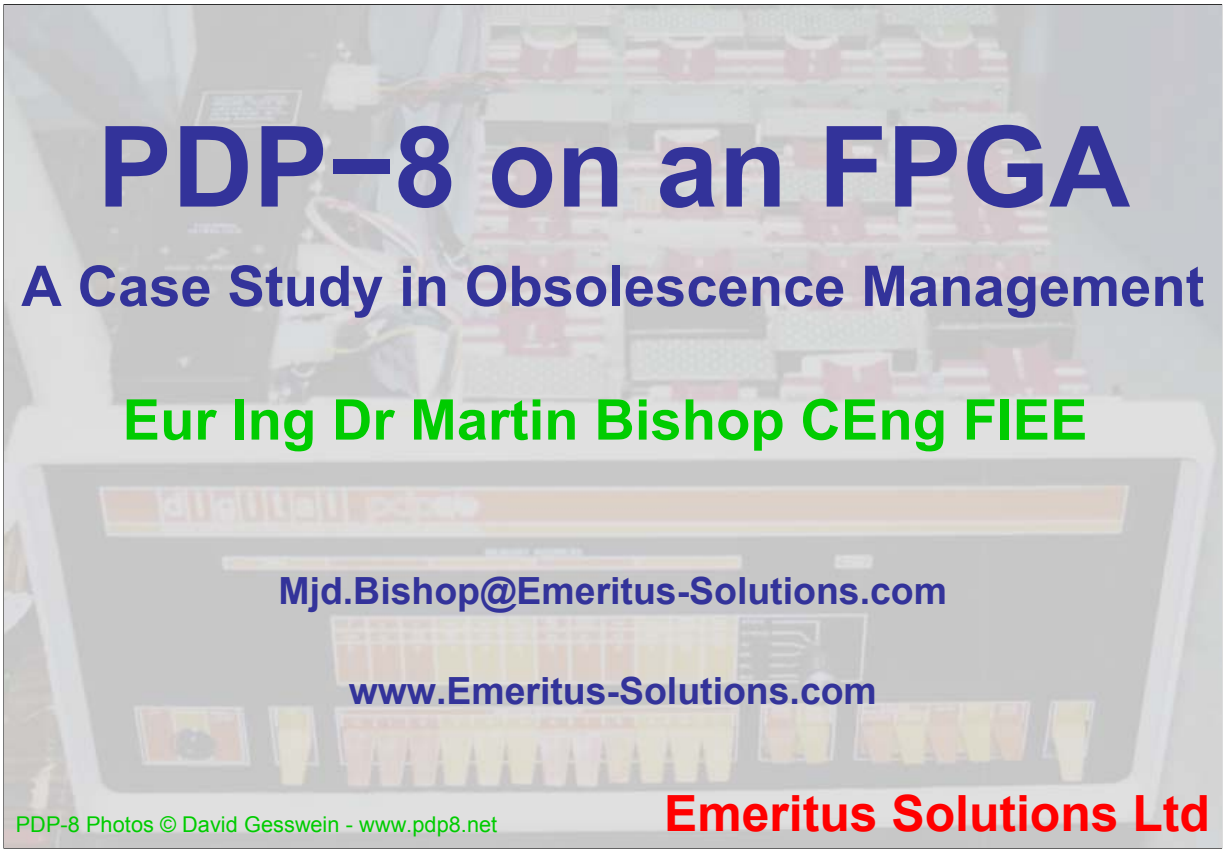
