KEY ISSUES IN VLSI EDUCATION

Peter J. Hicks*

Barely ten years have passed since VLSI was first introduced into the syllabuses of Electronic Engineering and Computer Science courses following the pioneering work of Carver Mead and Lynne Conway in the late 1970s and early '80s. Since that time the technology has progressed to the point where chips containing millions of transistors are routinely produced and the silicon industry is confidently predicting that hundreds of millions of transistors on a single chip will be attainable by the end of the decade. It is worth examining the ways in which attitudes to VLSI education have developed over the past ten years and what further changes the future may have in store. The purpose of this paper is to try and identify some of the key issues that need to be addressed in designing syllabuses for VLSI design courses.

1. Introduction

In 1979 Mead and Conway published their seminal text entitled "An Introduction to VLSI Systems" (Mead and Conway, 1979) which aimed to demystify the art of integrated circuit design and open up the silicon medium to those who normally functioned at the algorithmic or system level - namely computer scientists and systems designers. The book immediately captured the imagination of academics around the globe and soon many universities and colleges were offering Mead and Conway style VLSI courses to their students. Over ten years have elapsed since those pioneering days in the early 1980s and it is worth pausing to look back at the way in which VLSI design methodologies have evolved during this period. In particular it is interesting to reflect on the approach that industry has adopted to the design of Application-Specific Integrated Circuits or ASICs - sometimes referred to as User-Specific Integrated Circuits (USICs). Only by doing this is it possible to identify the skills that students should be taught to prepare them for a career in industry. One can then proceed to construct syllabuses for the courses needed to teach VLSI design.

In carrying out such a review of the requirements for VLSI education a number of key questions arise, and the purpose of this paper is to consider some of these in more detail.

* Department of Electrical Engineering & Electronics, UMIST, PO Box 88, Manchester M60 1QD, UK

2. VLSI Design Methodologies

It is appropriate to question the way in which graduates will use their VLSI design skills because we have a responsibility to ensure that they are adequately prepared to embark on careers as engineers in industry. One can reasonably assume that students will use their knowledge of VLSI in one of the following ways:

(a) Develop new semiconductor technologies and processes for fabricating integrated circuits.

(b) Develop new cells for incorporation into a silicon vendor's cell library.

(c) Develop CAD tools for VLSI design.

(d) Develop integrated circuits in the shape of either catalogue parts or ASICs.

Clearly the products of (a), (b) and (c) are processes, cells and software tools for use by (d). In general there is a much greater demand for engineers in category (d) than for those in (a), (b) or (c), so perhaps this should be taken into consideration when designing the syllabuses for VLSI courses. This leads to the first of the issues to be discussed in this paper, namely:

"At what level of abstraction should VLSI design be taught?"

Bearing in mind that many academic institutions followed the Mead/Conway approach when constructing their VLSI courses, it is interesting to compare this with the approach adopted by industry.

The methodology proposed by (Mead and Conway, 1979) was based on mastering complexity through the use of regular arrays built up from a relatively small number of purpose-designed 'primitive cells'. As such it required the designer to design the primitive cells at the silicon level by laying out the transistor geometries on the various mask layers of the fabrication process - so-called "polygon-pushing". An important aspect of the Mead/Conway approach was the ability to abstract away from the lower-level details of silicon IC design, thus demystifying the whole concept of chip design and making it accessible to non-specialists in semiconductor physics and device design. Tailoring the primitive cells to suit the global routing requirements of each individual design made it possible to optimise chip layout in terms of silicon area.

Industry's approach to VLSI design, at least so far as ASICs or USICs are concerned, has tended to be quite different to the one described in the previous paragraph. Commercial VLSI design methodologies have standardised on gate-array or cell-based design styles in which design is performed at the logic gate or functional block level. Generally speaking, optimised layout efficiency and hence silicon area has been sacrificed in order to achieve shortened design time and "right first time" design. Devotees of the Mead/Conway approach criticise these design styles as being 'inelegant' and wasteful of silicon area, although it is difficult to argue against the principle of re-use which lies at the heart of library cell based techniques. Especially important from the designer's point of view is the fact that all the cells in a silicon vendor's cell library have been fully characterised and tested and can therefore be relied upon to work correctly every time. Raising the level of abstraction from transistors to logic gates precludes the need for designers engaging in time-consuming activities such as 'polygon-pushing' and circuit simulation. The availability of fully-automated cell placement and routing software also serves to accelerate design times and increase the probability of success.

