

# A Carbon Nanotube Transistor based RISC-V Processor using Pass Transistor Logic

Aporva Amarnath, Siying Feng, Subhankar Pal, Tutu Ajayi, Austin Rovinski, Ronald G. Dreslinski  
University of Michigan, Ann Arbor, MI 48109  
{aporvaa, fengsy, subh, ajayi, rovinski, rdreslin}@umich.edu

**Abstract**—With silicon-based transistors approaching their scaling limits, multiple successor technologies are competing for silicon’s place. Due to recent fabrication breakthroughs, one promising alternative is the carbon nanotube field-effect transistor (CNTFET), which uses carbon nanotubes as the channel medium instead of silicon. Although logic gates using CNTFETs have been demonstrated to provide up to an order of magnitude better energy-delay product (EDP) over silicon-based counterparts, system-level design using CNTFETs show significantly smaller EDP improvement because of the critical path of the design, output load capacitance and corresponding drive strengths of gates. In this paper, we address this challenge by exploring various architectural design choices using CNTFET-based pass transistor logic (PTL) and create an energy-efficient RISC-V processor. While silicon-based design traditionally prefers complementary logic over PTL, CNTFETs are ideal candidates for PTL due to their low threshold voltage, low power dissipation, and equal strength p-type and n-type transistors. By utilizing PTL to design modules that lie on the processor’s critical path, systems can efficiently exploit CNTFET’s potential benefits. Our results show that while a CNTFET RISC-V processor using complementary logic achieves a  $2.9\times$  EDP improvement over a silicon design, using PTL along the critical path components in the ALU can boost EDP improvement  $5\times$  as well as reduce area by 17% over 16 nm silicon CMOS.

## I. INTRODUCTION

With the end of Dennard Scaling and the pending demise of Moore’s Law, silicon chip manufacturers are facing a widespread plateau in performance improvements. Clock frequencies and power have already stopped scaling due to the power wall [7], and many industry experts predict physical scaling to end with the 5 nm node in 2021 [10].

Extensive research is being undertaken towards the discovery of new alternative technologies to continue performance scaling while maintaining power density, including spintronics, quantum computing, and carbon nanotubes. Carbon nanotube field-effect transistors (CNTFETs) are one of the most promising competing technologies available, offering high current-carrying capacity [9], high carrier velocity [12], and exceptional electrostatics due to their ultra-thin body [2]. In addition, CNTFETs have made great strides in manufacturability in terms of both device scaling and yield, and they require relatively few changes to the silicon manufacturing process [4].

Prior work has investigated the impact of CNTFETs on small-scale designs, such as individual transistor properties or complementary gates [5, 11]. Bobba *et al.* have explored the impact of replacing Si-FETs with CNTFETs at the system level, designing an OpenRISC processor [3]. However, their processor’s EDP improvement is much lower in comparison

to their gate-level EDP reduction over silicon. This is primarily due to the critical paths within the design, output load capacitance and corresponding drive strengths of gates while creating larger designs. The EDP improvement at system-level will further be diminished due to variation caused by the fabrication process. Hence, this calls for more efficient design techniques and a better-suited logic family to reclaim the order of magnitude improvements that CNTFETs are capable of delivering. One of the key properties of CNTFETs is their low threshold voltage and low power dissipation, which lends very well to the use of a more efficient logic family like pass transistor logic (PTL) [6]. CNTFET-based systems can greatly improve EDP through the use of multiple logic families, and in particular with the use of PTL.

In this paper, we take advantage of CNTFETs’ exceptional electrical properties to explore the architectural design considerations that need to be made when creating large-scale CNTFET designs using PTL. We build a RISC-V pipeline using both complementary logic and PTL. Specifically, we compare several microprocessor components in 16 nm FINFET-based CMOS silicon (Si-CMOS), 16 nm complementary CNTFET (CCNT), and 16 nm PTL-CNTFET (PTL-CNT). We then expand our analysis to a full RISC-V pipeline design and evaluate the system-level impacts.

We show that the CNTFET RISC-V pipeline achieves a mere  $2.9\times$  improvement in energy-delay product (EDP) over a silicon-based design at 0.4 V. We improve this by using PTL for the critical path components and CCNT for the rest of the design, gaining a  $5\times$  improvement in EDP and a 17% reduction in area over 16 nm silicon CMOS.

