Teaching Students Computer Architecture for New, Nanotechnologies

Michael Thaddeus Niemier
University of Notre Dame
Dept. of Comp. Sci. and Eng.
Notre Dame, IN 46545
mniemier@nd.edu

Peter M. Kogge
University of Notre Dame
Dept. of Comp. Sci. and Eng.
Notre Dame, IN 46545
kogge@wizard.cse.nd.edu

Abstract:

Given the potential limitations facing CMOS, there has been an influx of work and research in various nano-scale devices. Most of the work related to nanotechnology has been done strictly with devices, with little attention given to circuits or architectures of them – the desired end result. In the past, these studies have usually lagged device development by many years. However, we propose a curriculum to help integrate the communities – device physicists and computer architects – earlier. One goal of such a curriculum would be to teach students how to generate a "Mead/Conway" methodology for a given nanotechnology. This would teach students not only how to help technology change and evolve, but eventually teach students how to adapt to changes after a technology evolution. Another goal would be to facilitate more (and earlier) interaction between device physicists and computer architects to prevent these two groups from developing diverging views of what is physically and computationally possible in a system of nano-scale devices.

1. Introduction:

Consider the following "quote" from the preface of a future book on nano-scale design:

"Until recently the design of integrated circuitry for nano-scale devices has been the province of circuit and logic designers working within nanotechnology firm research laboratories and select "pockets" of academia. Computer architects have traditionally composed systems from standard self-assembled nano-circuits designed and manufactured by these entities but have seldom participated in the specification and design of these circuits. Nano-engineering and Computer Science (NE/CS) curricula reflect this tradition with courses in nano-scale device physics and integrated circuit design (if any at all) aimed at a different group of students than those interested in digital system architecture and computer science. This text is written to fill a current gap in the literature and to introduce all NE/CS students to integrated system architecture and design for emerging nano-technologies. Combined with individual study in related research areas and participation in large system design projects, this text provides the basis for a course-sequence in integrated nano-systems." (Mead/Conway v)

With the potential physical and economic limitations facing CMOS, there has been a recent proliferation in research related to nano-scale devices – particularly those targeted toward computational systems. Much of this early work has been relegated to the development of the physical devices themselves; and while circuits and systems have probably been envisioned within each specific nanotechnology being considered, their development has usually not progressed beyond the conceptual stage. Furthermore, historically, computer architects have been disjoint from the process of actual circuit designs, and in the case of CMOS, comprehensive and integrated architectural and circuit design methodologies were not published until the late 1970s when Carver Mead and Lynn Conway's groundbreaking work appeared [1].

Interestingly, the above paragraph of *this* work is essentially verbatim from the preface of Mead and Conway's VLSI text. While written almost 25 years ago, it illustrates a problem that they faced – computer architects, who might be the "lowest common denominator" in designing a system to perform useful and efficient computation, did not take part in the development of the devices and basic circuits with which they were required to design. We are beginning to face this same problem now with regard to nano-scale devices, and this paper will propose the beginnings of a curriculum to help alleviate it.

At a recent NSF sponsored workshop on molecular scale devices and architectures [2], Lynn Conway reiterated that during the early years of CMOS development, while architects would sometimes work with MOS technologists, as a "group", most individuals did not span the whole range of knowledge required to design a complete computer system. Likewise, the scope required to do complex designs is large and it is not completely feasible for a device physicist to understand all of the issues a computer architect must consider. In the pre-Mead/Conway era, the development flow was for system architects to express a design at a high level, such as Boolean equations, and then turn it over to logic designers who converted the designs into "netlists" of basic circuits. Fab experts would then lay out implementations of the individual

logic blocks, and "just wire them together." Interaction between the architects and fab experts was limited. In terms of technology, MOS FETS were considered "slow and sloppy," and real design was in sophisticated bipolar devices.

The invention of the self-aligning FET gate allowed Mead and Conway to bridge this gap by changing the focus of fab from considering chips "in cross section" to an "overhead view" where it is the interconnect that is most visible. They did this by developing a set of design rules and abstractions that a computer architect could use to involve himself or herself in the circuit design process. They reduced the physics-dependent device descriptions to a scale-independent set of parameters based largely on area and shape, with some simple rules for first order modeling of how such devices would interact in combination with each other. They also introduced some simple but useful circuit primitives that changed the discussion from isolated logic gate performance to interconnect. This allows architects, who are experts in hierarchical designs, to extend their hierarchies one level down – to potentially new basic structures, and then take advantage of these structures in implementing larger and larger systems. The introduction and use of clever circuits using pass transistors is just one example of such an insight.

When coupled with the ability to cheaply fabricate real chips through MOSIS, this revolutionized the academic computer architecture community. Now, inexpensive, but adventuresome, prototyping could be carried on in an academic setting, by students (and faculty) whose growing expertise was in expressing and analyzing novel regular and hierarchical designs.