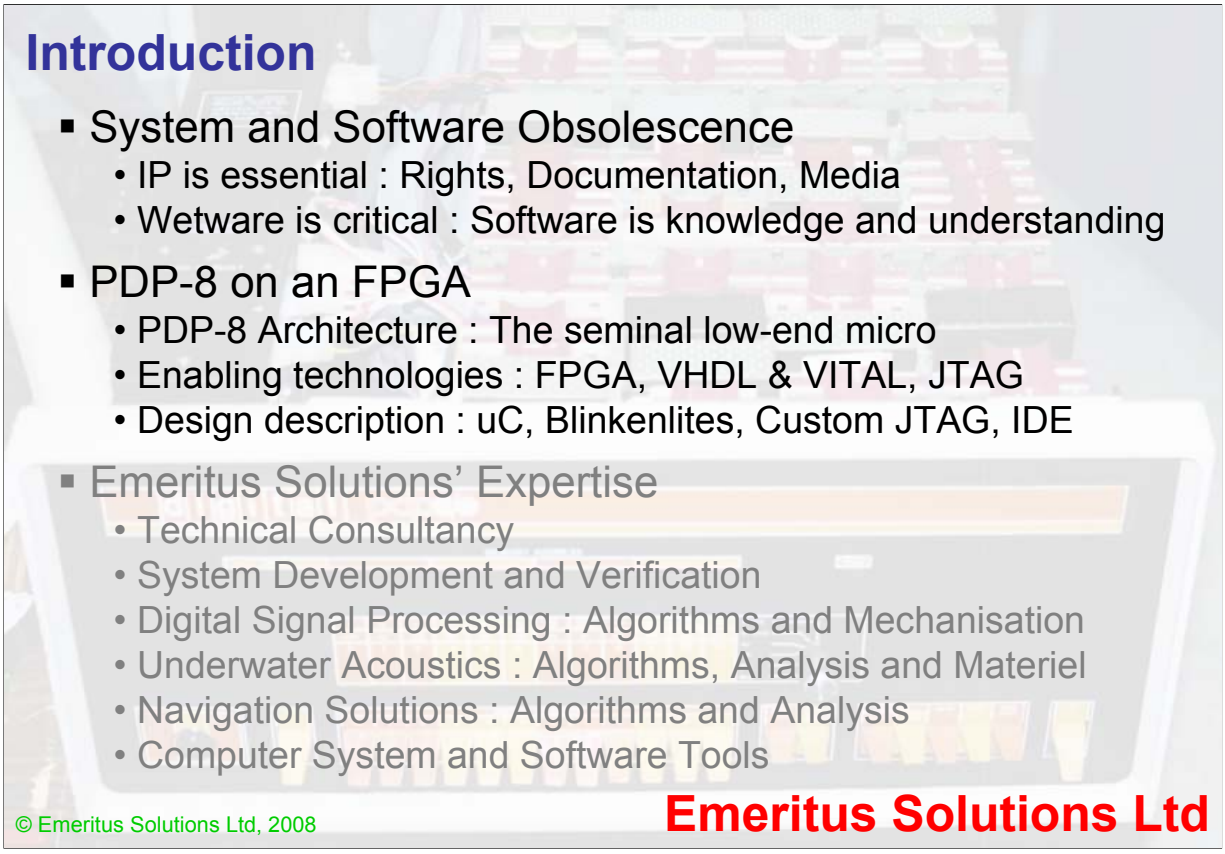


