THE PRACTICALITIES OF PROGRAMMABILITY – MICROCONTROLLERS, FPGAS AND PLDS IN A LEARNING ENVIRONMENT

T. M. Forcer.

The University of Southampton, UK

ABSTRACT

Most of today's electronic products are based around programmable integrated circuits. Therefore today's electronic engineering degree courses have to provide both teaching and practical experience of programmable technologies as an essential thread within the core subject area of digital electronics. This demand pull is matched by the push which programmability gives in terms of benefits to teaching. The benefits can only be realised in full if a variety of practical issues, ranging from device handling to software support, are acknowledged, understood, and dealt with.

INTRODUCTION

Practical exercises using digital electronics feature in degree courses in electrical, electronic and computer engineering, and computer science. Students tackle a range of design and construction tasks, from simple combinatorial logic circuits through to complex systems. If the teaching is to be up-to-date and relevant to current engineering practice, programmable digital devices must feature in the exercises.

Digital integrated circuits (digital ICs, also referred to as “logic” ICs) provide a very wide range of functions. The simplest are gates, counters and registers, while the most complex are probably the microprocessors used in computers. Students need to know about simple fixed function ICs, and be able to use them sensibly, but there are benefits from moving to programmable devices at an early stage.

Programmable logic ICs have user-defined functions. Basing practical exercises on these components offers benefits of reduced wiring time and inventory range. This approach also gives an introduction to the use of tools such as hardware description languages (HDLs), which are essential for advanced work in many areas of digital electronics. Forcer et al (1) illustrates the application of such a strategy to all years of both Bachelors’ and Masters’ degree courses in electronic and computer engineering. Calazans and Moreas (2), Brown and Vrana (3) and Newman et al (4) report the benefits of programmable logic in teaching computer scientists, computer engineers and electrical engineers – showing that programmable digital ICs can benefit many courses. While some of this curriculum could be delivered with designs evaluated only by simulation, there are benefits from realising designs in hardware, as illustrated by Ferlin and Eleuterio (5) Hamblen et al (6) and Williams et al (7).

Two further types of programmable IC are the microprocessor and the microcontroller. The programming of the former is essentially “computer programming”, and will not be considered further. Microcontrollers are ICs which incorporate a processor along with memory and input/output (I/O) ports. Many courses include opportunities to design and implement microcontroller-based systems – see, for example, Hamrita and McClendon (8) – and this paper considers practical aspects of their use in the teaching laboratory.

Both programmable logic and microcontroller technologies are mature. This has benefits and drawbacks for the teacher. On the positive side, devices are cheap, development tools are plentiful and well-understood, and there is substantial theoretical and practical information available. The downside is that state-of-the-art ICs are not well suited to the teaching laboratory, since they have very small pins, are not easy to mount in sockets, operate at voltages well below those of discrete-gate ICs. This paper suggests approaches which allow students to experience relatively current tools, techniques and technologies without creating undue difficulties for course designers and laboratory managers.

HISTORICAL PERSPECTIVE – DIGITAL ICS IN THE TEACHING LABORATORY

Integrated circuits were developed during the 1960s, and rapidly became common in digital systems where space was at a premium, before entering mainstream use as price reductions made them competitive with discrete component assemblies.

By the early 1970s, the commonest digital ICs provided “TTL” circuitry in dual-in-line (DIL) packages, operating from +5V and 0V. These DIL-packaged ICs (“DIPs”) have two rows of pins spaced at 0.1”, with the rows 0.3” or 0.6” apart. The 5V TTL functions were standardised as “74-series” TTL. Reducing costs made it affordable for institutions to provide whole-class practical exercises about logic gates and sequential elements using digital ICs. Students learned to assemble systems with these ICs, gaining valuable experience in digital prototyping and debugging as well as in digital electronics itself.

As technology advanced, a consequence of Moore’s Law was that multi-gate levels of complexity – sub-systems such as adders and counters – became widely available as single ICs. When microprocessor and other complex devices arrived, students could build complete digital systems in the laboratory. Supporting resources such as solderless prototyping breadboards (“protoboards”) meant that the components were re-usable, with low wastage costs.

Systems built with onto protoboards have a considerable amount of wiring, with a high attendant probability of errors. These drawbacks were alleviated when user-programmable logic ICs became affordable.