Before proposing any new and targeted curriculum for nanotechnologies, we will first revisit the existing core of the computer architecture curriculum at the University of Notre Dame – a representative subset of courses that would be taken by a student wishing to specialize in computer architecture. Also, because we propose that in the future there should be greater integration between communities of computer engineers/architects and those actually working on nano device development, we will include an overlay of relevant electrical engineering curriculum – especially that which is targeted toward electrical engineers interested in computer systems. This will be used to show how electrical and computer engineering curricula currently interact and will help define a base for an integrated curriculum targeted toward nano-scale architectures.

Fig. 1 illustrates the existing curriculum. It also includes a listing of goals and topics relevant to each course, shows any overlap between the two curricula, documents popular course sequences, and highlights available course sequences. By examining this figure one can clearly see that all of the pieces are in place to facilitate interaction and understanding between electrical and computer engineers (or device physicists and architects!). A set curriculum is already in place for electrical engineers who have an interest in computer systems and several course sequences are available for computer engineers interested in the "physics" of logic. (Note: an interesting side project might be to integrate this "roadmap" into the first course, Logic Design (CSE 221), of this sequence to help students see and understand the "bigger picture" earlier.)

At the same workshop mentioned above, when speaking of nano-scale devices, Conway also posed the question of when will there be some emerging areas where designers will be able to compile enough basic information to start generating interesting circuits. At the University of Notre Dame, we believe that one promising "emerging area" is the Quantum-dot Cellular Automata (QCA). QCA stores information within "cells" consisting of multiple quantum dots via the positions of single electrons, and performs logic functions not by electron flow, but by Coulombic interactions between electrons in neighboring QCA cells. Real QCA cells have been fabricated by Notre Dame device physicists that demonstrate the key properties of computation, information transfer, and storage. Also, researchers are on the verge of creating QCA cells consisting of single molecules which may be "self-assembled" into larger structures via attachment to DNA tilings. Truly, QCA is in the nano-scale realm and a subset of actual devices – both theoretical and experimentally proven – exists.

Prior to the beginning of the authors' research on design with QCA, little work had been done in considering systems of, circuits for, or an architecture for QCA devices. Ironically (and rather unintentionally), our initial work mimicked the experiences of Mead and Conway in more ways than one. First, our interactions with technologists were not as successful as they could have been – because "as a group, most individuals did not span the range of knowledge required to design a complete computer system." As a particular example, recently we discovered that a QCA circuit characteristic that we (as architects) deemed essential for useful and efficient circuits, was not a priority for device physicists. Clearly, this illustrates the need for better communication and understanding between the two communities. Second, when examining our design process, it has by in large mirrored the path proposed by Mead and Conway to help circuit designers understand the architectural possibilities of a technology.

Now, with many other nanotechnologies consisting of at least a subset of experimental devices, we propose