Abstract

Key issues in the obsolescence management of embedded computer systems are considered, using the re-implementation of the PDP-8 a classic 1960's minicomputer on an FPGA as an exemplar.

The interplay of system, software and hardware obsolescence is identified. A brief review of obsolescence in general and software obsolescence in particular is presented. The fundamental role of intellectual property rights in enabling obsolescence management is illustrated. The critical contribution of sentient software and system design expertise is emphasised.

The re-implementation of a diskless Digital PDP 8/E on an FPGA is described.



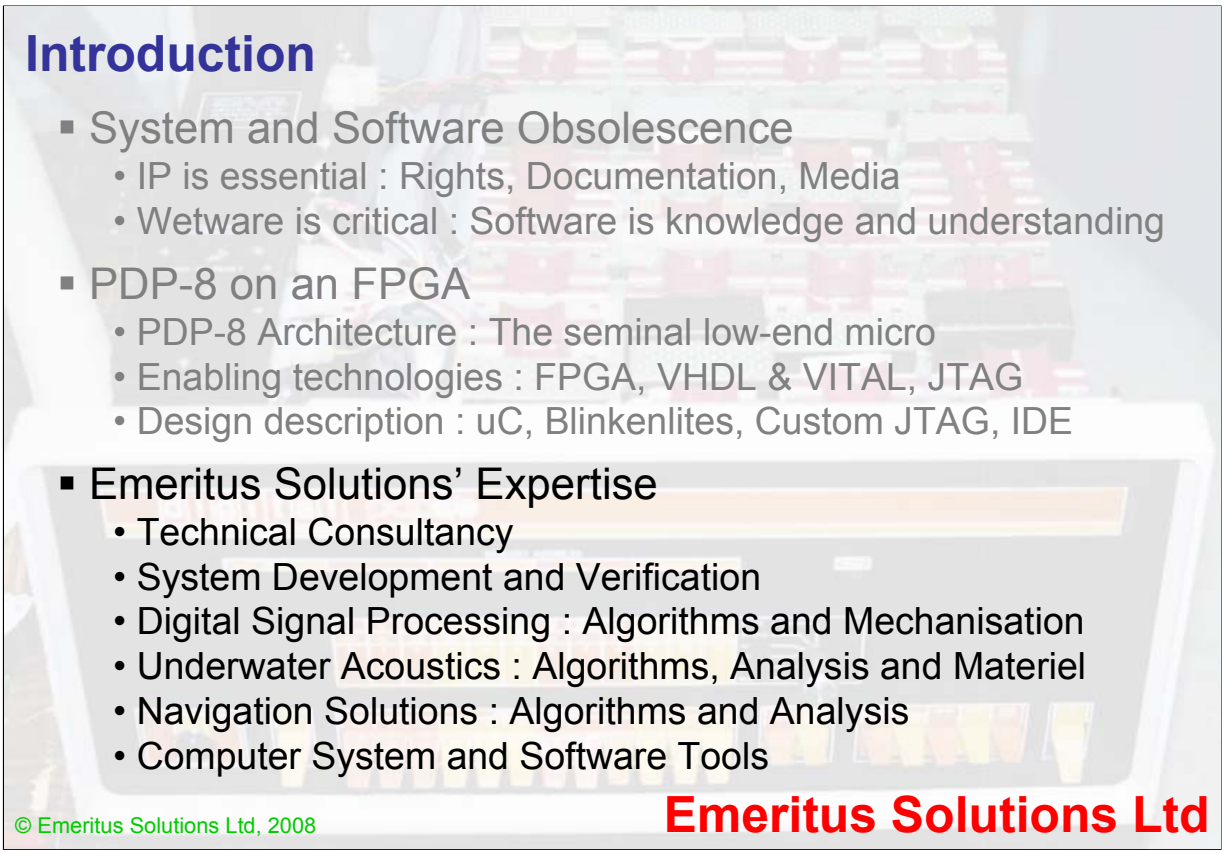
System and Software Obsolescence

Intellectual property (IP) is fundamental to the effective management of system and software obsolescence. Appropriate legal rights to executables, third party packages, software tools and operating systems are all essential. The necessary rights can range from the transfer of licensing to different hardware to access to source code held in escrow. Additionally, comprehensive (paper and pdf) documentation and complete soft (source and binary) copies of all programs, tools, OS, build scripts, etc are essential.

Knowledge and understanding of any software are however even more critical than it's availability. At one end of the spectrum, software which is actively maintained or developed implicitly possesses this critical capability and bears the attendant cost. At the other extreme, a pile of dusty files and disks may linger on a shelf with little or no corporate memory of there content. Commonly, the situation lies between these extremes with partial corporate memory of the system or software. Obviously, sufficiently expert personnel can (at a cost) reconstruct the required knowledge. However, starting from ignorance is expensive – not letting the wetware dissipate and desiccate is invariably preferable.

PDP-8 on an FPGA

The PDP-8 is used as an example because it is simple, yet useful, and because the necessary IP is available (*de facto* if not *de jure*), documentation and soft copy are available on the internet for architecture, utilities and (importantly for design verification) diagnostic software. The PDP-8 was implemented on a Xilinx Spartan 3 FPGA both to have the capability, but more importantly to evaluate the Xilinx ISE (free, Web Pack) FPGA tools and to evaluate the use of the JTAG user register to implement control and data exchanges between an FPGA and a laptop PC.



Emeritus Solutions' Expertise

Technical Consultancy is frequently the mechanism by which Emeritus Solutions' expertise is initially accessed. Specimen subject domains are: Oil and Gas, Sonar, Navigation, Communication and Computing.

Equally, system development and verification whether from a clean or well smudged sheet of paper are core competences with expertise available to identify current and necessary conformance to requirements. Typical systems will commonly include a mixture of PCs, embedded processors, microcontrollers, FPGAs and Digital Signal Processors (DSPs).

