

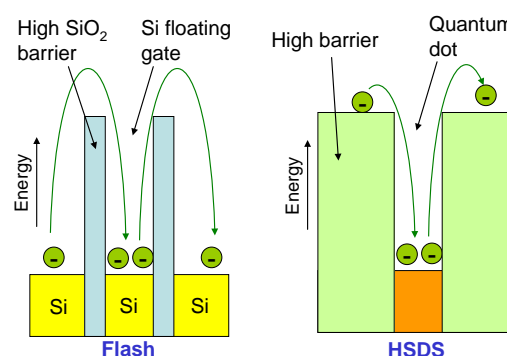
## Quantum Dot Memories

**Moore's law** 'predicts' that the number of transistors on a chip will double every two years, and has governed the microelectronics industry for more than 40 years. Today's state-of-the-art IC's have Si/SiO<sub>2</sub>-based transistors with minimum feature sizes below 50 nm – bringing them well into the regime of nanotechnology. However, this clearly cannot continue forever. Indeed, the limitations of standard CMOS technology have already resulted in introduction of high-performance materials: *e.g.* the insulating SiO<sub>2</sub> between the gate and the channel is being superseded by sophisticated high- $\kappa$  dielectrics. The latter was called 'The biggest change in CMOS technology for 40 years' by Gordon Moore, yet it will only allow a couple more generations of scaling. Nowhere is this problem more acute than in memories, where conventional scaling is expected to end by 2014 – just around the corner in terms of product development timescales. The race is therefore on to either inject some new materials into the Si process, or to go to for a completely new memory concept, with inherent performance advantages.

**Flash memory**, used in USB drives, digital cameras, mobile phones *etc.* is a non-volatile memory based on charge-storage in an electrically-isolated 'floating gate' placed between the conductive channel used for readout and the 'control gate'. Flash has been the memory technology driver since 2003, taking the leading position from dynamic random-access memory (DRAM), which is the capacitively-based memory used in PCs *etc.* These two memory types account for more than 95% of the \$60bn annual memory market. However, despite its fancy name, the performance of Flash is rather mediocre. Charging up the floating gate requires pushing charge across the SiO<sub>2</sub> barriers that isolate it (see figure), making writing very slow and, eventually, damaging it.

**Quantum dot (QD) memory** (see figure) is an entirely new memory concept which avoids the intrinsic problems associated with pushing charge through an insulating barrier. It works by storing the charge in a potential well created by surrounding a small volume of a narrow band-gap compound semiconductor (*e.g.* GaSb), with a wide band-gap compound semiconductor (*e.g.* GaAs). In such a device charge capture (write operation) is extremely fast – about 1000 times faster than for Flash. Clearly, removing the charge (erase) is still slow, but this can be engineered out in data-storage memory using appropriate architecture, so is not an issue. However, this is not good enough for RAM used to execute code in a computer: read, write and erase operations all need to be fast. Here, a second major advantage of QD memories comes into play. Rather than being stuck with a choice of Si and SiO<sub>2</sub> which always gives the same barrier height, there are a huge variety of compound semiconductors which can be combined in different configurations to allow tuning of the barrier height to give the right balance of erase speed and charge storage time. One can choose to have a very fast device (like DRAM) which needs to be 'refreshed', or a non-volatile device (like Flash) with slow erase. We believe that a QD memory RAM which is 10× faster than DRAM, and has 10× longer refresh (uses 10× less power) is possible.

QD memory research in Lancaster is focused on the growth and characterisation of the group III-V compound semiconductor nanostructures from which the devices are processed, and is supported by the Royal Society and by EPSRC in the framework of the [QD2D NanoSci E+](#) project with [U. Duisberg-Essen](#), [TU Berlin](#) and [TU Eindhoven](#), and in collaboration with [QinetiQ](#).



*Schematics of memory concepts: (Left) Si/SiO<sub>2</sub>-based Flash memory. Large SiO<sub>2</sub> barriers make writing and erasing the device slow. The control gate and readout channel are not shown. (Right) Quantum dot memory, labelled HSDDS (high-speed data storage), is based on III-V compound semiconductors, and has no barrier for writing, making it very fast. Erase is still slow, but this can be engineered out.*