## II. MOTIVATION

Historically, CNTFET designs have been plagued by manufacturing issues, particularly when creating a standard cell-based design. However, recent advances in fabrication techniques have made high-yield, reliable CNTFET devices possible for both p-type and n-type transistors, enabling the use of traditional CAD design flows. CNTFETs use carbon nanotubes as the channel medium between the source and the drain, instead of silicon. Hence, the behavior of a CNTFET is similar to a Si-FET: we observe a linear region followed by a saturation region in the drain current,  $I_{DS}$ , as a function of increasing gate-source voltage,  $V_{GS}$  [1]. In this section, we briefly discuss recent fabrication breakthroughs, provide an initial characterization of the device, and demonstrate why PTL is a promising logic family for CNTFET-based designs.

### A. CNTFET Fabrication

Although CNTFETs have faced several difficulties in efficient fabrication, recent techniques have improved the feasibility of CNTFET manufacturing. Shulaker *et al.* have demonstrated highly aligned CNTs with a density ( $\rho_{cnt}$ ) of about 100 CNTs/ $\mu\text{m}$  through chemical vapor deposition. Their method involves growing CNTs on a quartz substrate and repeatedly transferring them onto a wafer [20]. Hongsik *et al.* propose a technique where they fabricate and purify CNTs separately and suspend them on the substrate. Following this, they attract the CNTs into adhesive-filled trenches for alignment, resulting in a yield density of 20 CNTs/ $\mu\text{m}$  [17]. Recently, Brady *et al.* have achieved a  $\rho_{cnt} \approx 50$  CNTs/ $\mu\text{m}$  using the floating evaporative self-assembly (FESA) method [4]. Franklin *et al.* [8] characterize multiple FETs fabricated with varying width from 3  $\mu\text{m}$  to 15 nm on one CNT. Data extracted from these FETs are used to make more realistic CNTFET models [16].

### B. CNTFET Characterization

While integrated circuits are predominantly composed of Si-CMOS, CNTFETs offer a large number of advantages. In this section, we seek to quantify these benefits to understand how CNTFETs can be leveraged over Si-CMOS logic.

1) *Complementary Logic*: To investigate the characteristics of CNTFETs, we compare CCNT to Si-CMOS by using SPICE models of a minimum-sized 16 nm Si-CMOS inverter and an equivalent width 16 nm CCNT inverter. In Figure 1, we demonstrate the performance of the CNTFET inverter using fan-out-of-four (FO4) analysis. Our characterization in Figures 1a-1d shows that CNTFETs outperform silicon both in terms of energy and EDP across the voltage range. However, CNTFETs under-perform in comparison to Si-CMOS in FO4 delay at higher supply voltages due to the high contact resistance in CNTFETs. This changes at lower voltages (approaching 0.4 V), where CNTFETs edge out Si-FETs, because of CNTFETs' higher current properties at lower voltages. Figure 1d, in particular, shows that as the supply voltage decreases, the EDP advantage of CNTFET over Si-FET increases.

While previous work has theorized up to an order of magnitude in EDP improvement for a CCNT-based inverter over Si-CMOS at low voltages [3, 8], they used theoretical models that did not include factors such as the contact resistance and variable CNT pitch, which are present in CNTFETs that can be fabricated today. These properties limit the gains of CNTFETs to less than the theoretical numbers. Overall, we observed a  $1.8\times$  improvement in EDP using models based on experimental data at 0.4 V.

2) *Pass Transistor Logic (PTL)*: Traditionally, Si-FET designs avoid using PTL because of the rapid threshold voltage drop across each additional PTL gate. Restoring logic is often used to balance this drop, however this negates the area, energy, and delay benefits of PTL. CNTFETs possess three key properties that Si-FETs do not: CNTFETs have a *very low threshold voltage*, while having a *low power dissipation* and *equal strength PFETs and NFETs*. With these key properties,

