LETTER

doi:10.1038/nature12502

# Carbon nanotube computer

Max M. Shulaker[1], Gage Hills[2], Nishant Patil[3], Hai Wei[4], Hong-Yu Chen[5], H.-S. Philip Wong[6] & Subhasish Mitra[7]

**The miniaturization of electronic devices has been the principal driving force behind the semiconductor industry, and has brought about major improvements in computational power and energy efficiency. Although advances with silicon-based electronics continue to be made, alternative technologies are being explored. Digital circuits based on transistors fabricated from carbon nanotubes (CNTs) have the potential to outperform silicon by improving the energy–delay product, a metric of energy efficiency, by more than an order of magnitude. Hence, CNTs are an exciting complement to existing semiconductor technologies[1,2]. Owing to substantial fundamental imperfections inherent in CNTs, however, only very basic circuit blocks have been demonstrated. Here we show how these imperfections can be overcome, and demonstrate the first computer built entirely using CNT-based transistors. The CNT computer runs an operating system that is capable of multitasking: as a demonstration, we perform counting and integer-sorting simultaneously. In addition, we implement 20 different instructions from the commercial MIPS instruction set to demonstrate the generality of our CNT computer. This experimental demonstration is the most complex carbon-based electronic system yet realized. It is a considerable advance because CNTs are prominent among a variety of emerging technologies that are being considered for the next generation of highly energy-efficient electronic systems[3,4].**

CNTs are hollow, cylindrical nanostructures composed of a single sheet of carbon atoms, and have exceptional electrical, physical and thermal properties[5–7]. They can be used to fabricate CNT field-effect transistors (CNFETs), which are promising candidate building blocks for the next generation of highly energy-efficient electronics[1,2,8]: CNFET-based digital systems are predicted to be able to outperform silicon-based complementary metal–oxide–semiconductor (CMOS) technologies by more than an order of magnitude in terms of energy–delay product, a measure of energy efficiency[2–4].

Since the initial discovery of CNTs, there have been several major milestones for CNT technologies[9]: CNFETs, basic circuit elements (logic gates), a five-stage ring oscillator fabricated along a single CNT, a percolation-transport-based decoder, stand-alone circuit elements such as half-adder sum generators and D-latches, and a capacitive sensor interface circuit[10–16]. Yet there remains a serious gap between these circuit demonstrations for this emerging technology and the first computers built using silicon transistors, such as the Intel 4004 and the VAX-11 (1970s). These silicon-based computers were fundamentally different from the above-mentioned CNFET-based circuits in several key ways: they ran stored programs, they were programmable (meaning that they could execute a variety of computational tasks through proper sequencing of instructions without modifying the underlying hardware[17]) and they implemented synchronous digital systems incorporating combinational logic circuits interfaced with sequential elements such as latches and flip-flops[18].

It is well known that substantial imperfections inherent in CNT technology are the main obstacles to the demonstration of robust and complex CNFET circuits[19]. These include mis-positioned and metallic CNTs. Mis-positioned CNTs create stray conducting paths leading to incorrect logic functionality, whereas metallic CNTs have little or no bandgap, resulting in high leakage currents and incorrect logic functionality[20]. The imperfection-immune design methodology, which combines circuit design techniques with CNT processing solutions, overcomes these problems[20,21]. It enables us to demonstrate, for the first time, a complete CNT computer, realized entirely using CNFETs. Similar to the first silicon-based computers, our CNT computer, which is a synchronous digital system built entirely from CNFETs, runs stored programs and is programmable. Our CNT computer runs a basic operating system that performs multitasking, meaning that it can execute multiple programs concurrently (in an interleaved fashion). We demonstrate our CNT computer by concurrently executing a counting program and an integer-sorting program (coordinated by a basic multitasking operating system), and also by executing 20 different instructions from the commercial MIPS instruction set[22].

The CNT computer is a one-instruction-set computer, implementing the SUBNEG (subtract and branch if negative) instruction, inspired by early work in ref. 23. We implement the SUBNEG instruction because it is Turing complete and thus can be used to re-encode and perform any arbitrary instruction from any instruction-set architecture, albeit at the expense of execution time and memory space[24,25]. The SUBNEG instruction is composed of three operands: two data addresses and a third partial next instruction address (the CNT computer itself completes the next instruction address, allowing for branching to different instruction addresses). The SUBNEG instruction subtracts the value of the data stored in the first data address from the value of the data stored in the second data address, and writes the result at the location of the second data address.

The next instruction address is calculated to be one of two possible branch locations, depending on whether the result of the subtraction is negative. The partial next instruction address given by the present SUBNEG instruction omits the least significant bit. The least significant bit is calculated by the CNT computer, on the basis of whether the result of the SUBNEG subtraction was negative. This bit, concatenated with the partial next instruction address given in the SUBNEG instruction, makes up the entire next instruction address. A diagram showing the SUBNEG implementation is shown in Fig. 1a.

As our operating system, we implement non-pre-emptive multitasking, whereby each program performs a self-interrupt and voluntarily gives control to another task[26]. To perform this context switch, the instruction memory is structured in blocks, and each block contains a different program. To perform the self-interrupt, the running program stores a next instruction address belonging to a different program block; thus, the other program begins execution at this time. During the context switch, the CNT computer updates a process ID bit in memory, which indicates the program running at present. An example of the operating system running two different programs concurrently is shown in Fig. 1b.

The circuitry of the CNT computer is entirely composed of CNFETs, and the instruction and data memories are implemented off-chip, following the von Neumann architecture and the convention of most computers today. The off-chip memories perform no operation other

[1]Stanford University, Gates Building, Room 331, 353 Serra Mall, Stanford, California 94305, USA. [2]Stanford University, Gates Building, Room 358, 353 Serra Mall, Stanford, California 94305, USA. [3]SK Hynix Memory Solutions, 3103 North First Street, San Jose, California 95134, USA. [4]Stanford University, Gates Building, Room 239, 353 Serra Mall, Stanford, California 94305, USA. [5]Stanford University, Paul G. Allen Building, Room B113X, 420 Via Ortega, Stanford, California 94305, USA. [6]Stanford University, Paul G. Allen Building, Room 312X, 420 Via Ortega, Stanford, California 94305, USA. [7]Stanford University, Gates Building, Room 334, 353 Serra Mall, Stanford, California 94305, USA.