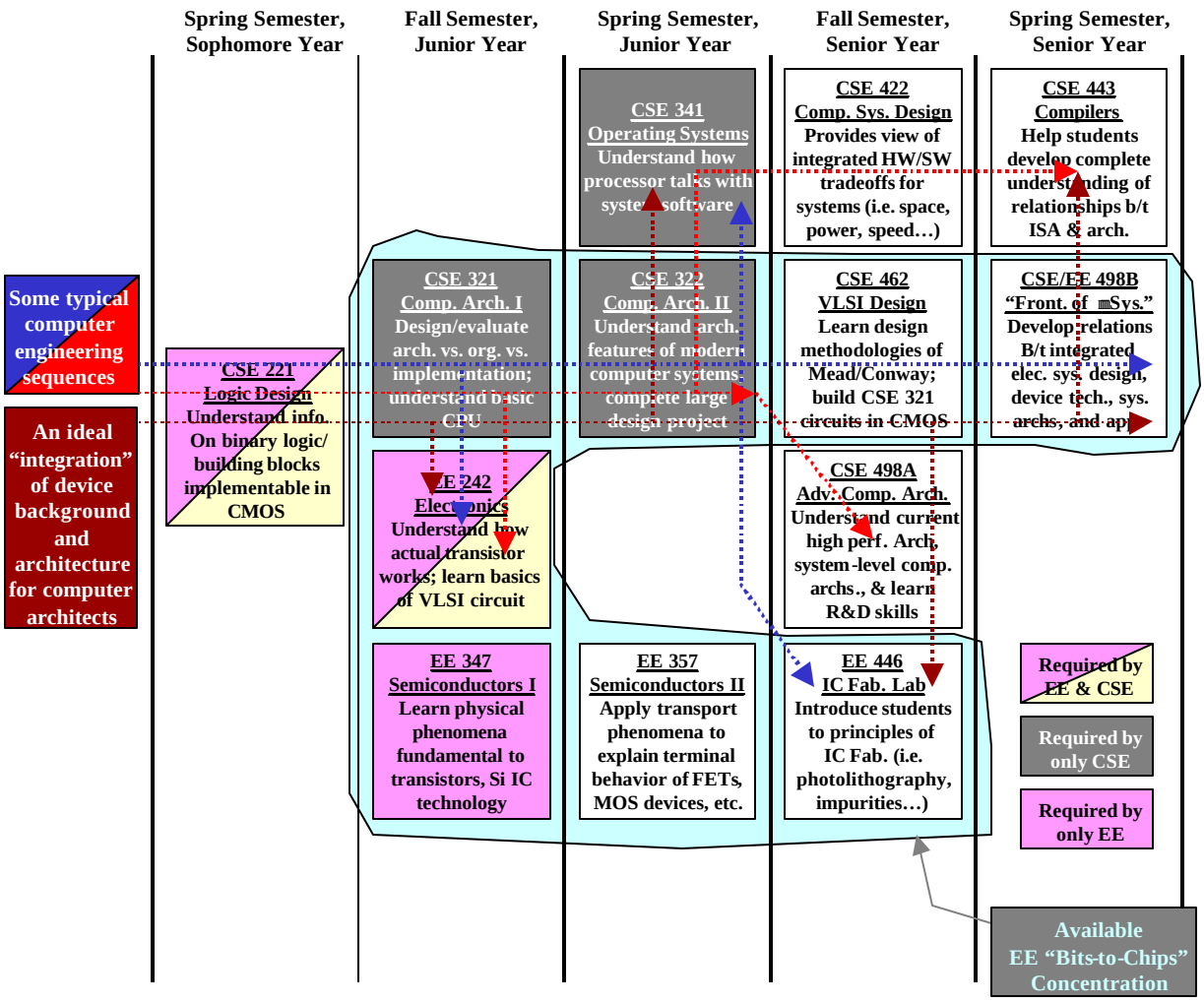

Fig. 1: Existing "core" of "conventional" computer architecture curriculum.

developing a curriculum to teach students how to develop a set of guidelines for computer architects and circuit designers for a specific nanotechnology. The context will include our experiences with QCA and the proven methodologies proposed by Mead and Conway for one of the most commercially successful computational mediums -- CMOS. Eventually an end result might *be* a "Mead/Conway" study for a specific nanotechnology. However, another (and earlier) goal of the curriculum is to teach students *how* to actually develop a "Mead/Conway" study for *any* nanotechnology. We also propose an extension of their work – namely, preparing computer architects and circuit designers to work with device physicists during actual device development. The end result envisioned is as a group, individuals who span the range of knowledge required to design better devices and complete computer systems.

With these thoughts in mind, Fig. 1 has been augmented in Fig. 2 to show a parallel curriculum that will end with a "Frontiers of Nano-Systems course" and accomplish one of the first goals stated above – namely educate students on *how* to develop a "Mead/Conway" for any nanotechnology. Interestingly, the second goal (preparing computer architects and circuit designers to work with device physicists during actual device development) should be accomplished by the course sequence itself as a.) it (like a VLSI or logic design course) would be targeted toward both electrical and computer engineers and b.) "the big picture" detailed in the figure below will be explained to students at the beginning of the sequence and act as a roadmap to help the students understand what they are working toward. Finally, Fig. 2 illustrates an approximate time sequence as to where these courses would fit into existing electrical and computer engineering curriculum. They could easily occur simultaneously with or after an appropriate course in "conventional" electronics and architectures. However, they could also be taught *before* the similar "conventional" course. This is based

on the idea that someone who is trying to develop an architecture for a specific nanotechnology might have better success with *less* knowledge of previous design evolutions and/or design methodologies. Would a potential computer architect be better off with just a sound basis of knowledge in the nanotechnology that he or she is trying to develop a "Mead/Conway" for? Would this lead to the best possible design methodology and architecture for *that* particular nano-scale device? Arguments will be made for both cases based on our experiences with QCA.

The rest of this paper will discuss the "CMOS independent" parts of our current curriculum, and what needs to be kept intact from it – largely the hierarchical design approach. We will also detail how we propose to educate students to accomplish the above goals. We will first discuss our proposed curriculum in detail and discuss what background students should bring to it and learn from it. The next section will discuss why we should – and how to – encourage students to think "outside the box" with regard to circuits and architectures for nanotechnologies. Next, we will consider mechanisms, examples, etc. for introducing students to the actual development of circuit design rules, techniques, and architectures. Finally, we will conclude and discuss future work. Interestingly, each of these sections will be introduced with an excerpt from the text of the Mead/Conway preface indicative of the fact that architects studying nanotechnology will have to face and solve many of the same problems that were first experienced during the last technology evolution.

