PRCEEE-86

Pacific Region Conference on Electrical Engineering Education

Engineering Australia's Future: What Can We Do?

RMIT

ROYAL MELBOURNE INSTITUTE OF TECHNOLOGY · 1887 · CENTENARY · 1987 ·

PROCEEDINGS

Vue Grand Hotel, Queenscliff December 14-17, 1986

DIGITAL SYSTEMS AND COMPUTER ENGINEERING AT RMIT

G. K. Egan, Principal Lecturer,
Department of Communication and Electronic Engineering,
Royal Melbourne Institute of Technology.

Abstract

The three year digital systems subject sequence at the core of the Digital Systems and Computer Engineering Degree Course was introduced in 1981. The syllabus underlying these subjects stressed a systems first approach in lecture material leading to a more detailed treatment in the later years and a strong emphasis on problem solving in the associated laboratory programs.

While the problem oriented laboratory programs and systems first approach in the subject sequence has proved highly motivating for students, experience has shown that insufficient attention to the provision of staff confident with the requisite broad skill base necessary to support a demanding systems based syllabus will result in serious problems.

1.0 INTRODUCTION

In 1981 the time allocated to the treatment of Digital Systems material in the first three years of the existing Communications and Electronics strands in the Department of Communication and Electronic Engineering at RMIT was greatly expanded. In addition to the expanded time a new mode of presentation was introduced. This stressed a systems first approach in lecture material leading to a more detailed treatment in the later years and a strong emphasis on problem solving in the associated laboratory programs. The lecture material drew on experience with degree programs at established computer engineering departments overseas; these programs led to the 1983 IEEE Computer Society Model Program in Computer Science and Engineering [1].

As a direct consequence of the introduction of this material a third strand was identified and a degree program approved. This will allow 1986 4th year students to graduate with a Bachelor of Engineering in Digital Systems and Computer Engineering. The I.E. (Aust.) is currently studying the syllabus as a precursor to accreditation of the degree [2].

The three degree courses in the Department have a common three year core. Within this core there are subject sequences for the Communications, Electronics and Digital Systems and Computer Engineering strands. The particular degree qualification is determined by the choice of specialist units in the fourth year of the course. The digital systems units currently include such topics as:

- Computer Operating Systems
- Assemblers and Compilers
- Programmed Logic
- Computer Graphics
- Parallel Processing Structures
- Digital Integrated Circuit Design
- Control Systems

The selection process is constrained so that students must select some units from the other strands.

The core subject sequence of the Bachelor of Engineering in Digital Systems and Computer Engineering course and experience with its implementation and maintenance is the subject of this paper.

2. DIGITAL SYSTEMS AND COMPUTER ENGINEERING CORE SUBJECTS

The following sections describe briefly the intent of the core subjects, their inter-relationship and the mechanisms that were intended to maintain their integrity. *It is important to note that the material in these sections was abstracted from the original course documentation and, as will be explained, does not reflect completely the current situation.*

The syllabii for the course core subjects are presented in full in the Appendix for comparison with other courses. The total time allocated to this material is 266 hours of which approximately half is devoted to laboratory work.

2.1 Course Orientation

The course is couched in terms of digital computer systems. The study of digital computer systems provides a coherent framework within which material applicable to digital systems in general may be studied.

The first year subject introduces students to the concept of top-down iterative design with a system study of conventional digital computers. The balance and interaction of hardware and software techniques is strongly emphasised, while the traditional approach of treating hardware and software as separate entities is strongly discouraged. A detailed study of a particular computer system is not attempted in the first year as the subject's aim is to establish a 'coherent' insight without the clutter associated with many real systems (a result of separating hardware and software).

The second year subject expands in depth the concepts and techniques introduced in the first year in the context of real systems including the computer based laboratory stations.

The core of the third year subject concentrates on the application of digital computer systems in engineering while the balance of the material concentrates on detailed digital computer design and implementation. Both components emphasise computer assisted formal design and construction using appropriate and current technologies.

2.2 Laboratory Programs

The laboratory program is intended to provide a high level of student motivation by exposing students commencing in the first year subject to the 'hands on' use of digital systems. Pascal language exercises will support the hardware laboratory exercises and the emulation, simulation, language translation and documentation aspects of the formal lectures.

The laboratory programs are oriented towards problem solving. This is in contrast to the more traditional laboratory programs which tend to be well established, implying ready availability of solutions passed down to following students, and 'cook book' based. The laboratory problems are changed from year to year to involve the tutors in new situations in which they are required to demonstrate their own problem solving abilities to the students.

The students are given a brief problem statement usually with a number of achievement levels. They must then solve the problem within the constraints of the laboratory computer workstations and a hardware kit of components which they purchase. The hardware kit consists of a constrained list of analog and digital components, prototyping board, wire-wrap and other hand tools. The laboratory workstations support Pascal and assembly language program development. Students are encouraged to use the workstations for modelling, commissioning and as appropriate an integral part of their problem solution (the protyping board being interfaced via a 32-bit parallel interface on the workstation).