It is probably true to say that the majority of companies employing electronics engineers as ASIC designers would prefer them to take a **systems** approach to design - that is, concentrate on getting the chip to function correctly as part of a complete system. Having said this, some transistor-level knowledge is important in order for the student to gain an appreciation of what is going on at the silicon level and understand how this can influence the performance of chips, for example. Furthermore, someone still has to possess the silicon-level expertise needed to design and characterise the cells which are to go into the silicon vendor's library. A place must therefore be reserved for this sort of knowledge in the VLSI curriculum, albeit perhaps at a fairly advanced level.

3. Technologies and Support Environments for VLSI Design

The next issue that arises is the question of whether it matters which technology is used for teaching VLSI design, or in other words:

"Should VLSI teaching be based primarily on CMOS technology?"

It is clearly important that students should understand the properties of the various semiconductor IC technologies and the trade-offs that result in terms of parameters such as switching speed, power consumption and packing density. From the point of view of a gate array or cell-based VLSI design methodology, however, it matters relatively little which technology the gates are cast in so far as teaching the principles of design are concerned. Certainly the student needs to be made aware of limitations imposed by things like gate delays, fan-out and power dissipation, but otherwise the technology is more or less hidden beneath the logic level of abstraction.

When it comes to teaching the principles of design at the silicon level the choice of technology is of far greater significance, although the ideas behind certain concepts such as layout design rules and the role of circuit simulation are essentially technology-independent. It has to be recognised that different skills are required to design circuits in a bipolar technology to those needed for CMOS, for example. On balance CMOS probably represents the most sensible choice at the moment because it is by far the most prevalent technology used for chip production. It also has the advantage that many of the lower-level details relating to MOS device behaviour can be abstracted out in the manner originally used by Mead and Conway. Bipolar IC design techniques tend not to have been widely taught in the past, the explanation perhaps being that it is not as easy to abstract away from low-level details as it is in the case of MOS design. Certainly there have been few texts published that would be suitable for teaching bipolar chip design as part of a VLSI course. Another important factor is that bipolar is still regarded as a 'niche' technology, its use being confined mainly to analogue and high-performance digital (ECL) applications. One way in which bipolar design could assume greater importance in the future is through the growing popularity of BiCMOS technology. This combines bipolar and CMOS technologies on the same chip and offers particular advantages to designers of mixed analogue/digital ICs.

Computer-Aided Design (CAD) tools are indispensable to the VLSI designer and it is therefore important that students should be aware of their capabilities - and limitations! However, some of today's VLSI design tools can be extremely complex and difficult to learn and this raises the question:

"Should students be trained to use advanced CAD tools?"

Although there can be a strong temptation to introduce students to sophisticated, state-of-the-art tools, the advanced features that such programs provide and their ability to cope with very complex designs are often not required in a teaching context. It may be preferable instead to use simplified tools with basic, easy-to-learn user interfaces that allow students to concentrate on learning fundamental principles, not tool-specific procedures. In the end the emphasis needs to be placed on education, not training.

4. The Role of Hardware Description Languages

The same complexity issues that motivated software engineers to develop higher order languages (HOLs) to support structured programming are now motivating hardware engineers towards the development of hardware description languages (HDLs) to support structured design.

An HDL must support both structural hierarchy and behavioural abstraction; this allows a design unit's behaviour and structure to be specified at any hierarchical level. The language should also provide architectural description capabilities which support hardware parallelism and concurrency.

Pure behaviour, architecture and pure structure can be thought of as a continuum. A purely structural description is a degenerate form of architectural description at one extreme of the continuum. Likewise, a purely behavioural description composed of a procedure is a degenerate form of architectural description at the opposite extreme of the continuum. In the case of a VLSI chip it is possible to view the design from a third perspective, namely the geometric representation. Gajski and Kuhn (1983) used a diagram to illustrate the relationship between the various hierarchical levels in the behavioural, structural and geometric views of a design - the Gajski-Kuhn 'Y' diagram (Fig. 1).

Behavioural design offers significant advantages over more common structural approaches. In describing how the designer wants a block to function, behavioural design is more flexible and efficient for designers and design automation tools. The ability to consider electronic system software and hardware jointly has tremendous potential in improving software/hardware performance trade-off analysis and system synthesis. The ability to design using behavioural descriptions (or preferably a mixture of behavioural and structural descriptions) should therefore be taught as an essential aim of any VLSI design course. The hardware description language most likely to be used for this purpose is VHDL which stands for VHSIC Hardware Description Language. (The VHSIC or Very High Speed Integrated Circuits programme was

```
  STRUCTURAL                     BEHAVIOURAL
REPRESENTATION                 REPRESENTATION
          \                         /
    Processor-Memory         System I/O
          Switch             Specification
             \                    /
      Register Transfer       Algorithmic
          Network                /
                  \         /
                 Gate  Boolean Expression
                      \ /
                       |
               Cell Geometrics
                       |
           Pattern Generation Data
                       |
                       |
                  GEOMETRIC
                REPRESENTATION
```