2. (Student) Background:

"We have chosen to provide and assume that students will bring with them just enough essential

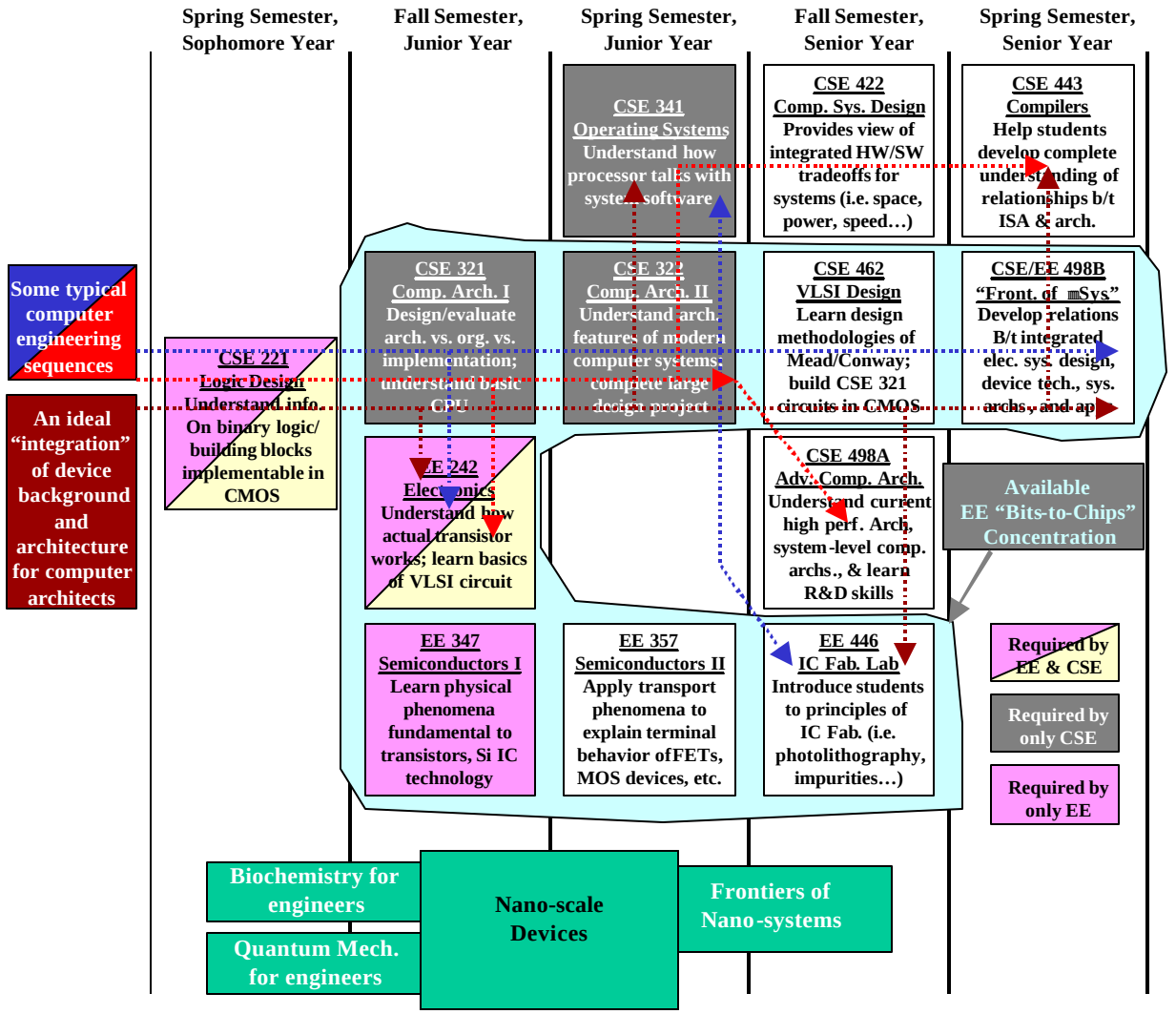

Fig. 2: Existing "core" of computer architecture curriculum augmented with proposed "nano"-curriculum.

information about devices, circuits, fabrication technology, logic design techniques, and system architecture to enable them to fully span the entire range of abstractions from the underlying physics to complete VLSI digital computer systems."
(Mead/Conway vi)

As stated in the introduction, an initial end goal of our curriculum is to teach students how to design a Mead/Conway study for any nanotechnology. The above excerpt from the actual Mead/Conway preface describes what knowledge the authors expected students (including computer architects!) to have in order to understand the design rules provided for VLSI systems. While the existing and "conventional" computer architecture course sequences will provide some needed background for a concentration in nano-scale design, clearly, preparing students for a technological evolution will require additional and different fundamental information as well.