Current laboratory problems range from simple traffic light controllers in first year through to the re-targetting of a Pascal sub-set compiler to a zero/one address architecture and the subsequent design, construction and commissioning of a data-path and associated micro-code assembler and micro-code to support this architecture.

2.3 Course Coherence

The course builds from a system level understanding to detailed design and implementation techniques. To maintain the course structure it is vital that the educational objectives of every year are met in full. To assist in achieving this the following mechanisms have been implemented:

1) lecturing staff are responsible for demonstrating that subject objectives are being met in the first instance to the satisfaction of the lecturer of the successor subject and secondly to the digital group (an informal grouping of interested staff including the lecturers of the subjects described herein) as a whole.

2) examination papers and laboratory exercises are moderated by the lecturer of the successor subject.

3) lecturing staff act as tutors in subjects that are prerequisite to their own.

2.4 Subject Coherence

As the subject material is intended to form a coherent body of knowledge, laboratory sessions are not split at boundaries of lecturer competence. Specifically the exercising of hardware and software techniques is not separated into programming tutorials and hardware laboratories.

3. CONCLUSION

In many respects the lectures and associated laboratory programs may be considered a success. Approximately half of the students completing the course this year have elected to take the Bachelor of Engineering in Digital Systems and Computer Engineering qualification. Almost half of the 27 research students in the Department are conducting research in areas related to digital systems and computer engineering; research, and research and development funds from industry and government bodies support these programs and contribute significantly to the provision of computing and laboratory facilities.

In a recent course amendment the recommendations of the IEEE Model Program [1] relating to discrete mathematics were incorporated into the mathematics program.

From the lecturer's viewpoint the course material is demanding and, consistent with the study of systems, requires a concise, well structured and balanced delivery.

Unfortunately the pool of engineers with the requisite balance of skills is extremely small and in heavy demand. As a consequence we have not been able to attract sufficient staff and those that we have been able to attract, along with existing staff, have not had sufficient time release to develop the additional skills necessary to present the course material with confidence.

A course of this nature requires a continuing process of refinement and adaptation to changing circumstances. However complacency and the absence of sufficient time and consequent staff motivation may lead lead to degenerative process occurring. This may be manifested in over emphasis of the material with which lecturers are most confident which in turn can lead to the loss of material essential to dependant subjects and the progressive disintegration of the interface between subjects in successive years.

Failure to redress this process, should it occur, will lead inevitably, some would say naturally, to a narrow course which may owe little to the study of digital and computer systems, a course which is detail first , product and device specific, and perhaps more appropriate to a technician level course. At this point the material should not legitimately claim to constitute the core of a computer system engineering degree program.

A. APPENDIX (CORE SUBJECT SYLLABII)

A.1 Digital Systems 1 (First Year)

The aim of the subject is to provide students with an initial insight into the operation of conventional computer systems so that they may see clearly the inter-relationship and relevance of techniques and concepts which are studied in depth in the following years of the course.

A.1.1 Syllabus Outline

Introduction :- conventional computer systems, the sequential fetch/ execute cycle, hardware/ software inter-relationship, unconventional computer systems, the concept of time-sharing, conventional system overload.

Algorithm Design :- top-down successive-refinement or sub-goaling as a general design approach, graphical design aids, structure diagrams, graphical templates for common algorithmic structures, introduction to the algorithmic language Pascal, documentation.

Pascal :- Pascal syntax, syntax diagrams, BNF notation, Pascal statements, recursive definition, mapping graphical templates into Pascal, concept of semantics.

Pascal Procedures and Functions :- concept of program modules, parameter passing, formal and actual parameters, call by name, call by value, local and global variables, concept of scope, side-effects, standard Pascal Procedures and Functions.

Pascal Data-Structures :- concept of type, standard types, sub-range types, simple arrays, records, sets, files.

Boolean Algebra :- basic Boolean laws, notation (graphical, textual), minimisation (Karnaugh maps), word Boolean algebra, computer verification of design, overview of integrated circuit scale (SSI, MSI, LSI, VLSI).

Registers and Arithmetic :- binary number system, registers, register transfer, register transfer language, gated register transfer, notation, twos complement arithmetic (including multiplication and division), arithmetic circuits (addition and subtraction), ALUs, computer verification of design.

Finite State Machines :- notations (state transition), implementation algorithms, races and hazards, ROM and PLA based sequencers, arithmetic circuits (unsigned multiply and divide), simulation.

A Simple Machine :- major blocks of a single address architecture (Code, Data, IO, Control), register transfer language description of machine operations, instruction encoding, assembly language, architecture emulator, concept of mapping Pascal statements into assembly language templates (compilation), introduction to the assembly process.

Language Translation :- assembly language syntax, one symbol look-ahead, context free grammars, the symbol table, generation of assembler structure from assembly language syntax (parsers), single-pass assembler, treatment of forward references.

Simple Machine Hardware :- control unit organisation (bussed and dedicated path), multiplexers, decoders, gating sequences in terms of register transfers, concept of microinstructions (horizontal), microprogramming, ROM based microprogram sequencers, microinstruction assembly, simulation.