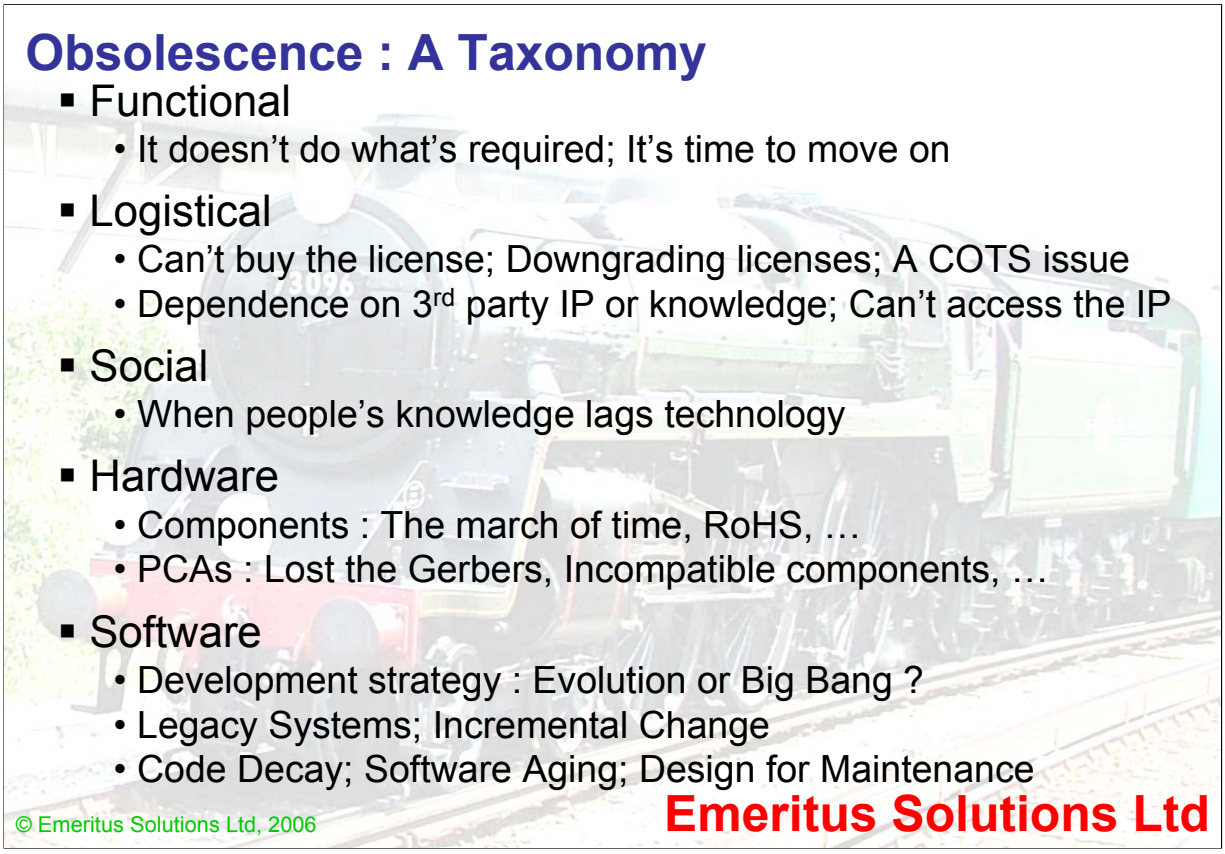
In DSP (Digital Signal Processing) Emeritus Solutions has extensive expertise in algorithm design and implementation for time series processing. This expertise encompasses filter analysis, design and mechanisation, complex signal (I+Q) representations and hetrodyne design, spectrum (fft / narrowband) and correlation (broadband) processing, non-linear processing, beamforming, signal measurement and of course signal conversion, data transmission and A to D & D to A conversion.

A profound understanding of underwater acoustics, sonar systems and algorithms is available. Our expertise encompasses a broad spectrum from propagation, through sonar equation evaluations, to system implementation and grooming.

Our navigation expertise is based on a profound understanding of underwater tracking systems. Specifically, Emeritus Solutions has sharp tools for the analysis and implementation of trilateration tracking solutions. Our tools for the analysis and synthesis of tracking systems while tailored to sub-sea requirements are applicable to the analysis of GPS, Loran-C and mobile phone systems. Additionally we also have extensive experience of GPS and highly accurate clocks.

In Computer Systems our expertise extends from machine code to productivity tools.

Finally, in Electronics and everyday engineering our expertise is of course extensive.



Functional Obsolescence

Steam railway locomotives provide an classic example. Yesterdays workhorse is today's dinosaur. Unfavourable metrics: power to weight, running cost, …

Logistical Obsolescence

"They won't sell you it", or you can't access the IP or knowledge. In addition to regulatory constraints, supplier's business practices can be an important factor.

Social Obsolescence

People's skills and knowledge become dated and inapplicable. Obsolescence affects people, processes and organisations – not just physical objects.