It should be reemphasized that Mead and Conway were proposing a "capstone" class in VLSI design, while we are proposing a curriculum to teach the development of their methodologies as an end goal (which will hopefully, eventually lead to an analogous "capstone" course for a specific nanotechnology). Consequently, we must also define what background – devices, logic design methods, fabrication techniques, etc. – students will need to meet this goal. This "background" must be provided in two different ways. First, an entirely new subset of courses must be developed to teach students the fundamentals of *nano-scale* devices and *nano-scale* fabrication techniques. What should such a sequence entail? This question can best be answered by looking at the different disciplines that are part of various nano-scale device developments. For example, in addition to electrical engineers, physicists, and computer architects, *chemists* are an integral part of the development of QCA. Additionally, other emerging nanotechnologies – DNA-based computing, carbon nanotubes, etc. – all have roots in chemistry. With this in mind we believe that any curriculum designed to teach students how to develop systems of nano-scale devices should include a course in biochemistry – but targeted toward engineers.

Other background information can most likely be derived from existing courses, albeit retargeted for different ends. For example, many emerging nanotechnologies are also rooted in quantum mechanics – Q-bits, QCA, etc. – and at the University of Notre Dame a course in quantum mechanics is available as part of the electrical engineering graduate curriculum (and available to interested undergraduates as well). Part of this existing course could easily be augmented/spun-off and should be targeted toward engineering students who are interested in circuit and system design.

Together, these two courses – biochemistry for engineers and quantum mechanics for engineers – would provide the foundation for a course in nano-scale devices which would eventually segway into a course intended to teach the development of Mead/Conway-esq design rules and methodologies. This specific course sequence is highlighted in Fig. 3 and each course is paired with its "conventional equivalent". By examining Fig. 3, one can conclude that the sequence of biochemistry for engineers and quantum mechanics for engineers would provide the same functionality for students desiring to study systems of nano-scale devices that the electrical engineering semiconductors course currently provides for students desiring to study systems of MOS devices. Namely, both teach students about the materials from which computational devices and their substrates can be built.

In the existing curriculum at the University of Notre Dame a course in electronics, which teaches students how computational devices constructed with various semiconductors actually function, occurs in parallel with the semiconductors course. Our proposed and parallel course in nano-scale devices fills the same role as a course in MOS electronics but occurs only after students have studied the fundamentals of how various *nano-scale* devices can actually be constructed. We believe that sequencing these course sets will provide engineering students with the greatest level of understanding about the computational devices.

Our course sequence concludes with a "Frontiers of Nano-Systems" course. The particular class is currently "paired" with the existing VLSI course (which employs and teaches the Mead/Conway design rules and methodologies for MOS) as well as the Frontiers of Microsystems course (which seeks to help students understand the relationships between integrated circuit design, device technology, system architecture, and applications for MOS devices) [3]. However, because there are many promising nano-scale devices and no heir-apparent to CMOS, our proposed "Frontiers of Nano-Systems" currently exists essentially as a combination of its two MOS equivalents. While it might involve case studies of architectures and design rules for existing and promising computational devices, it is more targeted toward helping students understand how such design rules were actually developed. Essentially, the goal of this course is to teach students how to help technology *evolve*.

Ideally, work completed and skills learned in a Frontiers of Nano-Systems course will someday lead to a specific Mead/Conway-esq course for a specific nanotechnology. Such a course might be offered when a nanotechnology has evolved enough that a MOSIS-like conglomerate exists for it. For MOS devices,

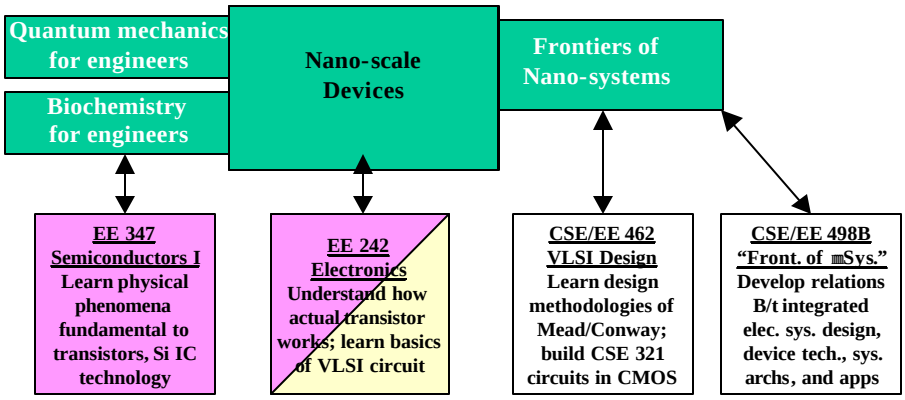

Fig. 3: The core of the "nano"-curriculum with "conventional" curriculum equivalents.