PLDS, FPGAS AND MICROCONTROLLERS

The term “programmable logic device” (PLD) can have differing meanings. As well as being a generic term for any IC capable of having its function defined by a user, it is also used conventionally to refer to those programmable ICs with a fixed internal connection and logic architecture, where the programming consists purely of determining which of the (permanently connected) signal paths are active in producing output functions. All references in this paper to “PLD” are to ICs of this type. ICs where the architecture is an array of programmable logic cells embedded in a programmable and heirarchical interconnection matrix are usually referred to as FPGAs – Field Programmable Gate Arrays. Because of their large number of pins, FPGAs are almost invariably produced in packages such as Plastic Leaded Chip Carrier (PLCC), Pin Grid Array (PGA), Quad-edged Flat Pack (QFP) and Ball Grid Array (BGA). Of these, only the first two can be fitted into affordable sockets. If a fault develops in a non-socketed IC, the complete circuit board has to be replaced.

An intermediate class of programmable digital IC is the Complex PLD (CPLD). These have architectures which are equivalent to, or approximate, a superset of PLD architecture.

As defined above, a microcontroller is a digital IC containing a processor and sufficient additional resources to provide a complete digital system. Microcontrollers with pin counts of 40 or less are usually available as DIPs.

APPROACHES TO DEVICE PROGRAMMING

The implementation of the desired functionality in a programmable IC is variously referred to as “programming”, “downloading”, “configuration”, “blowing” and “zapping”. Some of these terms date from the early PLDs, where sections of the internal interconnect were subjected to controlled over-current, producing an open circuit in a manner analogous to blowing a fuse. A dedicated programming station with a Zero Insertion Force (ZIF) socket is used to supply the closely-controlled voltages and currents.

Many FPGAs use SRAM technology. The IC is configured by an internal processor collecting configuration data via designated I/O pins, often from an external EPROM or serial PROM.

Fuse-based PLDs were unsuitable for standard laboratory exercises because they had to be discarded after every use. Fortunately, by the late 1980s, PLDs using Flash and electrically-erasable (EE) technology became available, and it was practical to have whole-class exercises targetting these devices.

The next major development in programmability was in-system programming (isp). This took the reconfiguration approach which was the norm for SRAM-based FPGAs, and applied it to non-volatile Flash/EE PLDs. Depending on the required programming voltages and other factors, it became possible for students to develop their designs, download them, and observe the behaviour of their system, without removing the PLD from their protoboard. This meant there was no queueing of students at the IC programming station.

Even where an IC is not fully isp, it is sometimes possible to produce a pre-assembled unit which provides the equivalent of isp. Such an approach was taken in the design of the “Microsystems Experimenter Unit” at Southampton. Shown in Figure 1, this is used for a range of experimental work. Sehati (9) reports on the teaching benefits of isp and related technology.

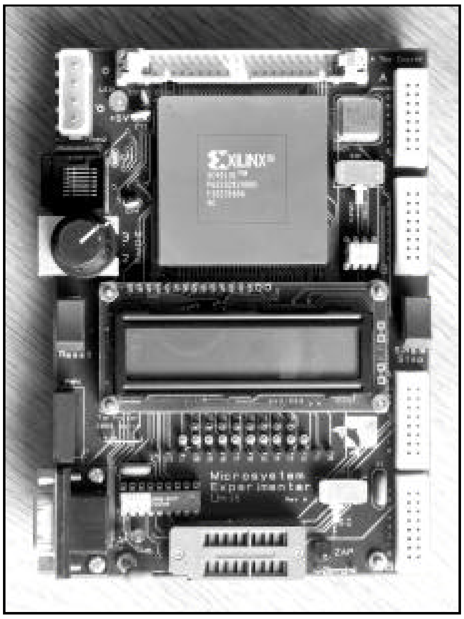

Figure 1 – FPGA and microcontroller experimenter unit

There are many isp schemes and systems. The range is reducing as new CPLDs and FPGAs increasingly comply with the JTAG boundary-scan standard – as extended to IEEE 1149.1 – which incorporates isp commands and features. Generally speaking, the older an IC design is, the less likely it is to implement IEEE 1149.1. Unlike the algorithms for programming non-isp PLDs, most isp protocols are published and can be implemented in user-designed equipment.

THE BENEFITS OF PROGRAMMABILITY FOR TEACHING

PLDs, FPGAs and microcontrollers can be used to advantage in taught courses, including in teaching laboratories, with benefits as detailed below.

Compatibility with commercial practice

Virtually all digital systems produced commercially now incorporate programmable digital ICs. It is essential that students who will be working on such designs after graduation have experience of relevant techniques during their course.

