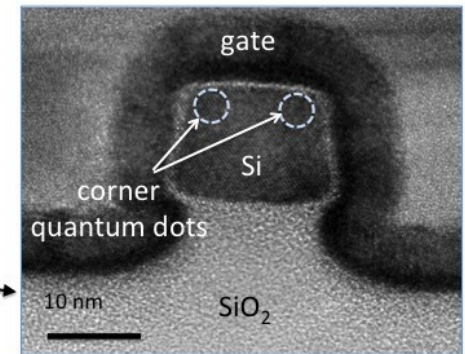
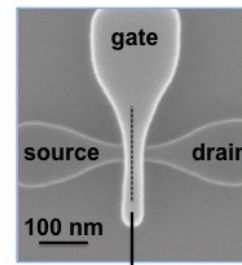
**Quantum bit:** Store information as a **coherent superposition** of «  $|0\rangle$  » and «  $|1\rangle$  » states

**Benefits:** Exponential speed-up with respect to classical information for some problems

**Possible implementation:**  $|0\rangle \equiv |\downarrow\rangle$ ,  $|1\rangle \equiv |\uparrow\rangle$  states of an electron in a « **single electron transistor** »

The spin is manipulated with RF pulses on the gates

**Quantum computation:** Entanglement by exchange (Coulomb+tunnel) interactions between qubits



**Strong need for modeling:**  
Quantum confinement and Coulomb correlations, manipulation based on spin-orbit coupling, decoherence....

# Objective: Modeling of the exchange coupling between two qubits and time-dependent simulation of a two qubit gate.

**Framework:** Electronic structure calculations on realistic geometries (HPC).  
Interactive Python scripts for time-dependent simulations.  
Analytical, effective models for interacting qubits.

```
z = TDSE.matrix(Ci.load_observable("V1", Z))
Su = bdir[0]*Sx+bdir[1]*Sy+bdir[2]*Sz

Loading signals and observables...

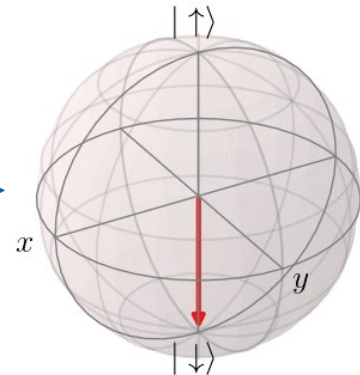
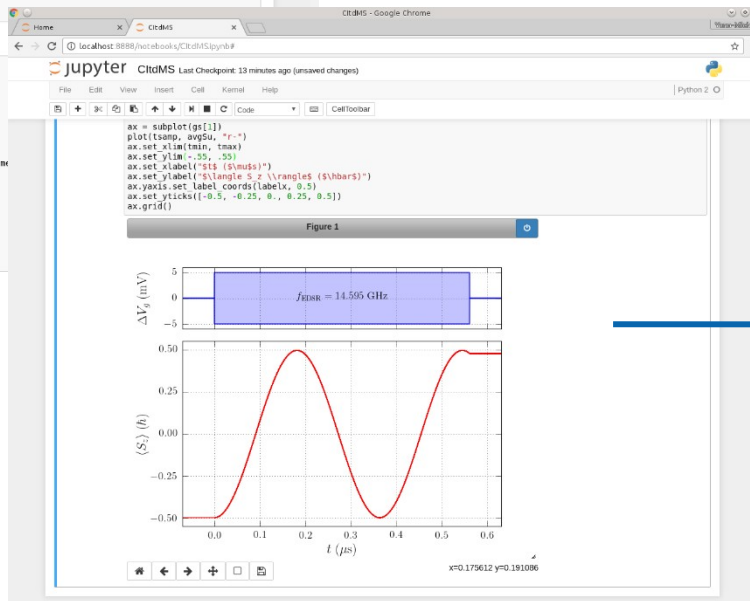
In [6]: # Set-up clock and signals.
f0 = (eci[up]-eci[down])/(2.*pi)
print "Clock frequency = %e Hz." % (f0/uat)
clock = Clock(f0)
t, nt, dt = clock.tgrid(-delay, tpulse+delay, npp)
p = Vpulseclock.pulse(t, 0., tpulse)
s = clock.clock(t)*p

Clock frequency = 1.459450e+10 Hz.

In [7]: # Set-up initial state and compute evolution.
psi = TDSE.ket(dowm: 1.)
print ""
print "Computing time evolution (%s)..." % method
print "Norm of the initial state = %f." % norm(psi)
tsamp = copy(t[::npp:8])
psi, tsamp, psisamp = TDSE.evolution(t, s, psi, tsamp, method = method)
print "Norm of the final state = %f." % norm(psi)
print "Execution time = %f s." % TDSE.texec

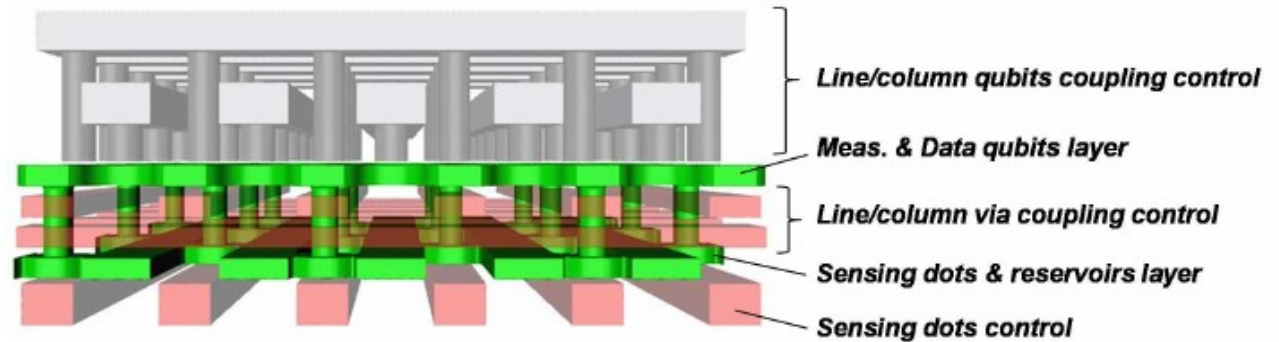
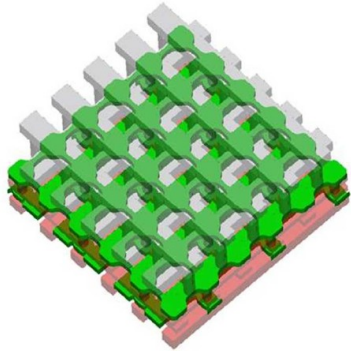
avgSu = expectation_values(Su, psisamp)
avgX = expectation_values(X, psisamp)
avgY = expectation_values(Y, psisamp)
avgZ = expectation_values(Z, psisamp)

Computing time evolution (HEVSIEV) ...
Norm of the initial state = 1.000000
Norm of the final state = 1.000000
Execution time = 1.000000
```



**WE WANT YOU!**  
yniquet@cea.fr

**Context:** ERC Synergy “quCube” (2019-2024) gathering CEA/LETI, CEA/INAC and CNRS/Néel around the realization of 2D arrays of qubits (budget: 14 M€).



Strong connections already existing with experimental teams at CEA and in Europe (6 joint publications in 2018).