Accessing Data-Structures :- multiple-register two-address architecture, advantages (time, instruction length), reserved register use (program counter (PC), stack pointer (SP), name base (NB)), assembly language, accessing arrays without program modification (register indirect), record field accessing (indexed), addditional addressing modes (immediate, direct, register direct), instruction format, multi-word instructions, architecture emulation.

Procedure Entry and Exit :- saving the return address (register, data location), deficiencies, concept of a stack, call and return instructions, parameter passing (register, data location), deficiencies, passing parameters on the stack, the push and pop instructions, call by name (address), call by value (current value), local variables, concept of name base pointer (NB), displays.

Multiple-Register Machine Hardware :- control unit organisation, register files, multiplexers, horizontal microinstructions, microinstruction sequences for machine instructions, constants, effective address calculation, simulation.

Input and Output :- character devices, character sets, decoding and encoding of numbers, concept of error checking (parity), slow input-output devices, device status and control, overlapped input-output, memory mapped input-output, polling, buffering, introduction to interrupts.

Multiple User and Multiple Task Systems :- interrupts, scheduling, priority, resource allocation (storage, devices), relocation, shared resources, deadlocks, semaphores.

A.2 Digital Systems 2 (Second Year)

The aim of the subject is to extend the techniques and concepts introduced in Digital Systems 1.

A.2.1 Syllabus Outline

Introduction :- review of Digital Systems 1.

FORTRAN :- FORTRAN templates, comparision with Pascal (types, data- structures, control structures, subroutine entry, commons), advantages, deficiencies.

Software Structures :- code generation by compilers, PL/0 compiler, high-level/low-level language tradeoffs (runtime speed ratios, rate of production, documentation, maintenance), interfacing high-level and low-level language program modules, libraries, management, interpreters.

Micro-programmed Processor Units :- vertical microinstructions, bit-slice microprocessor based machines, re-examination of the multiple-register machine control-unit, minimisation techniques, asynchronous systems, simulation.

Hardwired Processor Units :- comparision with microprogrammed units, advantages (speed), disadvantages (flexibility), design methods (state table, delay-element, sequence counter), notation, minimisation, simulation.

Arithmetic and Mathematical Functions :- floating point numbers, floating point arithmetic, trigonometric functions, microprogram implementation, hardwired implementation, tradeoffs, simulation.

Internal System Communication :- busses, dedicated path, protocols, arbitration, timing.

Storage Devices :- access modes (random, serial), storage media (semiconductor, core, disks, bubble, tapes), access time, operation, physical characteristics.

Primary Storage :- virtual memory, locality, hierarchies, access time, segmentation, paging, replacement policies.

Secondary Storage :- file devices, organisation (contiguous, indexed, indexed sequential, linked sequential, random), directories.

Input and Output :- serial, parallel, introduction to analog input-output (ADCs DACs), introduction to controllers (DMA, CRT, disk, communications), multiplexers and demultiplexers.

Operating Systems :- overview of internal organisation, basic modules (interrupt handlers, schedulers, peripheral handlers), internal communication (message passing), protection, system languages (C, Concurrent Pascal, Modula), distributed operating systems.

Architecture Case Studies :- examples of micro, mini, and mainframe architectures.

A.3 Digital Systems 3 (Third Year)

The aim of the subject is to examine in detail implementation techniques and technologies specific to the application of digital systems, and to develop and exercise formally the design techniques and implementation technologies specific to the construction of digital systems.

A.3.1 Syllabus Outline

Introduction :- review of Digital Systems 2, subject overview.

Digital Components :- current technologies, speed, power, packaging, power supply requirements, component documentation, selection criteria.

Signal Transmission :- propagation, skewing, noise, bypass, transmission line effects, crosstalk, cabling, termination (passive, active).

Input and Output :- ADCs, DACs (selection, design for simplicity/ speed), digital controllers (types, selection, design), interfacing (level translation, buffering), environmental considerations (noise, hazards), protection.

Error Correction Techniques :- error mechanisms, coding techniques (parity, hamming, fire, orchard, CRC), detection/ correction circuitry, syndrome logging, simulation.

Digital Networks :- network topologies (star, ring, tree, redundant path), real and virtual circuits, switching, routing, layered protocols, system implications (processor pipelines, front-end processors), queueing theory, traffic simulation.

Instruction Sets :- application considerations, design, parallel software development (emulation), evaluation (benchmarks).

Computer Aided Design :- hardware description languages, hardware compilers, implementation technology templates, partitioning, placement, layering, tracking, design data-bases, simulation, development systems, problem areas.

Computer Aided Documentation :- formal specification, specification languages, logic diagrams, wire lists, mechanical diagrams, test generation.

Fabrication Considerations :- racking, power supplies, cooling, board interconnection, printed circuit board manufacture, automatic and semi-automatic wire wrap.

Computer Aided Commissioning :- bottom-up commissioning, in circuit emulation, diagnostics, diagnostic/ maintenance processors, documentation revision, unwrap/rewrap schedules, performance evaluation.

REFERENCES

[1] The 1983 IEEE Computer Society Model Program in Computer Science and Engineering, IEEE, Dec. 1983, ISBN 0-8186-0932-X.

[2] Perspectives on Engineering and Computing Systems, Report to the Institution of Engineers, Australia, May 1985.