Consistent approach through the degree course

Programmable ICs can be introduced early in degree courses. Nixon developed such an approach to the initial teaching of digital design into a successful course text (10), and has discussed the principles and specifics of the implementation (11). This lays a sound foundation for later years as noted in Forcer et al (1).

Inventory

The stock of programmable ICs required to support a range of exercises beyond basic gate operation is substantially smaller than the equivalent stock of fixed-function ICs, in both the number of different devices required and the total number of ICs. It can also be easier to check that all items are fully-functional, since the testers need only support this much smaller number of different ICs. The reduced range of ICs is easier to manage.

Student construction

The substantial reduction in wiring required for systems based on programmable ICs compared to fixed-function ICs, and the similar reduction in re-wiring to implement changes in design, has two desirable consequences. First, students take less time to assemble a system, allowing more time to be devoted to debugging and development. Second, the simpler wiring is easier to debug and much less prone to the introduction of new errors as an exercise progresses.

Short redesign times

Particularly where isp is available, it takes little time for a student to go round the loop of amending a design, implementing the new design on the target hardware, and evaluating the revision. Clearly, students need to be taught how to make good use of this facility, so that development is systematic rather than erratic.

Motivation

If students are enthusiastic and motivated, they learn more, and learn faster. By reducing dull tasks such as wiring and by providing scope for innovation and individual approaches to problems, students soon come to regard using programmable electronics as enjoyable. Just because engineering is a serious subject, there is no reason why exercises cannot be fun. The versatility of programmable ICs aids the development of enjoyable experiments at all levels. The positive response of students is noted by many practitioners, particularly: (4), (5), (6) and (7).

PROBLEMS AND SOLUTIONS

For the designer of a laboratory exercise involving programmable ICs, four classes of practical issue can be of particular concern: IC packaging and technology; programming arrangements; design environment; cost. These issues are considered separately below, although there is inevitably some linking between them.

IC packaging and technology

DIPs are easily interconnected with 1/0.6 insulated wire on protoboards, as shown in Figure 2.

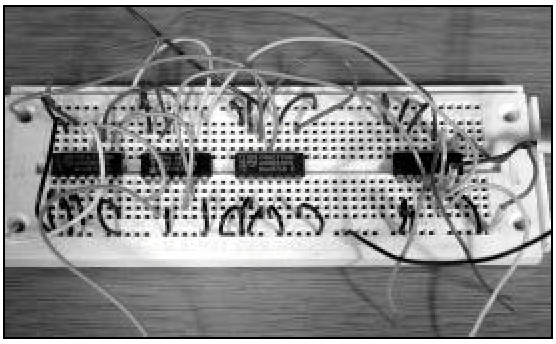

Figure 2 – ICs wired on a protoboard

A DIP is vulnerable to mechanical damage as it is removed from the protoboard. If students are equipped with an IC extraction tool, such damage can be minimised. The tool does not have to be a “professional” type such as shown in Figure 3a – the cheap and simple tool of Figure 3b works well.

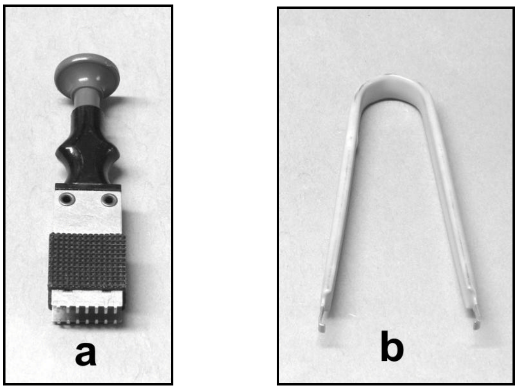

Figure 3 – DIP IC extractors

ICs packaged other than in DIPs cannot be fitted directly into a protoboard. An adaptor can convert a non-DIP IC to DIP, as shown in Figure 4. Alternatively, it can be incorporated into a pre-assembled unit, so that the student either has no wiring to do (the unit includes all input and output facilities) or needs only to connect a subset of the device’s pins via suitable wiring points. Figure 1 illustrates the latter approach – the circuit board provides all power and ground connections to the FPGA, along with clock drives, control signals and an LCD. User connections are wired via the small protoboard blocks seen at top right of the board.