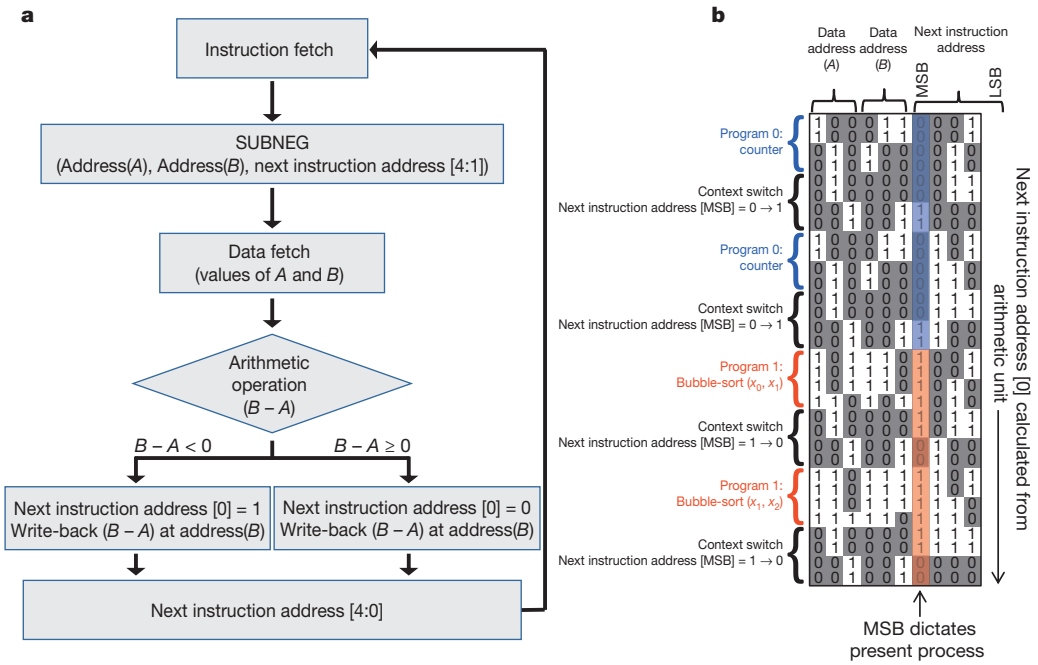



**Figure 1 | SUBNEG and program implementation.** **a**, Flowchart showing the implementation of the SUBNEG instruction. **b**, Sample program on CNT computer. Each row of the chart is a full SUBNEG instruction. It is composed of two data addresses and a partial next instruction address. The (omitted) least significant bit (LSB) of the next instruction address is calculated by the arithmetic unit of the CNT computer, and the most significant bit (MSB) of the next instruction address indicates the running program, either a counter or bubble-sort algorithm in this instance.

than performing a single read or a single write in a clock cycle. The address, data (for write), and read and write enable signals are provided by the CNT computer; the values, once read, are stored in D-latches in the CNT computer, built entirely using CNFETs. A full schematic of the CNT computer is shown in Fig. 2a. The CNT computer performs four tasks.

(1) Instruction fetch: this task supplies instruction memory with the address to read. On the first clock (Clock1), the SUBNEG instruction is read from the instruction memory and saved in a bank of ten D-latches. The SUBNEG instruction contains the partial next instruction address (as explained above), and the addresses of the two single-bit data values to operate on (represented as [*A*] and [*B*], both of which comprise three bits).

(2) Data fetch: this task supplies the data memory with the addresses given by the SUBNEG instruction to read. On Clock1, the first data address ([*A*]) is read and the value is saved in a D-latch. On the second clock (Clock2), the second data address ([*B*]) is read and the value is saved in another D-latch.

(3) Arithmetic operation: this task performs the computation (subtraction and comparison with zero) on the two data values supplied by the data-fetch unit.

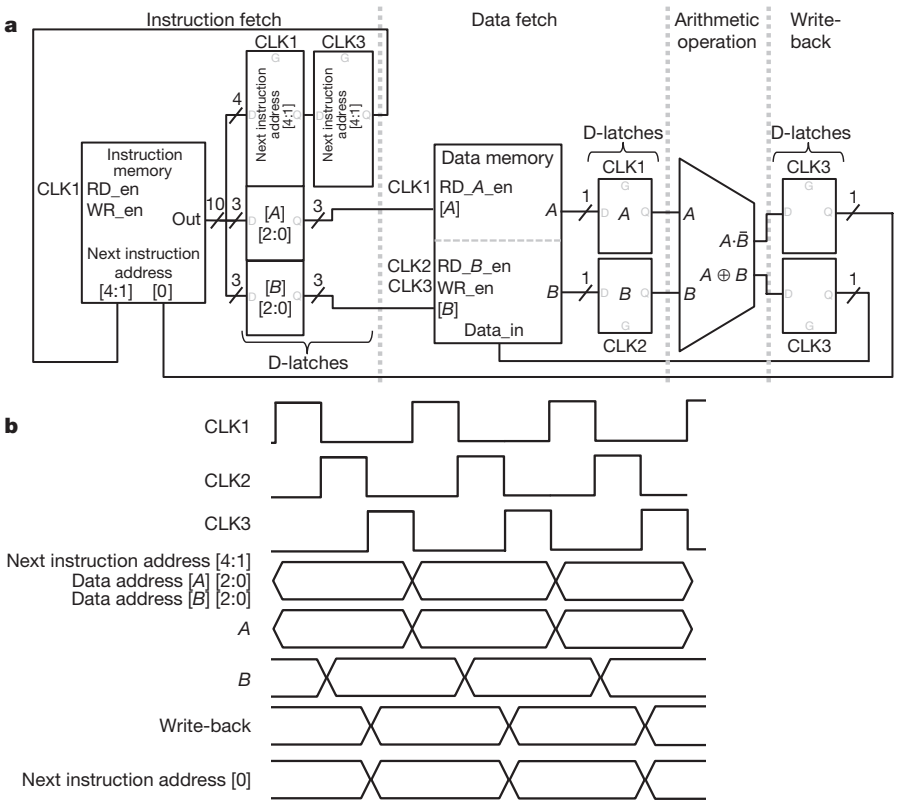



**Figure 2 | Schematic of CNT computer.** **a**, Schematic of the entire CNT computer, composed of the four subunits: instruction fetch, data fetch, arithmetic operation and write-back. All components apart from the memory are implemented entirely using CNFETs. CLK1–CLK3, Clock1–Clock3; D, D-latch input; Q, D-latch output; G, D-latch clock; RD_en, read enable (instruction memory); WR_en, write enable (instruction memory); RD_*A*_en, read enable address *A* (data memory); RD_*B*_en, read enable address *B* (data memory); Data_in, data for data memory write. **b**, Timing diagram of the CNT computer. The lines show the waveforms corresponding to each signal; of particular note are the transitions of the lower five signals with respect to the clock signals.

(4) Write-back: this task writes back the result of the SUBNEG ($B - A$) in the data memory at the address of the second data address. On the third clock (Clock3), the result from the arithmetic-operation unit is saved in two D-latches. Simultaneously, Clock3 enables the write-back to the data memory. D-latches from the instruction-fetch unit supply the data address, and the D-latch from the write-back stage supplies the value to be written.

A timing diagram depicting the above description and using three non-overlapping clocks is shown in Fig. 2b.