Fig. 1.

sponsored by the United States Department of Defence in the 1980s.) VHDL is emerging as the standard for hardware description languages and is beginning to find widespread use in industry. However, like the programming language Ada on which VHDL is based, the language itself is complex and not very easy to learn. This therefore raises the question:

"Should VHDL be taught as a Hardware Description Language"

One possible approach might be to define a subset of the full VHDL language which was simplified for educational use, assuming that this could be achieved without sacrificing too much of the power and flexibility of the language. Another alternative would be to use a conventional programming language such as Pascal to convey the basic principles of designing with HDLs, although it is recognised that this approach is not without problems.

5. The Value of Fabricating Student-Designed Chips

Most people involved in teaching VLSI courses would agree that there is no substitute for getting students to design their own chips. The value of such an exercise is greatly enhanced if the chip can be fabricated and returned to the student for testing and characterisation. Not only is the prospect of receiving their own chip a strong motivating factor to complete the design, it can also be used to reinforce considerations such as design-for-test, power dissipation, packaging and pin-out limitations and many other important issues. In the past there have been a number of major obstacles in the path of any attempt to get student-designed chips fabricated. The first is cost, although in recent years this has become much less of a problem because of the success of schemes such as EUROCHIP. The second problem has been, and still is, the length of time it takes to fabricate chips through a silicon foundry. Even now the average turnaround time for a typical CMOS process is little short of 12 weeks and it can be difficult to accommodate such a long delay into the timescales for the design exercises and projects that students engage in. So, in addressing the issue:

"Should student-designed chips be fabricated?"

The answer is a qualified "yes". If a way can be found of satisfactorily incorporating the entire design-fabricate-test cycle into the academic teaching calendar then there are many good reasons why this should be done. But if the chips arrive too late for the students to 'close the loop' by putting them on a tester then it is doubtful whether the exercise is worth the time, effort and expense involved.

A recent development that has proved to be very beneficial from the point of view of teaching ASIC design is the Field Programmable Gate Array or FPGA. Devices such as the Logic Cell Array from Xilinx or the ACT-1 arrays from Actel have to a large extent superseded mask-programmable gate arrays as a vehicle for the practical teaching of digital system design. The advantage of FPGAs over conventional gate arrays is that they permit students to design chips with up to several thousand gates with desk-top programmability. The turnaround time is therefore reduced from 12 weeks to a matter of minutes!

Custom or cell-based CMOS still has a role to play in the teaching of silicon-level cell design, analogue IC design and for research applications.

6. Quality and Reliability

Issues such as quality and reliability are seldom regarded as subjects of academic interest, although in the context of VLSI design they are obviously extremely important. Thus the tendency in the past has been to teach design methods and not necessarily design-for-quality. The issue here is therefore:

"Should quality and reliability feature in VLSI education?"

In Japan, for example, design-for-quality is often incorporated into courses through the teaching of approaches such as the Taguchi Method. This method introduces an extra step during design in which system component values and tolerances are systematically varied to produce a range of product behaviours. The designer then selects system component values that will make the expected performance of the overall system as close as feasible to the desired performance, with the smallest possible variance over the product's life.

7. An Outline for a VLSI Curriculum

Having considered some of the issues that are likely to influence the content of a course aimed at teaching VLSI design, it is appropriate at this point to examine the list of subjects that such a course might include. These are presented below and have been grouped into three levels; Foundation, Intermediate and Advanced.

Foundation Topics

Properties of materials
Semiconductor physics
Circuit theory
Electromagnetic theory
Software engineering
Logic design
Mathematics

Intermediate Topics

Semiconductor device models
Analogue & Digital circuit design
ASIC design methodologies
Hardware Description Languages
Transmission line theory

Advanced Topics

Semiconductor processing
VLSI cell design & layout
CAD tools and test methods
VLSI architectures - DSP etc.
Reliability, Yield, Quality....?

The list only includes topics which are seen to have some bearing on VLSI design and is not intended to be either definitive or exhaustive.

8. Conclusion

This paper has attempted to identify some of the issues that are relevant to the present-day teaching of VLSI design. The point has been emphasised that for the majority of students a systems approach is of more relevance than learning to construct cells at the silicon level. Hardware description languages such as VHDL have a vital role to play in teaching a systems approach to VLSI design since they can be used to demonstrate the need to get the chip to work as part of a complete system.

References

Gajski D. and Kuhn R. (1983): *New VLSI tools.*- IEEE Computer, v.16, No.12, pp.11-14.

Mead C. and Conway L. (1979): *Introduction to VLSI Systems.*- Addison-Wesley.