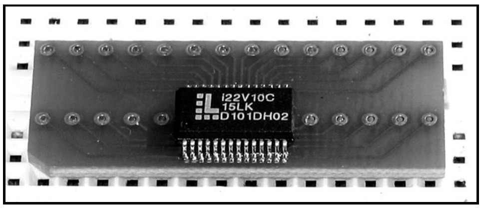

Figure 4 – small-outline IC mounted on adaptor to allow use in 0.1” protoboard

An issue deriving from advances in IC technology is that of incompatible logic-level standards. To make best use of the Moore’s Law advances in IC fabrication, operating and switching voltages have had to be reduced. ICs produced in 3.3V and 3.0V technology can often have their inputs driven from 5V-powered “TTL” (5VTTL) outputs without any ill effects, and the outputs of 3.3V and 3.0V logic should drive 5VTTL inputs correctly. But at lower supply voltages there is not usually any viable arrangement allowing direct connection to 5VTTL.

JEDEC standards exist defining the output and input requirements for various families of digital logic operating at nominal supply voltages 3.3V, 3.0V, 2.7V, 2.5V, 1.8V, 1.5V and 1.2V (12). All of these are used by at least one range of FPGAs, although not all are still in production. Very few 5V-technology FPGAs are on sale today. There are several ways of dealing with this issue:

Use only 5VTTL and compatible technology. This approach is not viable for new FPGAs, but is otherwise robust and straightforward.

Use 5VTTL-compatible signals for all wired connections. Conventional ICs can be used for discrete gating, and TTL signal generation and monitoring equipment remains fully usable. Level-translating buffers must be provided between incompatible devices. A disadvantage of this approach is that pins have to be grouped in sets of inputs and outputs, while bidirectional signals require data direction signals to be associated with them. Some commercial pre-assembled experimenter units have include buffers – any that do not should be treated with caution. Buffers have the benefit of protecting the primary ICs from erroneous connection. Replacing a failed buffer is much cheaper than replacing the primary IC.

As far as possible, use the programmable IC’s voltage supply and interface standard. This is probably the best approach for project work, as it allows the maximum benefit to be gained from the IC’s features. For whole-class exercises, the opportunities for error – and damage – could be significant due to the ease of inadvertently using the wrong standard for a particular device. Multiple standards are not recommended for introductory exercises for this reason.

Shift the whole laboratory standard for digital work from 5VTTL to a lower voltage scheme. As yet, there is no clear evidence of benefit from this approach. Southampton expects to move in this direction within a few years – probably changing second-year practice first, then moving down to first-year. The issue has been kept under review for some time, and some equipment procurement and design has taken into account the need for future change.

Providing hardware which is sanitised and protected to the point that the student obtains no sense of being "hands-on" can be counter-productive, as it removes an important element of real-world development practice. Conversely, the chances of damaging ICs by inappropriate handling, or by the use of out-of-specification voltages, must be minimised. Until recently, it was possible to use relatively advanced FPGA and CPLD devices while retaining a 5VTTL environment. That option is no longer available if the "relatively advanced" aspect is to be retained. While it is arguable that the levels of complexity of the newer FPGAs are far beyond the needs of whole-class exercises, the pressure on institutions to be state-of-the-art in their teaching is likely to drive an increasing penetration of low-voltage ICs into the teaching laboratory. In this context, note that students can be demotivated if they feel they are working with obsolete technology.

For the time being, it is recommended to: use 5VTTL DIPs unless there is good reason to do otherwise; issue students with IC extractors; use adaptors so that non-DIPs can be used as in protoboards; use pre-assembled units with appropriately conditioned I/O circuitry. Note that some commercial pre-assembled units are electrically good in that last respect, but do not provide easy mechanical connection to the I/O for user circuitry.

IC programming arrangements

Most non-isp PLDs have to be programmed with a properly-specified programming station, costing at least £300. These tend to be made available on the basis of only one or two per laboratory. Students transfer files to the programmer (often using floppy discs) and may have to queue to use the it. Using PLDs with isp makes every workstation into a programming station, avoiding file-transfer problems and eliminating queues.

One of the commonest PLDs is the 24-pin 22V10, with 10 input/output pins and 12 dedicated input pins. CPLDs are often described in terms of how many 22V10s they can replace. A drawback of the 22V10 is that it is not isp. This gap in the market was filled by the introduction of the 28-pin ispGAL22V10 – the extra four pins providing the programming port. This is the IC shown in Figure 4, and an adaptor was required since the IC is only produced in surface-mount packages.

FPGAs can usually be programmed in-system. A programming station is of no use for SRAM-based ICs – except for programming fixed configurations into serial EEPROMs.