The CNFET computer is composed of 178 CNFETs, with each CNFET comprising ~10–200 CNTs, depending on relative sizing of the widths of the CNFETs. Figure 3 shows transistor-level schematics of the subcomponents, D-latches and the arithmetic unit. We use logic circuits that use only p-type transistors, because our CNFETs are p-type without modifications. Consequently, relative sizing of the widths of pull-up and pull-down CNFETs is crucial; the ratio of all pull-up CNFET widths to pull-down CNFET widths in our design is either 20:1 or 10:1 (Methods). There is a maximum of seven stages of cascaded logic in the computer, demonstrating our ability to cascade combinational logic stages, which is a necessity in realizing large digital systems.

The CNT-specific fabrication process is based on the process described in refs 21, 23, 27, and is described in detail in Methods. Importantly, the fabrication process is completely silicon-CMOS compatible owing to its low thermal budget (125 °C). We use standard cells for our subsystems, designed following the imperfection-immune methodology, which renders our circuits immune to both mis-positioned and metallic CNTs. Because this method ensures that the immunity to CNT imperfections is encapsulated entirely within standard cells, the fabrication is completely insensitive to the exact positioning of CNTs on the wafer and there is no per-unit customization, rendering our processing and design VLSI (very large-scale integration) compatible. The entire CNT computer is fabricated completely within a die on a single wafer. Each die contains five CNT computers, and each wafer contains 197 dies. There is no customization of any sort after circuit fabrication: all of the CNFETs and interconnects are predetermined during design, and there is no post-fabrication selection, configuration or fine-tuning of functional CNFETs. Just like any von Neumann computer, off-chip interconnects are used for connections to external memories. Our CNT-specific fabrication process and imperfection-immune design enables high yield and robust devices; waveforms of 240 subsystems (40 arithmetic logic units and 200 D-latches) from across a wafer are shown in Fig. 3. The yield of the subsystems, such as D-latches, typically ranges from 80% to 90%. The primary causes of yield loss—particles resulting in broken lithography patterns, adhesion issues with metal lift-off and variations in machine etch rates—are consequences of the limitations of performing all fabrication steps in-house in an academic fabrication facility.

A SEM image of a fabricated CNT computer is shown in Fig. 4a. To demonstrate the working CNT computer, we perform multitasking with our basic operating system, concurrently running a counter program and an integer-sorting program (performing the bubble-sort algorithm). Although CNFET circuits promise improved speed[2,4,8], our computer runs at 1 kHz. This is not due to the limitations of the CNT technology or our design methodology, but instead is caused by capacitive loading introduced by the measurement setup, the 1-μm minimum lithographic feature size possible in our academic fabrication facility, and CNT density and contact resistance (Methods). The measured and expected

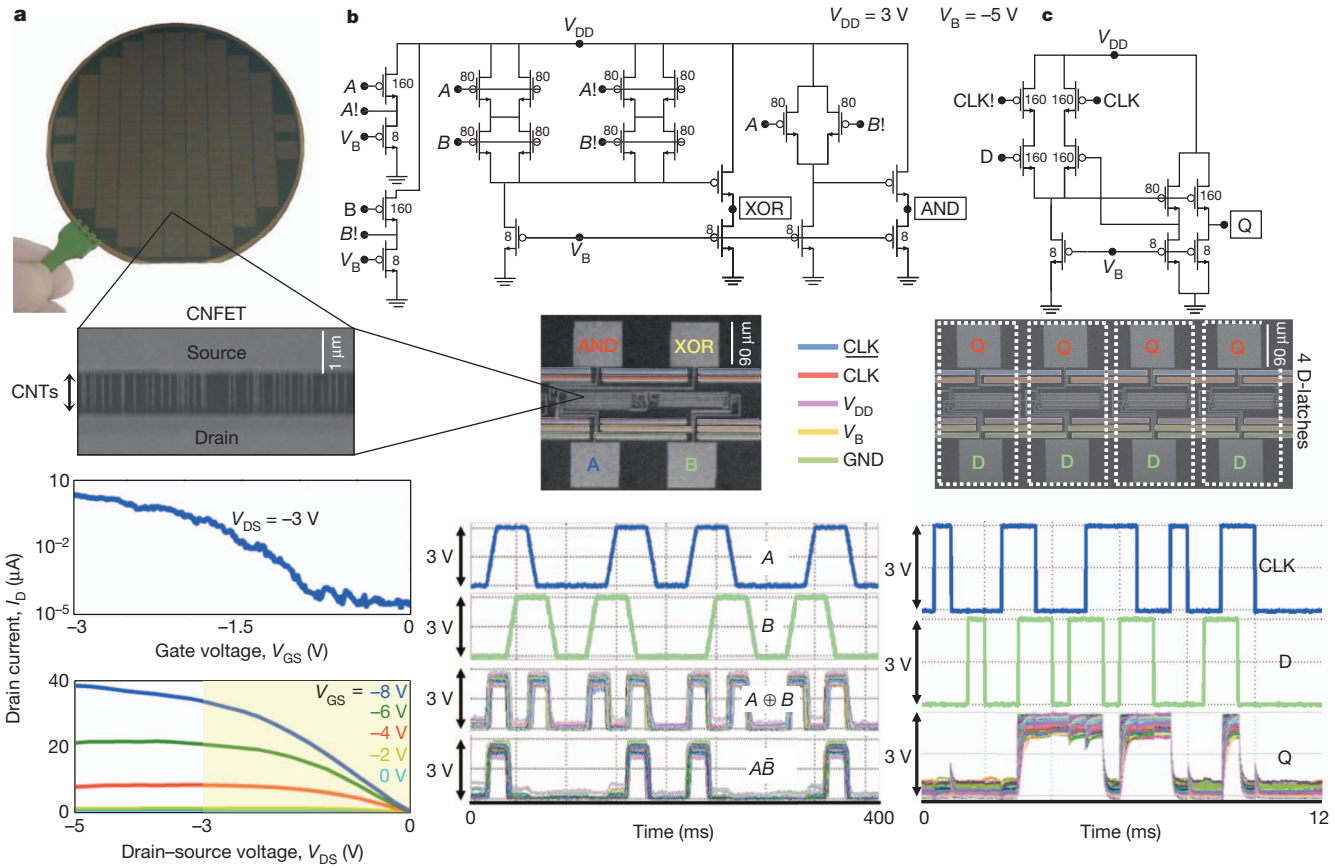



**Figure 3 | Characterization of CNFET subcomponents.** **a**, Top: Final 4-inch wafer after all fabrication. Middle: scanning electron microscope (SEM) image of a CNFET, showing source, drain and CNTs extending into the channel region. Bottom, Measured characterization (current–voltage) curves of a typical CNFET. The yellow highlighted region of the $I_D$–$V_{DS}$ curve shows the biasing region that the CNFET operates in for the CNT computer. **b**, Top: transistor-level schematic of arithmetic unit. Numbers are width of transistors (in micrometres). Middle: SEM of an arithmetic unit. Bottom: measured outputs from 40 different arithmetic units, all overlaid. **c**, Top: transistor-level schematic of D-latches. Numbers are width of transistors (in micrometres). Middle: SEM of a bank of 4 D-latches. Bottom: measured outputs from 200 different D-latches, all overlaid.