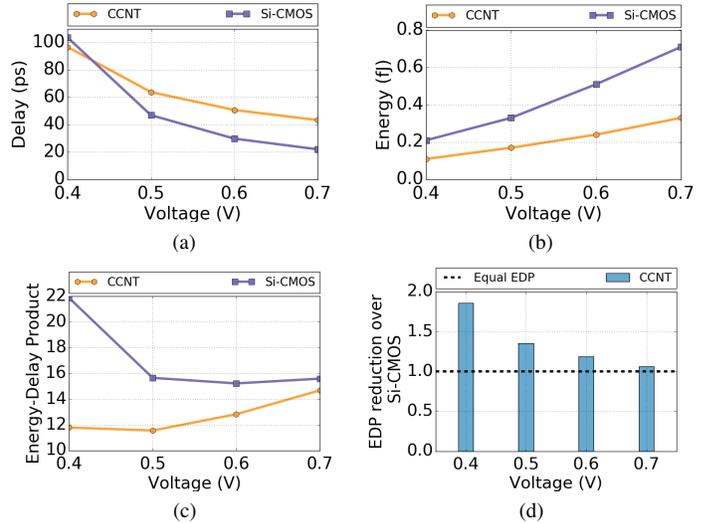


Fig. 1. Comparison of FO4 inverter in CCNT and Si-CMOS

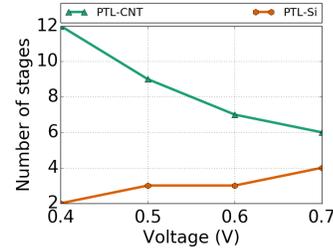


Fig. 2. Restoring logic for cascaded full adders

CNTFETs have been shown to enable PTL as a viable logic family [6].

However, to build larger designs using PTL, restoring logic is required. Figure 2 demonstrates the impact of using PTL with CNTFETs. We show the number of stages after which a restoring buffer needs to be placed for cascaded full adders in both PTL-CNT and PTL-Si. For silicon, PTL requires frequent restoring logic (every 2-4 stages), which only worsens as the supply voltage decreases. PTL-CNT, however, requires much less frequent buffering due to its low threshold voltage and requires  $6\times$  fewer buffers than PTL-Si at 0.4 V. The buffering for PTL-CNT actually worsens as voltage increases, due to high contact resistance in the CNTFETs, although the total amount of required buffering remains superior.

From this initial characterization, we find that CNTFETs outperform comparable Si-FETs in terms of EDP and are more amenable to PTL.

### III. RELATED WORK

Leveraging the availability of theoretical CNTFET models, prior works have constructed the basic building blocks of a processor using CNTFETs. Cho *et al.* [5] compare various CNTFET-based standard cells against their counterparts made using Si-CMOS. Kumar *et al.* [11] propose a low-power full adder using CNTFETs, showing an 80% power reduction in comparison to a Si-CMOS based one. Most of the work, however, has either been fragmented at the transistor-level or involved small building blocks.

In the work by Ding et al. [6], the authors explore building basic PTL gates using CNTFETs. They also calculate the output voltage levels of a PTL-CNT single-bit adder and subtractor, and demonstrate a functional multiplexer and D-latch. However, their work neither studies scaling PTL to larger blocks, nor the challenges that accompany it.

Prior work has looked into building full systems based on CNTFET technology. In the work by Shulaker et al. [19], the authors fabricate and demonstrate a functional, Turing-complete, subneg-based one-instruction-set computer. Further, Bobba et al. [3] show a 1.5x improvement in EDP of an OpenRISC processor, built using yield-enhancing standard cells, over Si-CMOS at 16 nm. However, these do not investigate the potential EDP improvement in system-level design that CNTFETs provide in gate-level designs, nor do they explore the benefits of a suitable logic family, like PTL.

To the best of our knowledge, our work is the first of its kind to construct an entire CNTFET-based RISC-V processor with all its critical-path components such as the full adder, ALU, multiplier, and registers using PTL-CNT. We employ a pessimistic CNTFET model to account for process variation, yet are able to demonstrate EDP improvements exceeding those that have been reported previously [3].

#### IV. RISC-V PROCESSOR PIPELINE

To address the challenges of system-level design and optimization of CNTFET-based systems, we build a single processor pipeline design using 3 different techniques: Si-CMOS, CCNT, and a hybrid (CCNT + PTL-CNT) configuration.

For our analysis, we use the V-scale core, which is a 32-bit, single-issue, in-order, 3-stage pipelined processor [14]. V-scale is an open-source design implemented in Verilog and is comparable to an ARM Cortex-M0 core. It is based on the open RISC-V instruction set architecture [21]. The critical modules of the core are implemented in each of the chosen configurations (Si-CMOS, CCNT and PTL-CNT) and then integrated into the full system.

The processor's ALU performs 14 different operations, including add/subtract, shift and comparison. We first implement the full adder circuit in the three different configurations. For comparison, we implement the 32-bit adder both as a ripple-carry and a Kogge-Stone design. A ripple-carry adder (RCA) consists of 32 full adders cascaded one after another and a Kogge-Stone adder (KSA) is a tree implementation of the carry-look ahead adder. While the KSA is faster and more energy-efficient than an RCA, it has a larger routing congestion and area [18]. Therefore, most present-day processors use sparse-tree adders that are a hybrid of both KSA and RCA.