MOSIS (Metal Oxide Semiconductor Implementation Service) provides system designers with a single interface to the constantly changing technologies of the semiconductor industry and allows for the fabrication of their circuits. Were an "NIS" ("Nanotechnology Implementation Service) to exist, a set of design rules (or single interface) for a specific nanotechnology would also exist. It is these design rules that would form the core of a course that would **not** teach students how to help technology *evolve*. Instead, such a course would not only allow computer architects to prototype and analyze novel and regular devices for a nanotechnology, it would also help a community *adapt* to a new computational medium. Essentially Frontiers of Nano-Systems would become two courses – one to teach students how to adapt, the other to teach students how to keep evolving. (Also, even those who do not participate in an eventual "NIS-targeted" course will have at least seen and experienced what is required to *adapt* to a new technology).

Finally, there are three important generalizations to make about our proposed curriculum for designing with nano-scale devices. First, when examining its "conventional equivalent" one can see that it consists of a mix of electrical engineering and computer engineering courses – specifically one electrical engineering elective, one electrical engineering and computer engineering requirement, and one computer engineering elective. Note that it contains no explicit or existing computer architecture courses (more on this next). However, it does contain a significant "deviation" from the "conventional" curriculum. Namely, previously, a semiconductors course was not a requirement or even a common elective for computer engineers (i.e. computer architects). However, because we want to facilitate closer interactions between electrical and computer engineers (device physicists and computer architects) who are trying to develop nano-scale devices, we believe a semiconductors-like course should be part of the core curriculum. Here this takes the form of biochemistry and quantum mechanics for engineers which will help ensure that computer architects understand the limits and constraints of what can be built, constructed, or designed with a specific nano-scale device.

Second, as mentioned above, there are no explicit logic design or computer architecture courses that are part of our proposed curriculum. New or retargeted courses are not proposed because in order to understand a simple CPU or build simple computational logic circuitry students still must learn basic logic design techniques and hierarchical design methodologies that "conventional" classes like logic design and computer architecture provide. Now, if a semiconductors-like background should be a requirement for any computer architect working on developing nano-scale devices, then similarly a background in logic design/computer architecture would be ideal for device physicists. While a bits-to-chips sequence for electrical engineers is highlighted in Fig. 1, the front-end of that sequence – logic design and computer architecture – is most essential for cementing a close working relationship.

Third, and finally, in the introduction we posed the question of whether it would be best for a student to take part in this curriculum with either a through or a minimal background in logic design, device physics, and principle of VLSI design methodologies. Until now, we have left our proposed course sequences vague with regard to where they would fit into an academic timeline. One could make the argument that it would be best to teach students how to design for a nano-scale device before little or any of the "conventional curriculum" is taught (where "conventional curriculum" refers to MOS equivalent courses as well as courses in computer architecture or VLSI design). This way a student would have no preconceived notions of what a circuit or system must look like or has looked like and might develop the best set of system design rules for a particular nano-scale device. However, an argument against this approach would obviously be that a student would have little if any knowledge about basic design or even how a simple CPU works, severely limiting

what he or she might design. One could also argue that it would be best to prepare students for a technology change only after they have experienced all of the "conventional curriculum". Then, they would will have not only learned basic principles of logic and CPU design, but also will have learned advanced architecture techniques and studied design rules and methodologies for a proven computational medium – CMOS. However, this approach does not separate the process of technology from the process of design and may cloud students' thinking by teaching them one way to design and study large systems of integrated circuits. Would *this* result in the best, original set of design rules for a particular nano-scale device?

A better answer might actually be a mix of the two arguments. A nano-engineering course in quantum mechanics and/or biochemistry should take place concurrently with a "conventional" semiconductors or electronics class. This way, students will learn the fundamentals of each technology in parallel and will be less inclined to "think" in terms of one technology over another. Similarly, a nano-scale device course should take place concurrently with a computer architecture course sequence and after a "conventional" electronics class. This will allow students to consider how basic CPU requirements and hierarchical design methodologies learned in computer architecture might apply to nano-scale devices. Also, the "conventional" electronics course will provide a student with a good foundation of what a computational device **has** to do, but not necessarily **how** it must do it. Finally, the Frontiers of Nano-Systems class could take place in conjunction with a "conventional" VLSI class (so students thinking is left "unclouded") or after it (for a better foundation in what designing a Mead/Conway set of design rules is all about). However, we would suggest that students take it before some of the more advanced computer architecture classes. Why? Students will have a generic idea of what a CPU must do but will not be tied to more complex architectural techniques – hopefully leading to a set of original, targeted, and unclouded set of design rules for a particular nano-scale device. Additionally, one could always take advanced architecture courses later and apply techniques learned in them to an existing nano-scale system.