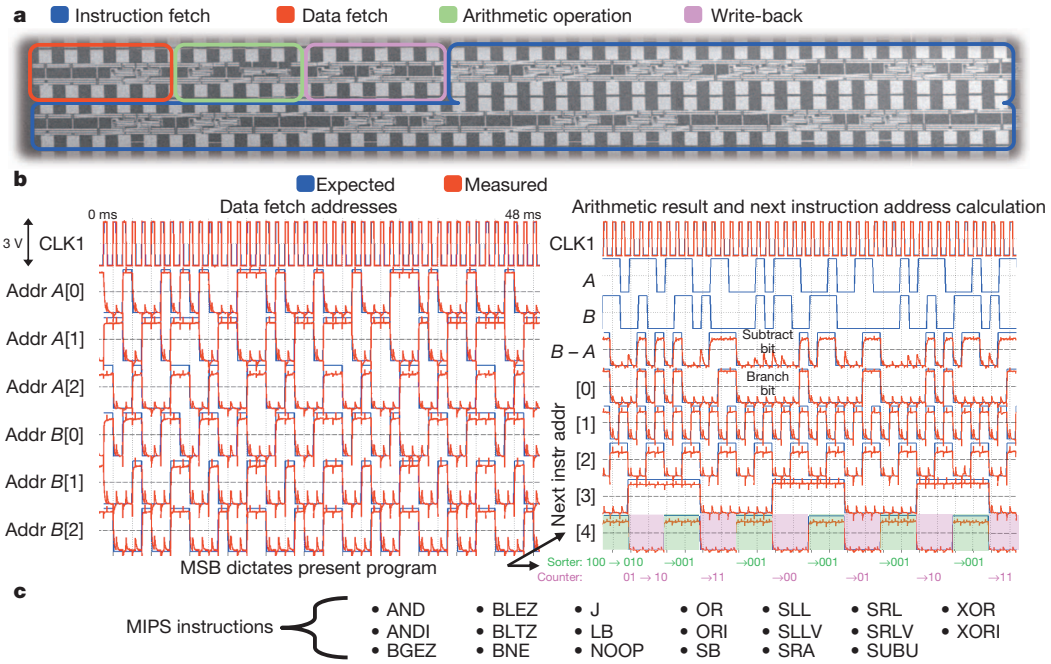



**Figure 4 | CNT computer results.** **a**, SEM of an entire CNT computer. **b**, Measured and expected output waveforms for a CNT computer, running the program shown in Fig. 1b. The exact match in logic value of the measured and expected output shows correct operation. As shown by the MSB (denoted [4]) of the next instruction address, the computer is switching between performing counting and sorting (bubble-sort algorithm). The running results of the counting and sorting are shown in the rows beneath the MSB of the next instruction address. **c**, A list of the 20 MIPS instructions tested on the CNT computer.

outputs from the CNT computer (Fig. 4b) show correct operation. To demonstrate the flexibility and ability of the SUBNEG computer to implement any arbitrary instruction, we additionally perform 20 MIPS instructions (Fig. 4c) on the CNT computer. Although the CNT computer operates on single-bit data values, this is not a fundamental limitation, because any multibit computation can be performed with a single-bit computer through serial computation[23]. Additionally, having shown the ability to cascade logic, fabricating a larger multibit CNT computer is not a fundamental obstacle, but rather affects only yield; as a demonstration, we show a two-bit arithmetic logic unit (composed of 96 CNFETs with a maximum of 15 stages of cascaded logic) in Extended Data Fig. 2 (see also Methods).

We have reported a CNT computer fabricated entirely from CNFETs, and have demonstrated its ability to run programs, to run a basic operating system that performs multitasking, and to execute MIPS instructions. To achieve this we used the imperfection-immune design methodology and developed robust and repeatable CNT-specific design and processing. This demonstration confirms that CNFET-based circuits are a feasible and plausible emerging technology.

## METHODS SUMMARY

The fabrication process is depicted in Extended Data Fig. 1. The CNTs are grown on a quartz substrate to yield highly aligned CNTs[14], and are transferred onto the target $SiO_2$ wafer[14]. Before CNT transfer, the wafer undergoes processing to define bottom-layer wires and the local back gates of the transistors[28]. Lithographically defined trenches are etched using a combination of dry plasma etch followed by wet etch, and are filled by electron-beam evaporation of platinum and smoothed by a subsequent plasma sputter etch. A 24-nm high-*k* dielectric of $Al_2O_3$ is deposited by atomic-layer deposition, and contact holes are etched through this layer to the embedded metal wires and gates through another combined dry- and wet-etch process. After CNT transfer, the source and drain (bilayers of palladium and platinum) are lithographically defined through a lift-off process, and mis-positioned CNTs are etched away using optical lithography followed by oxygen plasma[15]. A metal layer of gold is lithographically patterned with lift-off and connects every other source and drain, and separately connects every gate, effectively forming a single CNFET composed of all of the single CNFETs in parallel. Electrical breakdown is performed once on this entire structure to remove >99.99% of metallic CNTs[29,30]. This gold layer is then selectively etched away, and the top metal layer connecting the circuit in the proper configuration is lithographically patterned and deposited with lift-off.

**Online Content** Any additional Methods, Extended Data display items and Source Data are available in the online version of the paper; references unique to these sections appear only in the online paper.