However, PTL implementations of these adders require custom addition of restoring logic between the stages, as discussed in Section II-B2, due to varying loads seen by each transistor, especially for sparse-tree adder designs closer to a KSA.

The multiplier is implemented as a 32-bit, two-stage array-based pipelined multiplier. It uses carry-save adders, which are a row of full and half adders cascaded one after another. As with the ripple-carry adder, the multiplier unit also requires restoring logic in the carry-save adders when implemented

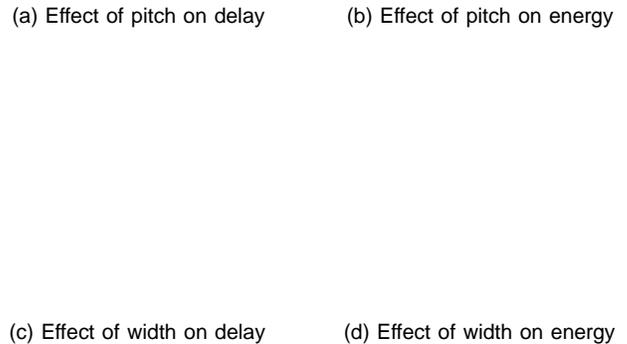


Fig. 3. Varying pitch and width of the CNTFET

in PTL. These buffer insertions are periodic and are placed optimally to reduce the critical path delay and energy.

#### V. METHODOLOGY

This section details the design methodology used for our evaluation. We include descriptions of how our models were created and how we leveraged them to build standard cell libraries. Finally, we detail how we use those libraries to create custom blocks and the final V-scale pipeline.

##### A. Operating Voltage

Threshold voltage of the intrinsic CNT channel in a CNTFET can be approximated to the half band gap, which is an inverse function of the diameter [12]. For a 10% diameter (0.2 nm) variation, we get a threshold voltage of 0.33-0.39 V. Hence, 0.4 V is selected to be the lower bound of supply voltage scaling in the voltage study.

Simulations are performed using the 16 nm Virtual Source CNTFET HSPICE model from Stanford University's Nanoelectronics Group [16]. The model is built on experimental data collected from multiple transistors built on one CNT with varying channel lengths from 3m to 15 nm. However, the model assumes CNTs are perfectly aligned, equally spaced and are of a fixed diameter. Hence, to address this, we choose slightly more pessimistic design parameters, as described in the following subsections.

##### B. CNTFET Design Parameters

The strength of a CNTFET is determined by the width of the transistor,  $W$ , as well as the CNT pitch,  $p_{cnt}$ . While high  $p_{cnt}$  has been reported in previous work, the control of  $p_{cnt}$  still remains to be mastered. Lee et al. predict that a density of 180 CNTs/  $\mu\text{m}$  is required to meet the ITRS targets of off-state and on-state currents at the 5 nm technology node [13].

Considering these features of CNTFETs, we study the effects of varying  $p_{cnt}$  and  $W$  on an FO4 inverter's delay and energy as shown in Figure 3. While the delay increases with

increasing CNT pitch), the energy increases with increasing 1) Full Adder: We compare a 20 transistor PTL-based transistor width ( $W$ ). We also see that has a minimal full adder implementation against a traditional CCNT-based effect on energy. The decrease in delay from decreasing 28 transistor mirror adder [18] as well as its counterpart is countered by the increase in power due to an increase in Si-CMOS. We designed this 20T full adder to obtain a the number of CNTs ( $N_{CNT}$ ). Similarly, increasing  $W$  has no fast Sum and  $C_{out}$  with only two transistors on the critical effect on delay as the FO4 inverter sees an equivalent increase in path, as shown in Figure 4. We reduced the load  $C_{out}$  in its output load capacitance. We choose a pessimistic pitch by de-multiplexing the shared part of the circuit with Sum, of 40 nm to incorporate worst case variation of CNT pitch creating two separate circuits to reduce degeneration during and removal of metallic-CNTs. This pitch value is used for cascading of the full adder for larger blocks, unlike the adder the rest of the CNTFETs characterized in this paper, and is a and subtractor built by Ding et al. [6]. line with contemporary fabrication techniques.