Microcontroller programmers are cheaper than PLD programmers, and there are many self-build designs published on the Web for the more popular hobbyist devices. Some programming schemes can be implemented as isp.

Where the target IC is pre-assembled onto an experimenter unit (either commercial or in-house design), the unit will always provide for isp.

The principle of every workstation having a programming station should be adopted wherever possible. This is easier when isp ICs are used.

Design environment

To implement designs on programmable ICs, an appropriate suite of software is required. This may be as simple as a text-editor and freeware assembler for a microcontroller, or as complex as the integrated compiler, simulator, synthesiser and downloader used for FPGA designs using VHDL or Verilog. Some software is available at a discount through the FPGA manufacturers' University programmes, or via Europractise. After the software has been installed, there will normally be a need to manage updates and revisions.

A complication is that new versions of applications may not support older ICs. Where a set of exercises and experimenter units has been developed over several years, an institution may have to face the difficult choice between a redesign to target new ICs, or continuing with outdated software. Since researchers and some project students will probably wish to work with the newer ICs, both versions of the software may have to be kept operational.

A way of avoiding issues of this type is to rely entirely on "turn-key" solutions. By purchasing (or being donated) complete hardware/software packages, comprising all design applications and pre-assembled experimenter units, there is little likelihood of students or educators finding that designs cannot be implemented.

There is no single ideal design environment to suit all course needs. Institutions should expect to provide both an introductory and a full-featured HDL suite, as well as support for a range of microcontrollers. The specific choices should be informed by considerations of costs (including licence renewal and support fees), versatility (it is undesirable to be locked into ICs from a single manufacturer) and the benefits to graduates of competence in the particular packages.

Costs

Almost everyone in higher and further engineering education has to balance limited funding against a range of pressures to keep up to date with fast-moving technology. Some apparently attractive options, such as free or heavily-discounted design software, may not be as beneficial as they first appear. The total costs of providing the teaching programme must be considered, including any need to upgrade the workstations on which the software will run. Experimenter units solve many problems of high-density IC packages, but if they are to be used with external circuitry, the additional costs of interfacing arrangements can be significant – while if the interfacing does not include sacrificial buffers, sufficient units must be bought to cope with the inevitable losses when units are wired up incorrectly.

CONCLUSIONS

Programmability is a significant aspect of electronics, and must be addressed in many courses. While there are a range of potential difficulties concerned with using programmable ICs, these can be avoided or minimised by adopting clear strategies which are well-integrated into the teaching programme. Such strategies should: favour in-system programming; minimise ineffective student time spent on construction; promote the acquisition of transferrable skills – including skills learned during introductory exercises being re-used for advanced work; represent sound investments of the institution's resources. Following these guidelines should result in happy and motivated staff and students.

ACKNOWLEDGEMENTS

The author is grateful to the Head of School, Electronics and Computer Science, Professor Wendy Hall, for permission to publish this paper.

The author acknowledges valuable assistance from many colleagues, particularly Drs. Mark Nixon and John Carter.

1 Forcer, T., Nixon, M., and Zwolinski, M., IEE Proc. Eng Ed 2002, 37/1-37/6.

2 Calazans, M., and Moraes, F., 2001, IEEE Trans. Ed., **44**(2): 109-119.

3 Brown, G., and Vrana, N., 1995, IEEE Trans. Ed., **38**(2): 118-125

4 Newman, K., Hamblen, J., and Hall, S, 2002, IEEE Trans. Ed., **45**(3): 289-296

5 Ferlin, E. and Eleuterio, M., 1999, Proc ICEE'99, paper 190

6 Hamblen, J., Owen, H., Yalamanchili, S. and Dao, B., 1999, IEEE Trans. Ed., **42**(1): 8-14

7 Williams, R., Klenke, R. and Aylor, J., 2003, IEEE Trans. Ed., **46**(2): 296-301

8 Hamrita, T. and McClendon, R., 1997, IJEE, **13**(4), 269-274

9 Sehati, S., 2000, IJEEE, **37**(1): 86-94

10 Nixon, M., 1995, "Introductory Digital Design – a_programmable_approach", Macmillan, London, UK

11 Nixon, M., 1997, IEEE Trans. Ed., **40**(3): 193 – 206

12 EIA/JEDEC, 1999, JEDS-8B Interface Standard for Nominal 3V/3.3V Supply Digital Integrated Circuits (and subsequent addenda to JEDS-8).