**Received 12 May; accepted 24 July 2013.**


1. Franklin, A. D. *et al.* Sub-10 nm carbon nanotube transistor. *Nano Lett.* **12,** 758–762 (2012).
2. Wei, L., Frank, D., Chang, L. & Wong, H.-S. P. in *Proc. 2009 IEEE Intl Electron Devices Meeting* 917–920 (IEEE, 2009).
3. Chang, L. in Short Course *IEEE Intl Electron Devices Meeting* (IEEE, 2012).
4. Nikonov, D. & Young, I. in *Proc. 2012 IEEE Intl Electron Devices Meeting* 24–25 (IEEE, 2012).
5. Javey, A., Guo, J., Wang, Q., Lundstrom, M. & Dai, H. Ballistic carbon nanotube transistors. *Nature* **424,** 654–657 (2003).
6. Javey, A., Wang, Q., Kim, W. & Dai, H. in *2003 Intl Electron Devices Meeting Tech. Digest* 31–32 (IEEE, 2003).
7. Appenzeller, J. Carbon nanotubes for high-performance electronics—progress and prospect. *Proc. IEEE* **96,** 201–211 (2008).
8. Deng, J. *et al.* in *Proc. 2007 IEEE Intl Solid State Circuits Conf.* 70–78 (IEEE, 2007).
9. Iijima, S. Helical microtubules of graphitic carbon. *Nature* **354,** 56–58 (1991).
10. Martel, R. A. , Schmidt, T., Shea, H. R., Hertel, T. & Avouris, P. Single-and multi-wall carbon nanotube field-effect transistors. *Appl. Phys. Lett.* **73,** 2447 (1998).
11. Tans, S. J., Verschueren, A. R. & Dekker, C. Room-temperature transistor based on a single carbon nanotube. *Nature* **393,** 49–52 (1998).
12. Chen, Z. *et al.* An integrated logic circuit assembled on a single carbon nanotube. *Science* **311,** 1735 (2006).
13. Cao, Q. *et al.* Medium-scale carbon nanotube thin-film integrated circuits on flexible plastic substrates. *Nature* **454,** 495–500 (2008).
14. Patil, N., Lin, A., Myers, E. R., Wong, H.-S. P. & Mitra, S. in *Proc. Symp. VLSI Tech.* 205–206 (2008).
15. Patil, N. *et al.* Scalable carbon nanotube computational and storage circuits immune to metallic and mis-positioned carbon nanotubes. *IEEE Trans. NanoTechnol.* **10,** 744–750 (2011).
16. Shulaker, M. *et al.* in *Proc. 2013 IEEE Intl Solid State Circuits Conf.* 112–113 (IEEE, 2013).
17. von Neumann, J. First draft of a report on the EDVAC. *Ann. Hist. Comput.* **15,** 27–75 (1993).
18. McCluskey, E. J. *Logic Design Principles with Emphasis on Testable Semicustom Circuits* (Prentice-Hall, 1986).
19. Cao, Q. *et al.* Arrays of single-walled carbon nanotubes with full surface coverage for high-performance electronics. *Nature Nanotechnol.* **8,** 180–186 (2013).

20. Zhang, J. *et al.* Robust digital VLSI using carbon nanotubes. *IEEE Trans. CAD* **31,** 453–471 (2012).
21. Patil, N. *Design and Fabrication of Imperfection-Immune Carbon Nanotube Digital VLSI Circuits*. PhD thesis, Stanford Univ. (2010).
22. Patterson, D. A. & Hennessy, J. L. *Computer Architecture* (Kaufmann, 1990).
23. Lin, A. *Carbon Nanotube Synthesis, Device Fabrication, and Circuit Design for Digital Logic Applications*. PhD thesis, Stanford Univ. (2010).
24. Herken, R. (ed.) *The Universal Turing Machine: A Half-Century Survey* (Springer, 1995).
25. Nürnberg, P., Uffe, W. & Hicks, D. A grand unified theory for structural computing. *Metainformatics* **3002,** 1–16 (2004).
26. Jeffay, K., Donald, S. F. & Martel, C. U. in *Proc. Real-Time Systems Symposium* 129–139 (IEEE, 1991).
27. Shulaker, M. *et al.* Linear increases in carbon nanotube density through multiple transfer technique. *Nano Lett.* **11,** 1881–1886 (2011).
28. Bachtold, A., Hadley, P., Nakanishi, T. & Dekker, C. Logic circuits with carbon nanotube transistors. *Science* **294,** 1317–1320 (2001).
29. Collins, P. G., Arnold, M. S. & Avouris, P. Engineering carbon nanotubes and nanotube circuits using electrical breakdown. *Science* **292,** 706–709 (2001).
30. Patil, N. *et al.* in *Proc. 2009 IEEE Intl Electron Devices Meeting* 573–576 (IEEE, 2009).

**Acknowledgements** We acknowledge the support of the NSF (CISE) (CNS-1059020, CCF-0726791, CCF-0702343, CCF-0643319), FCRP C2S2, FCRP FENA, STARNet SONIC and the Stanford Graduate Fellowship and the Hertz Foundation Fellowship (M.M.S.). We also acknowledge Z. Bao, A. Lin, H. (D.) Lin, M. Rosenblum, and J. Zhang for their advice and collaborations.

**Author Contributions** M.M.S. led and was involved in all aspects of the project, did all of the fabrication and layout designs, and contributed to the design and testing. G.H. wrote the SUBNEG and testing programs, and contributed to the design and testing. N.P. contributed to the design, and N.P., H.W. and H.-Y.C. contributed to developing fabrication processes. H.-S.P.W. and S.M. were in charge and advised on all parts of the project.

**Author Information** Reprints and permissions information is available at www.nature.com/reprints. The authors declare no competing financial interests. Readers are welcome to comment on the online version of the paper. Correspondence and requests for materials should be addressed to M.M.S. (maxms@stanford.edu).

## METHODS

The fabrication process is depicted in Extended Data Fig. 1.

**CNT growth and transfer.** The CNTs are grown by chemical-vapour deposition with methane at 865 °C. The growth substrate is an annealed quartz substrate, with parallel catalyst stripes of iron lithographically patterned on the wafer. Quartz is used to achieve 99.5% alignment of the CNTs, which align along the crystalline boundary owing to a minimized Lennard–Jones potential in this orientation[14]. After growth, the quartz wafer with CNTs is coated with 150 nm gold, and a thermal release tape is applied on top of the gold. When this tape is peeled from the wafer, it peels off the gold with embedded CNTs from the quartz wafer. The tape is then applied onto the target wafer and heated to 125 °C, at which point the thermal release tape loses adhesion and is removed from the wafer, leaving the gold with embedded CNTs on the target wafer. The surface of the wafer undergoes oxygen and argon plasma etching to remove any residue from the tape, followed by a selective wet etch to remove the gold, leaving exposed, highly aligned CNTs on the wafer[14].

**Local back gate.** Before transfer, the target wafer is first prepared, starting with a silicon wafer with 110 nm thermal oxide growth. To form the local back gate[28] and bottom layer of wires, a two-layer resist stack is lithographically patterned on the surface. Following development of the pattern, the wafer goes through a quick oxygen plasma de-scum, followed by an anisotropic $O_2/SF_6$ plasma etch. After the plasma etch, a quick HF dip is used to smooth the surface and remove any sidewall deposition from the plasma etching. Next, an adhesion layer of Ti followed by Pt is evaporated, filling the trenches etched in the previous step. The bilayer of resist is dissolved away, lifting off the extra metal and leaving the metal in the trenches. An argon sputter etch follows, and, owing to the difference in etch rate between the Pt and $SiO_2$, the surface of the wafer is smoothed until the offset between the local back gate height and the wafer is less than a nanometre.