Further, for ease of area comparison against Si-CMOS transistors, we approximately match the width of the minimum-sized transistor in Si-CMOS to our minimum-sized transistor, i.e. a 4- $n$  Si-FET of width 240 nm (about 60 nm contributed by each  $n$ ) is matched to a CNTFET of width 200 nm, resulting in at least 5 CNTs per minimum sized transistor.

### C. Implementation

Since CNTFETs have similar characteristics to Si-FETs, it is fairly straightforward to derive basic CNTFET gates from already existing Si-FET gates. Using these gates, we created a CCNT standard cell library to analyze the system-level delay, energy and EDP improvement over Si-CMOS. Similarly, we created a PTL-CNT library of the basic cells required for the ALU and multiplier units. We performed synthesis of the processor using Synposys Design Compiler and preserved the boundaries around the ALU and multiplier units. These components were separated so that they could be probed individually. The gate-level netlist obtained from synthesis was then converted into an HSPICE netlist for each unit, using the CCNT and PTL-CNT standard cell libraries. 32-bit versions of an RSA and KSA adder, an ALU and a multiplier were created using this methodology as well. The PTL-CNT versions of these modules were further analyzed and restoring logic was inserted periodically for RSA-based designs and optimally, depending on the varying output capacitance, for KSA and the sparse-tree adder. Each of these building blocks were then evaluated at varying voltages for delay and energy. We compare PTL-CNT results against both CCNT designs as well as Si-CMOS results. Based on both delay and energy numbers, a hybrid design of V-scale was made using PTL-CNT and CCNT modules. We maintain performance and reduce area by using PTL-CNT modules for components along the critical paths of the V-scale pipeline, while using low-energy CCNT modules for the rest of the chip.

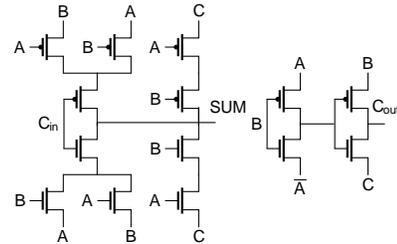


Fig. 4. Pass transistor-based full adder

Figure 5 compares the effect of voltage scaling on the three full adder designs. The results show that although the delay trends are similar, our PTL-CNT design clearly dominates in terms of energy, leading to a 7-19EDP reduction over Si-CMOS in the supply voltage range of 0.7-0.4 V.

2) 32-bit Adder and ALU: We implemented an RCA, whose results are shown in Figure 6a and Table I. In addition, results for the KSA are shown in Figure 6b and Table II. Our analysis shows that the implementation of a 32-bit RCA using the full adder in PTL-CNT entails a high EDP reduction over the CCNT and Si-CMOS implementations. Although some of the gains seen in the full adder are consumed by the addition of restoring logic placed for PTL. The PTL-CNT KSA implementation saw a smaller improvement in EDP compared to Si-CMOS. This occurred because the KSA required significantly more restoring logic than the RCA, more than offsetting the gains obtained in delay.

## VI. EVALUATION

In this section, we evaluate each of our core components implemented in Si-CMOS, CCNT, and PTL-CNT. We then evaluate the overall performance of the V-scale pipeline implemented with Si-CMOS, CCNT, and hybrid CCNT/PTL-CNT.

### A. Adder Analysis

We begin our analysis by studying a single full adder cell, then build both an RCA and KSA adder. Finally, we analyze an ALU design that leverages a hybrid of RCA and KSA.

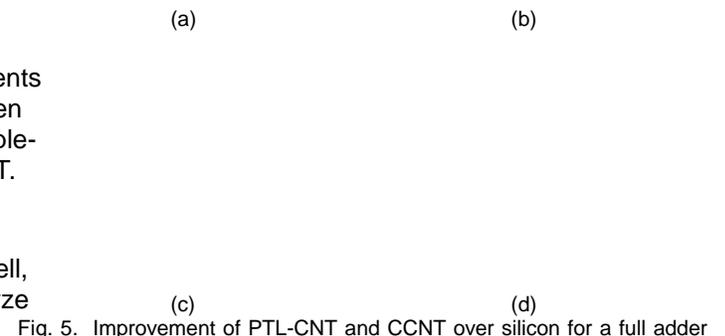


Fig. 5. Improvement of PTL-CNT and CCNT over silicon for a full adder

Fig. 6. Improvement of PTL-CNT and CCNT over silicon for (a) ripple-carry adder, (b) Kogge-Stone adder and (c) V-scale ALU