Hardware Obsolescence

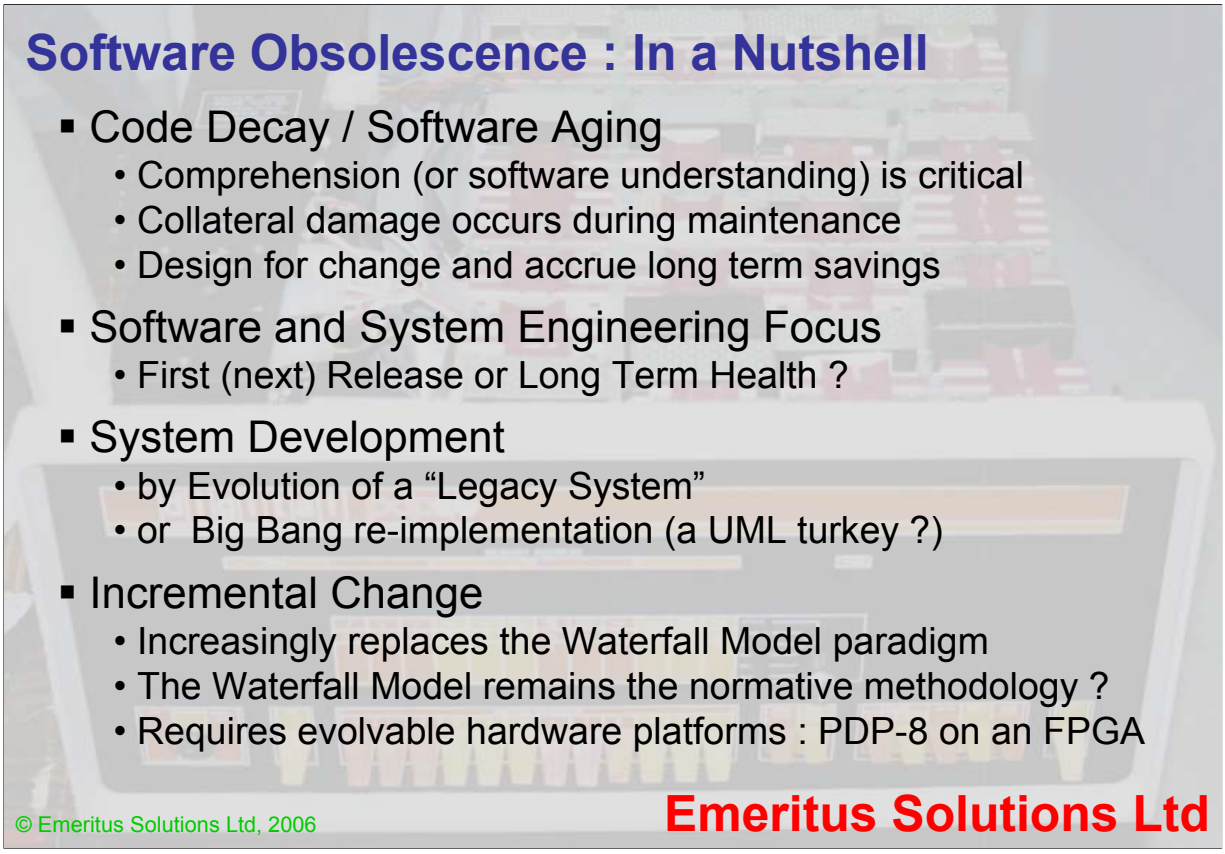
Components : today's bleeding edge is tomorrows unobtanium.

Printed circuit assemblies : eventually you can't build them without redesign.

Software Obsolescence

Software obsolescence can result both from external (functional, logistical, social and hardware) factors and from internal factors (e.g. code decay and software aging). The risk perceived to be associated with software modification is a good litmus test of software obsolescence. Software design / development style and software maintenance choices profoundly affect the rate and extent of software obsolescence.

Broadly, software obsolescence is inversely proportional to software expenditure. Software is very much a "capability in being", only the aspects which are actively used and maintained remain capable of (rapid) modification or development.



Code Decay / Software Aging

Code aging happens without remediation, unless effort is applied to update the software's structure, all modifications reduce code quality. Understanding the code is fundamental: both to avoid side effects which introduce defects and to maintain the code's structure, style and comprehensibility. Software can, and must if you don't want it to rot, be designed for maintenance – but judicious design costs more.

Software and System Engineering Focus

Is the time horizon Friday, this quarter or the long haul. Is what matters delivery or sustainability. Software maintainability and (inherent) obsolescence are substantially defined by the project environment and corporate culture.