**Initial transistor fabrication.** We use ~24 nm $Al_2O_3$ as our high-$k$ back-gate dielectric. This is deposited through atomic-layer deposition on the wafer described above, covering the local back gates and bottom-level wires. Before CNT transfer, the deposited surface undergoes an oxygen plasma etch to clean the surface of any contaminants and a forming gas anneal, followed by the CNT transfer process described above. Immediately following transfer is source–drain definition of the individual transistors. A bilayer of resist is patterned and developed, and a bilayer of 20 nm Pd and 20 nm Pt is deposited for both the source and drains. This is followed by a traditional lift-off process. In addition to the source and drain, a second layer of metal wiring is patterned and deposited. This second layer of metal wiring is permanent through the rest of the process. After the metal deposition, mis-positioned and unneeded CNTs are removed by covering the active area of the transistors with photoresist and etching away the unprotected CNTs with oxygen plasma. The layout of the active area of the transistors follows the mis-positioned CNT immune design[20,21], and guarantees that no mis-positioned CNTs can cause incorrect logic function. This renders the circuit immune to mis-positioned CNTs. Contacts to the bottom-layer wires and local back gates are lithographically defined and etched with an $Ar/CL_2/BCL_3$ plasma etch, followed by HF dip, with the embedded metal acting as a natural etch stop.

**Metallic CNT removal.** To ensure high $I_{on}/I_{off}$ ratios and correct logic functionality, it is necessary to remove >99.99% of the metallic CNTs from the circuit, while leaving the semiconducting CNTs predominantly intact. This is achieved through electrical breakdown, which biases the gate of the transistor to turn the semiconducting CNTs off, and pulses a large current through the metallic CNTs, causing joule self-heating until the metallic CNTs oxidize and are removed, thus no longer conducting current[29]. Rather than performing breakdown on the individual transistors, we employ VLSI-compatible metallic CNT removal[30] (VMR). VMR allows electrical breakdown to be performed on the chip scale. To do so, we lithographically define and pattern a gold layer through the lift-off processes described above. The gold is patterned to short every gate, source and drain together. This effectively forms a single large CNFET, composed of all of the single CNFETs connected in parallel. The shorted structures make use of the power rails and clock distribution networks to minimize area overhead. We then perform electrical breakdown on the entire structure once, enabling quick and efficient breakdown of hundreds of transistors and thousands of CNTs simultaneously (though this is not a fundamental limitation of the size of a VMR structure). After electrical breakdown, the gold layer is removed. The third and final metal layer of Pt with an adhesion layer of Ti is deposited and lifted off, forming the final circuit layout configuration.

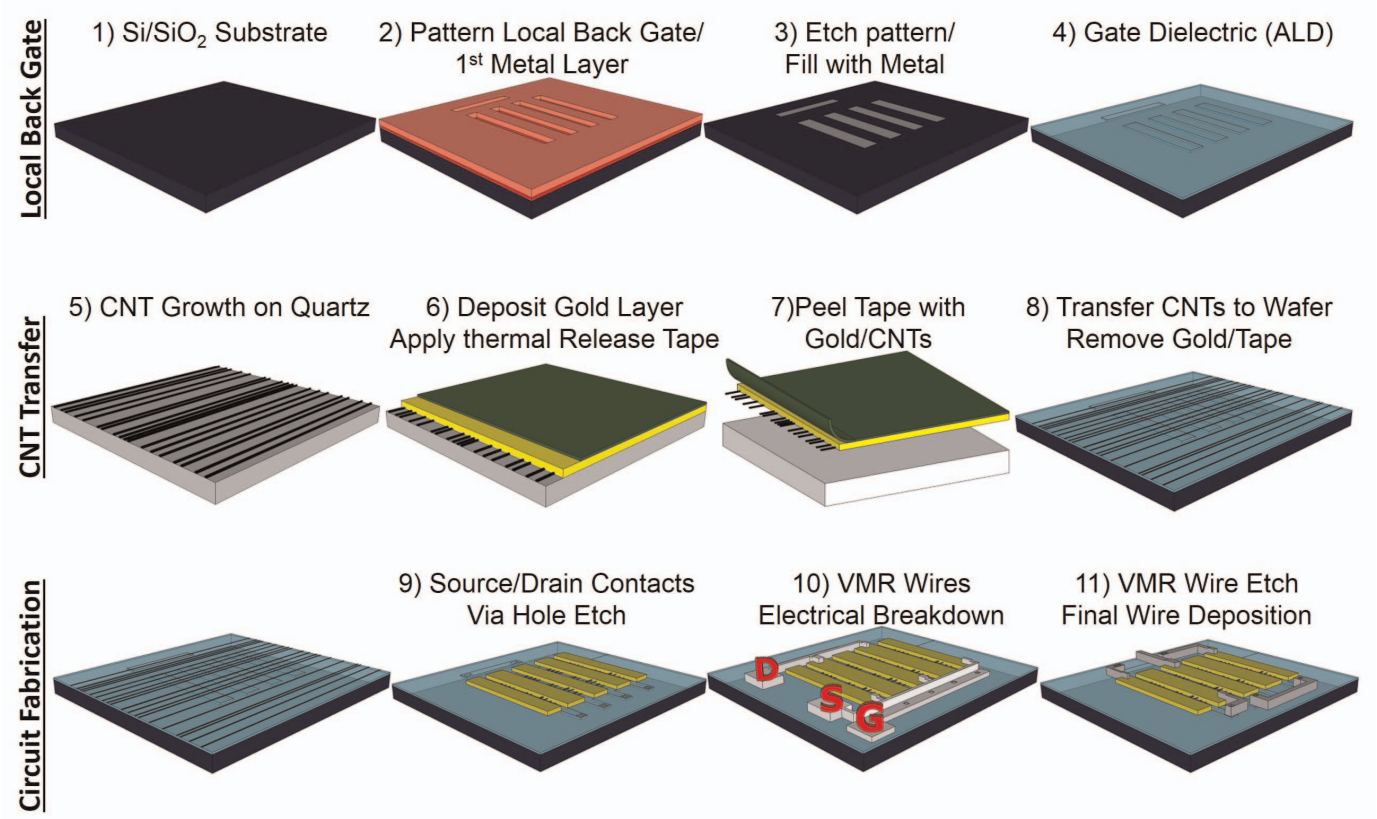
**Test set-up.** As shown in Fig. 4a, the CNT computer has four rows of probe pads, each containing 39 pads. A custom probe card is used to probe all of the pads simultaneously, although many of the pads are unused (and are simply present to ensure that the probe tips from the probe card always land on metal). Through the probe card, the pads are either connected to a supply voltage ($V_{DD}$, GND, $V_{BIAS}$) or to the inputs or outputs of the computer (the address outputs and input values to and from the off-chip memories). All other connections are made on-chip, as shown in Extended Data Fig. 3. A National Instrument DAQ (data acquisition hardware, #9264) is used to interface with the probe card and read and write the inputs and, respectively, outputs to the CNT computer, and Agilent oscilloscopes (#2014A) are additionally used to record the analogue traces of the outputs of the CNT computer (Fig. 4b).