TABLE I  
RIPPLE-CARRY ADDER DESIGN RESULTS

Volt. (V)	Delay (ns)			Energy (fJ)		
	PTL-CNT	CCNT	Si-CMOS	PTL-CNT	CCNT	Si-CMOS
0.4	1:9	2:9	3:1	2:4	6:5	14:4
0.5	1:2	2:0	1:4	4:2	10:5	23:3
0.6	1:0	1:6	0:9	8:0	16:2	34:5
0.7	1:0	1:4	0:7	22:0	24:4	48:3

TABLE II  
KOGGE-STONE ADDER DESIGN RESULTS

Volt. (V)	Delay (ns)			Energy (fJ)		
	PTL-CNT	CCNT	Si-CMOS	PTL-CNT	CCNT	Si-CMOS
0.4	1:0	0:8	1:0	4:9	5:1	9:8
0.5	0:4	0:6	0:4	7:7	8:2	15:8
0.6	0:3	0:4	0:3	11:6	12:6	23:4
0.7	0:2	0:4	0:2	16:7	18:9	32:8

TABLE III  
V-SCALE ALU RESULTS

Volt. (V)	Delay (ns)			Energy (fJ)		
	PTL-CNT Hybrid	CCNT	Si-CMOS	PTL-CNT Hybrid	CCNT	Si-CMOS
0.4	2:1	3:2	3:5	20:5	25:4	43:5
0.5	1:2	2:2	1:6	38:3	44:4	72:7
0.6	1:0	1:8	1:0	73:4	79:1	109:6
0.7	0:9	1:5	0:7	127:4	118:5	156:5

TABLE IV  
ARRAY MULTIPLIER RESULTS

Volt. (V)	Delay (ns)			Energy (fJ)		
	PTL-CNT	CCNT	Si-CMOS	PTL-CNT	CCNT	Si-CMOS
0.4	3:7	2:2	2:8	276:2	293:4	560:6
0.5	1:9	1:5	1:2	429:1	470:6	906:9
0.6	1:4	1:2	0:7	610:0	728:1	1351:4
0.7	1:1	1:1	0:5	930:7	1071:9	1902:7

For the ALU design, we used Synopsys Design Compiler implementations. Though Si-CMOS performs better than CCNT to generate a synthesized netlist. The result, a sparse-tree adder, drops by a small margin at higher voltages, the CCNT implementation borrows elements from both KSA and RCA. We implemented a similar sparse-tree adder for our V-scale ALU implementation in Figure 8. Figure 6c and Table III present the results of the ALU design. We find that the PTL-CNT ALU clearly outperforms the Si-CMOS ALU with an EDP reduction of 2.1 at 0.4 V.

## B. Multiplier

Results for the multiplier design are presented in Figure 7 and Table IV. We find a similar trend at higher voltages. The PTL-CNT multiplier has an EDP gain of 1.6 at 0.4 V, which is less than the 2 of the CCNT multiplier, due to the large overhead of restoring buffer insertion in the PTL-CNT design. Hence, we choose a CCNT-based multiplier for our pipeline.

## C. Registers

Since a D-flip flop mostly consists of inverters and transmission gates, we only build Si-CMOS and CCNT-based implementations.

Fig. 7. Improvement of PTL-CNT and CCNT over silicon for the multiplier

Figure 9 and Table V present the results of our full RISC-V pipeline design. We find that the V-scale core built using CCNT shows a 1.0-2.9 improvement in EDP over a Si-CMOS based core for a supply voltage range of 0.7-0.4 V. To improve this further, we analyzed the critical path and found that the ALU and parts of the multiplier were on the critical path. For that reason, we constructed a V-scale pipeline with the PTL-CNT versions of the ALU components. We obtained a 2-5 reduction of EDP over Si-CMOS with this implementation, which is also a 1.7-2 improvement over the entirely CCNT design. The results clearly show that CNTFETs are a better fit for low voltage and energy-efficient designs, and that judicious use of PTL can greatly improve the effectiveness of CNTs. While the individual components show on average a 2 improvement in EDP, the overall CPU pipeline shows a 5 improvement. This happens because the analysis for individual components were done at the maximum frequency for those components. When integrated into the entire pipeline, the critical path is comparatively longer than the propagation time of each individual component on it, and hence those units only contribute leakage power to the system's power for rest of the clock cycle. Since Si-CMOS has a larger penalty for leakage than CNTFETs, this compounds to produce the 5