3. Out of the Box:

"VLSI electronics presents a challenge, not only to those involved in the development of fabrication technology, but also to computer scientists and computer architects. The ways in which digital systems are structured, the procedures used to design them, the trade-offs between hardware and software, and the design of computational algorithms will all be greatly affected by the coming changes in integrated electronics." (Mead/Conway v)

This Mead/Conway excerpt essentially describes what biochemistry for engineers, quantum mechanics for engineers, and nano-scale devices must teach students to do in our new and parallel curriculum. Obviously a major purpose of these classes and the case studies that will be analyzed in them will be to help students learn "the basics" of the promising nanotechnologies and initialize a close working relationship between device physicists and computer architects. This relationship is critical to prevent these two groups/entities from developing diverging views of what is physically and computationally possible in a system of nano-scale devices. It is best illustrated and explained here (and eventually to students in a class) with a short case study from our experiences with QCA.

Earlier, we alluded to the fact that a QCA circuit characteristic that we (as architects) deemed essential for useful and efficient circuits was not a priority for device physicists. Specifically, an idealized QCA device (or cell) can be viewed as a set of four charge containers or "dots" positioned at the corners of a square. The cells contain two extra mobile electrons which can quantum mechanically tunnel between dots but, by design, cannot tunnel between cells. The configuration of charge within the cell is quantified by cell polarization, which can vary between P=-1, representing a binary "0", and P=+1, representing a binary "1". Unlike CMOS (in which multiple layers of metal can facilitate data routing), there really is no "third dimension" in which to route wire in QCA. However, a wire formed by QCA cells rotated by 45 degrees can cross a wire formed by 90-degree (unrotated) QCA cells in the plane with no interference of either value on either wire. Early in our architectural/circuit design study of QCA, this property was considered to be of the utmost importance as it provided our only other "dimension" of routing. However, when discussing our designs with chemists (who are working on DNA substrates on which QCA molecules could be attached) we realized that they had not yet even considered the interaction of 45-degree cells with 90-degree cells (as for them, this was a very complex design problem). This early collaboration has resulted in some relatively minor changes in the way our circuit and system designs will be structured and has led the device physicists and chemists to reconsider this problem. The result should be a more feasible design with potential for earlier implementation.

Now, we also mentioned in the previous section that the courses in our sequence discussed above would and should take place in parallel with "conventional" logic design and computer architecture curriculum. This should allow and facilitate student thinking about

how the fundamental computational and CPU requirements detailed in these courses could best be mapped to systems of nano-scale devices. This brings us to the second purpose of this course sequence and one that was alluded to when detailing the nano-scale devices course. Namely, by now students will have realized that computational devices *have* to do certain things. However, with nanotechnology, how they do them is very much "up in the air". Students must be taught to embrace this and how to think outside of the box. Again, this is best presented with a short case study.

An important feature of MOS electronics is a pass transistor that essentially allows current (i.e. binary information) to flow between *a* and *b* in either direction. However, in QCA information is not moved by electron flow but rather by Coulombic interaction between electrons in quantum dots. Because nearness between QCA cells is required to move information from *a* to *b* there is no obvious way to create the equivalent of a pass transistor (either bi- or uni-directional) using only QCA devices. (For example, this would make generating the equivalent of a switching matrix – i.e. for a simple FPGA – in QCA much more difficult – although not impossible). Also, unlike the standard CMOS clock, the QCA clock is not a signal with a high or low phase. Rather, the clock changes phases when potential barriers that affect a group of QCA cells (a clocking zone) pass through four clock phases: *switch* (unpolarized QCA cells are driven by some input and change state), *hold* (QCA cells are held in some definite polarization -- i.e. some binary state), *release* (QCA cells lose their polarization), and *relax* (QCA cells remain unpolarized). One clock cycle occurs when a given clocking zone has cycled through all four clock phases. To understand how the equivalent of at least a uni-directional QCA pass transistor or switch might be implemented, its worthwhile to consider the exact purpose of the *relax* clock phase. Without it, QCA cells in the *switch* phase could be driven from two different directions (i.e. from cells with a definite polarization in the adjacent *hold* phase and cells with an initial polarization in the adjacent *release* phase). The *relax* phase acts as a buffer to ensure that this does not occur. Thus, the *relax* phase has the effect of "removing" a group of QCA cells from a given design. Using this idea, routing could be accomplished by using the clock to selectively "turn off" groups of QCA cells to create switches.

The timeline of this integrated, "conventional" curriculum and "nano" curriculum is ideal because students will have acquired some knowledge about the fundamental requirements for a CPU and logic as well as what devices are commonly used to implement them in their "conventional" courses. However, simultaneously, courses such as nano-scale devices will teach students what is and what is not physically possible in the "nano"-realm. One lesson might show how some functionality and logic will certainly map from a standard technology to an evolved technology (i.e. CMOS → QCA). However, another lesson might best be summarized as follows: "You understand device *X*, you've used *X* a lot, well, now *X* is no longer physically possible and you'll need to find a new way to either recreate its functionality or a completely different way to do task *Y*."