**Biasing.** The biasing scheme for the circuits is shown in Fig. 3, with $V_{DD} = 3$ V and $V_{BIAS} = -5$ V. There is no individual tuning of biasing voltages for individual transistors. Scaled supply voltages can be achieved by scaling the transistor channel lengths from 1 μm at present (due to the limitations of academic fabrication capabilities) to smaller channel lengths[1].

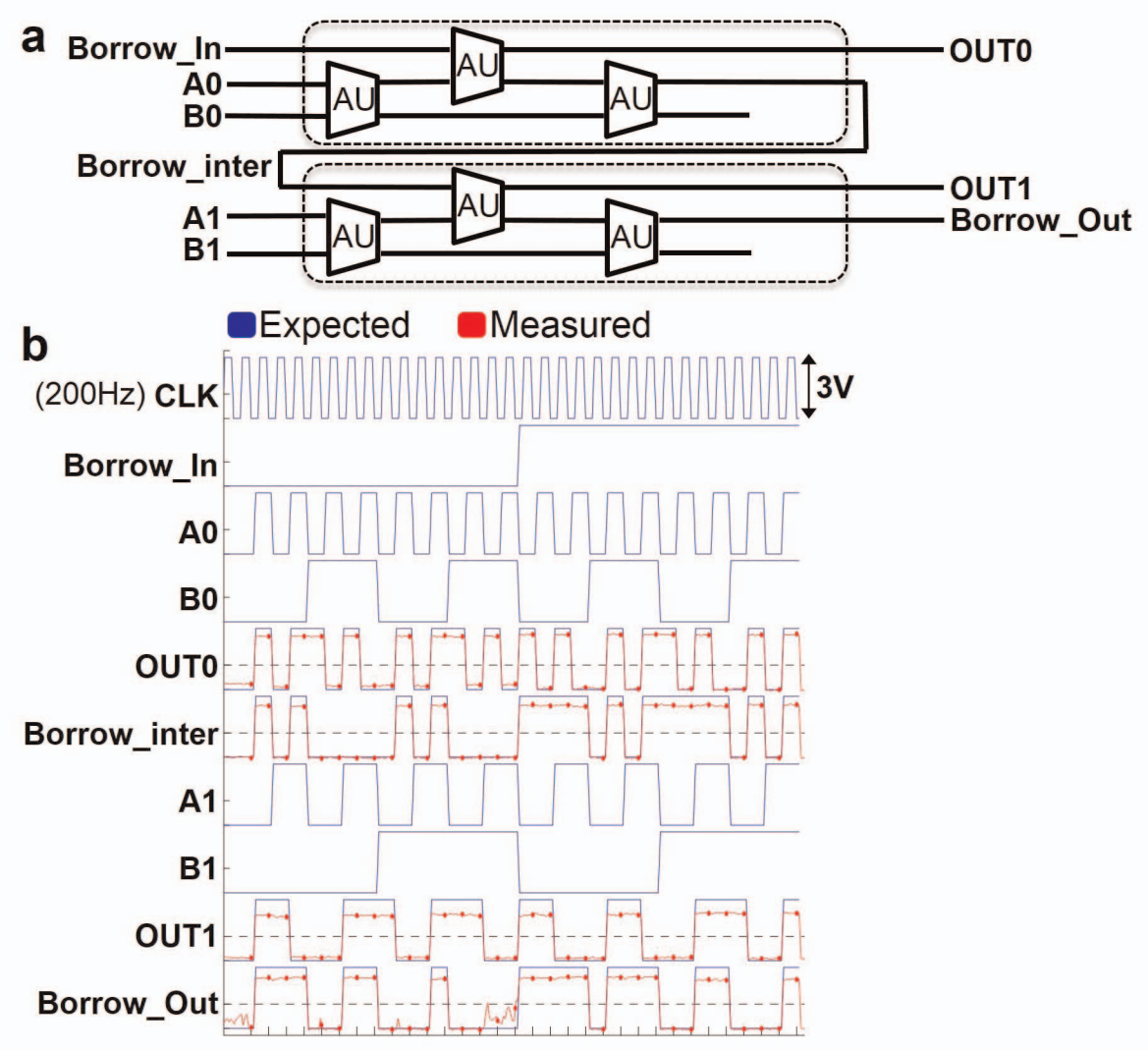
**Speed.** The probe pads and probe card with connecting wires used to connect to the CNT computer add additional capacitive loading to the circuit, limiting the frequency of operation to 1 kHz. However, this is not a fundamental limitation, because commercial chips are packaged and connected to memory and external devices without the use of probe cards, greatly reducing parasitic capacitances. The speed is also limited by the fact that the CNFET gate length is ~1 μm, set by the minimum lithographic feature that can be patterned in our academic clean-room; in field-effect transistors, on-current increases as the gate length decreases[1]. Lithographic overlay accuracy of ~200 nm further increases parasitic capacitances resulting in reduced speed. Moreover, the CNT density in this work is ~5 CNTs per micrometre, whereas the target CNT density for increased current drive is 100–200 CNTs per micrometre[8]. Several published approaches show promising methods of achieving this target CNT density[27]. CNT contact resistance must also be improved for high-performance circuits, and is another source of variation between devices.

**PMOS-only logic.** Logic circuits which use only p-type transistors are known as PMOS-only logic. The design of PMOS-only logic, which is well documented in the literature, is shown in Extended Data Fig. 4. Extended Data Fig. 4a depicts a PMOS-only inverter, whereas Extended Data Fig. 4b depicts a PMOS-only NAND gate. As is apparent from comparison of the two circuits, the pull-down network is always a single p-type transistor, whose gate is biased to remain on continuously. The pull-up network follows the design of typical CMOS circuits. The p-type transistors in the pull-up network create a conducting path from the output to $V_{DD}$ when the output should be logic 1. When the output should be logic 0, the pull-up network is designed to no longer have a conducting path to $V_{DD}$, and, thus, the single p-type transistor in the pull-down network pulls the output to logic 0. The relative sizing of the pull-up network and pull-down network is critical, because the pull-down network is always biased on. Thus, when the pull-up network should pull the output to logic 1, the pull-down network will still be attempting to pull the output to logic 0. Thus, in our design, the transistors in the pull-up networks are always sized with a width of 10–20 times the pull-down transistor width. Exact transistor sizing is shown in Fig. 3.

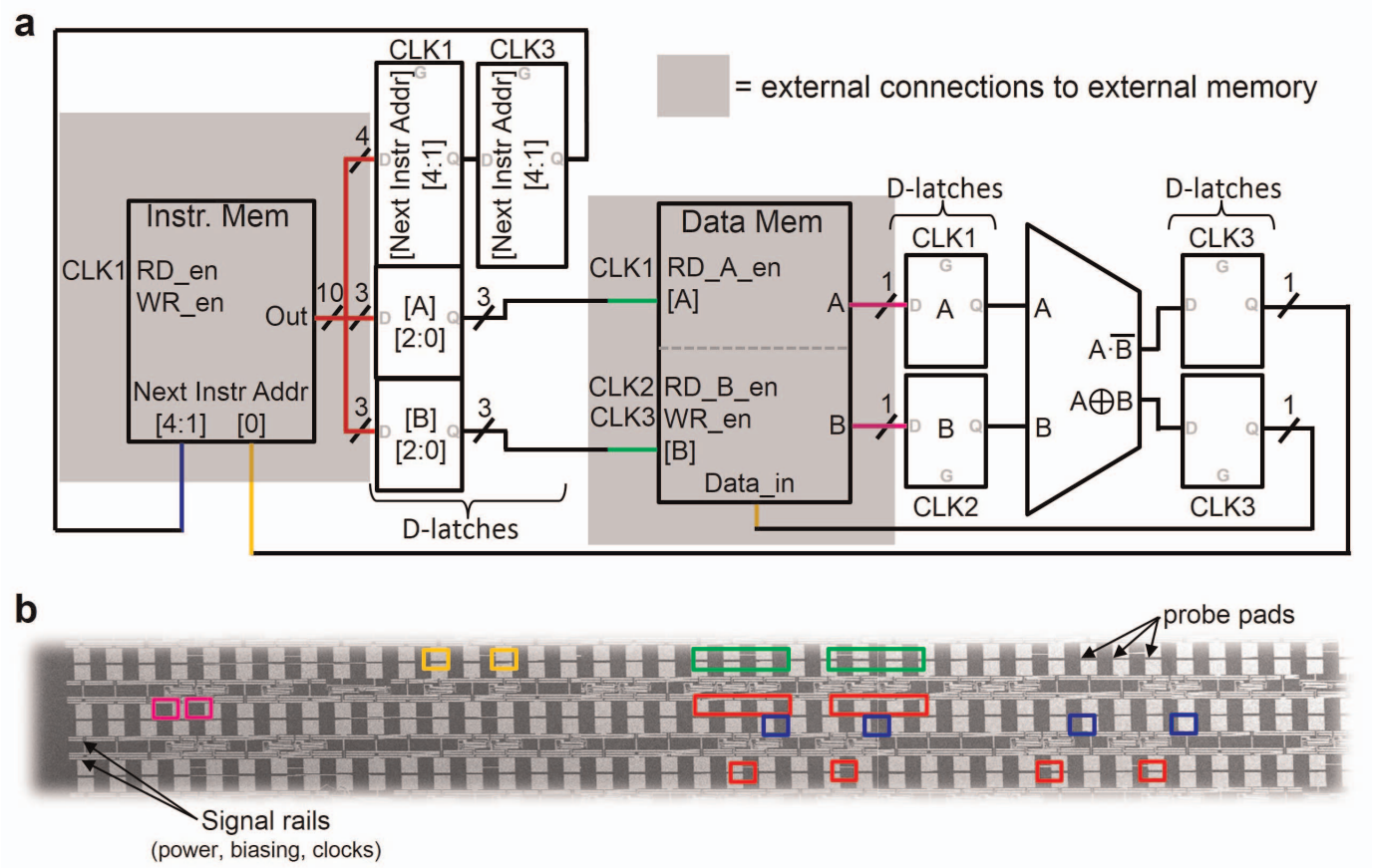
**Multibit arithmetic unit.** Additionally, having shown the ability to cascade logic, fabricating a larger multibit CNT computer is not a fundamental obstacle, but rather only affects yield; as a demonstration, we show a two-bit arithmetic unit (composed of 96 CNFETs with a maximum of 15 stages of cascaded logic). The two-bit arithmetic unit is shown in Extended Data Fig. 2. The output waveform tests for all possible inputs, and shows correct operation. Additionally, we show that the circuits regenerate the signal between stages, a necessity for cascading digital logic, by highlighting the noise in the 'borrow out' output. Even with noise somewhere within the arithmetic unit (which can have multiple causes: a stage with low swing, electrical noise on the inputs, mobile charges in an oxide and so on), owing to the gain of each stage the final output levels (logic 0 and logic 1) always stay either below or above the threshold for logic 0 or logic 1, respectively (as shown by the horizontal black dotted line).



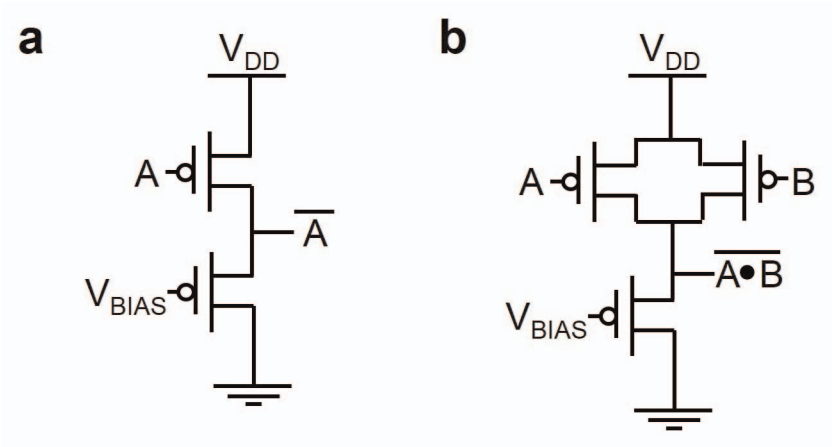

**Extended Data Figure 1 | Fabrication flow for the CNT computer.** Steps 1–4 prepare the final substrate for circuit fabrication. Steps 5–8 transfer the CNTs from the quartz wafer (where highly aligned CNTs are grown) to the final $SiO_2$ substrate. Steps 9–11 continue final device fabrication on the final substrate.




**Extended Data Figure 2 | Multibit arithmetic unit.** **a**, Schematic of a two-bit arithmetic unit, comprising six individual arithmetic logic units (ALU) as shown in Fig. 3b. **b**, Measured and expected output waveforms testing all possible input combinations of the two-bit arithmetic unit, showing correct operation.




**Extended Data Figure 3 | Internal versus external connections of CNT computer.** **a**, Schematic of the CNT computer, showing that all connections are fabricated on-chip and that only signals reading or writing to or from an external memory are connected off-chip. **b**, SEM of the CNT computer, showing which connections are made to and from the CNT computer from the probe pads. The SEM is colour-coded to match the coloured wires in **a**.




**Extended Data Figure 4 | PMOS-only logic schematics.** **a**, Schematic of PMOS-only inverter. **b**, Schematic of PMOS-only NAND gate.