4. Frontiers:

"In any given technology, form follows function in a particular way. The most efficient first step towards understanding the architectural possibilities of a technology is the study of carefully selected existing designs. However, system architecture and design, like any art, can only be learned by doing. Carrying a small design from conception through to successful completion provides the confidence necessary to undertake larger designs." (Mead/Conway vii)

The above quotation from the Mead/Conway preface actually describes both courses which could eventually result from the sequence biochemistry/quantum mechanics for engineers and nano-scale devices. In the nearer term, a Frontiers of Nano-Systems course will teach students how to develop a set of design rules and system architecture using the methods described in the above excerpt. Explaining how this will be done will best be accomplished (and illustrated) via a series of case studies and comparisons between them.

For example, let's revisit our work with QCA. Prior to our research, little work had been done in considering systems of, circuits for, or an architecture for QCA devices. Consequently, as with other technologies that preceded it, and like Mead and Conway proposed above, initial studies of QCA started off by designing basic circuit elements that would be needed for a processor. Next, it was determined that a simple microprocessor should be constructed QCA cell-by-QCA cell (essentially in the same manner in which many of the early Intel microprocessors were designed). The processor of choice was simple enough to be designed by hand, yet it still contained the basic elements that are part of any microprocessor (i.e. arithmetic and logic units, registers, latches, etc.). Hence, solutions to the difficulties encountered and overcome in this design would be applicable to even more complex systems and processors. Problems encountered during this design process were largely related to floorplanning – which in turn arose from the interdependence of layout and timing with QCA. As we saw above, the nature of the QCA "clock" leads to

an inherent self-latching of the QCA device. Given this constraint, and before making any further attempts at a large scale design, we felt the need to develop methods to successfully factor the constraints generated by the inherent self-latching of QCA out of the "equation" of a design and furthermore find a means to exploit it. Thus, an extensive study of floorplanning was conducted and several viable floorplans for QCA circuits were developed. After the floorplanning study was conducted, a complete layout of the dataflow for our microprocessor was finished. During this design process, register designs, feedback mechanisms, interconnect problems, etc. were developed and/or identified. Design rules were compiled and formed the engine of a simulator written to test circuits for logical correctness. These design tools were then used to simulate and reanalyze existing design schematics.

Work then proceeded to studying control flow. Interesting results from this work include the lack of a need for an explicit flip-flop to hold a bit of state information in a QCA state machine (the inherent latching in wire stores the bit), more intelligent floorplans to ensure that QCA cells representing bits of state actually change clock phases and polarizations at the proper time, an algorithm for intelligent state placement, and a one-hot state machine that could properly control a QCA dataflow and yet not maintain the "classical" properties of a "true" one-hot (i.e. all bits of state switch at a time *relative* to a set of inputs that determine state). While physically unrealizable in the short-term, when this work is finished the first complete QCA microprocessor will have been designed. Most importantly, this effort will provide the first real insight into how an architecture for a (self-latching) nanotechnology should be organized. Furthermore, as discussed in the third section of this paper, work with hand-crafted designs resulted in the opportunities to review them and collaborate with device physicists which in turn led to a more physically realizable near-term implementation target.

A next logical step will be to examine similar design rule evolutions and compare and contrast them – particularly determining and teaching the characteristics and needs for common threads between existing "Mead/Conway"s (i.e. floorplanning). Finally, as mentioned in the second section of this paper, when an NIS conglomerate exists for a specific technology, this class can itself evolve into a course that specifically teaches that set of system design rules – and helps students adapt to a new computational medium.

5. Wrap-up:

"The general availability of courses in VLSI system design at major universities marks the beginning of a new era in electronics. The rate of system innovation using this remarkable technology need no longer be limited by the perceptions of a handful of semiconductor companies and large computer manufacturers. New metaphors for computation, new design methodologies, and an abundance of new application areas are already arising within the universities, within many system firms, and within a multitude of new small enterprises. There many never have been a greater opportunity for free enterprise than that presented by these circumstances."

After changing "VLSI" to "nanotechnology" in the above Mead/Conway excerpt, nothing else need be said.

Acknowledgements:

The authors would like to emphasize the depth of insight we owe to Lynn Conway, whose comments at the MAW workshop and in email exchanges during the preparation of this paper were invaluable.

References:

[1] Carver Mead and Lynn Conway. *Introduction to VLSI Systems*. Addison-Wesley Publishing Company, Inc., Philippines, 1980.

[2] Molecular Architecture Workshop, Univ. of Notre Dame, Nov 12-13, 2001, www.cse.nd.edu/cse_proj/maw

[3] G.H. Bernstein, and J.B. Brockman, and G.L. Snider, and P.M. Kogge and B.E. Walvoord. "From Bits to Chips: A Multidisciplinary Curriculum for Microelectronics System Design Education", American Society for Engineering Education IL/IN Sectional Conference, April 12, 2002 – Illinois Institute of Technology, Chicago, IL 2002.