System Development

The evolution of legacy systems, or a cascade of prototypes, implicitly uses incremental development. A great advantage is that radical mistakes cannot be made and that incremental feedback on system design and use can be obtained.

The diametrically opposite approach of a BigBang re-implementation, perhaps using executable UML, nominally saves the time "wasted" on intermediate steps. However, the output is frequently all or nothing, and few people can really "read" and comprehend drawings, specifications and designs without a concrete system. Therefore incremental development often ensues, as night follows day.

Incremental Change

Incremental change and the waterfall model (of development) can be mixed and matched. The waterfall model could be used at each iteration and is specifically appropriate to the V&V of (safety) critical (sub-)systems. The non-trivial cost of testing, useful documentation and design for re-usability should not be forgotten.

PDP-8 / DEC Minicomputer Timeline

- PDP-1 (18 bit x 4 ki words) : 1960
- PDP-8 (12 bit x 4 ki words) : 1965 .. 1990, ~50k built
  - PDP-8 1965 Transistor; Negative IO bus
  - PDP-8/S 1966-70 Transistor; Serial ALU – very slow
  - PDP-8/I 1968-71 TTL (SSI)
  - PDP-8/E 1970-78 TTL (MSI); Omnibus
  - PDP-8/A 1978-80 Semiconductor memory
  - VT78 1978-80 Intersil 8100 (LSI) workstation
  - DecMate 1980-90 Harris 6120 (LSI) workstation
  - www.faqs.org/faqs/dec-faq/pdp8/
- PDP-11 (16 bit x 32 ki words) : 1970 .. 1997, ~600k built
- VAX-11 (32 bit x 1 Gi byte) : 1978 .. 2000
- Alfa AXP (64 bit x 4 Ei byte) : 1992 .. 2007

Emeritus Solutions Ltd

The Digital Equipment Corporation

The Digital Equipment Corporation, also known as DEC and Digital, was founded in the late 1950's in Maynard, Massachusetts to build Programmable Data Processors (PDPs). The PDPs were some of the first mini-computers, the disinformative choice of name was intentional as Computers were always over time and budget …

The PDP-8

The PDP-8 was DEC's first volume product. A 12 bit machine with a 4 ki Word linear address space, it is one of the classic minicomputers. The PDP-8 was "state of the art" in the second half of the 1960's. Subsequently, it was used as an embedded building block until the advent of the microprocessor in the late 1970's. In the late 1970's, it was implemented in LSI as a microprocessor and used as the basis of several DEC workstations.

The PDP-8/E

The PDP-8/E represents the mature PDP-8 architecture. As it lies at the end of the paper tape era and before the beginning of the LSI era both diagnostics and (comprehensible) circuit diagrams are available. Additionally, as the PDP-8 was targeted at system integrators, for laboratory and OEM applications, comprehensive architecture and interface handbooks were published. How many modern products possess equivalent documentation and diagnostics for verification and validation ?

Subsequent DEC MiniComputers

The PDP-11 was DEC's second volume product, a 16 bit machine with a 32 ki Word linear address space. Again one of the classic minicomputers, it was "state of the art" from the early to late 1970's.

The VAX-11 and Alfa AXP followed on from the PDP-11. HP continue to offer the OpenVMS operating system to a loyal high availability following. Interestingly the increase in word length and address space leads Windows by almost twenty years.

DEC PDP−8/I : 1968 .. 1971

Photo © David Gesswein - www.pdp8.net

Emeritus Solutions Ltd

PDP-8/I

The PDP-8/I simply because a nice picture of a blinkenlites console was available. The PDP-8/I was implemented using SSI TTL logic. The front panel is an important HCI (Human Computer Interface) on the PDP-8, used to initiate bootstraps, for engineering (maintenance) work and to provide an indication of it's operation. The octal numbers printed at the left of the front panel are for the rim loader (bootstrap).

The Blinkenlites

Separate lights were provided on the 8/I for each of the major processor registers. On later models the display was multiplexed to a single row of lights on the 8/E, to an octal display on the 8/A and by 1980 was no longer provided on minicomputers.

The accumulator and link (carry) registers are at the arithmetic heart of the PDP-8.

The Multiplier Quotient and Step Counter registers belong to the EAE (Extended Arithmetic Extension) which primarily provided a hardware multiply capability.

Memory Address and Buffer registers indicate traffic on the memory bus.

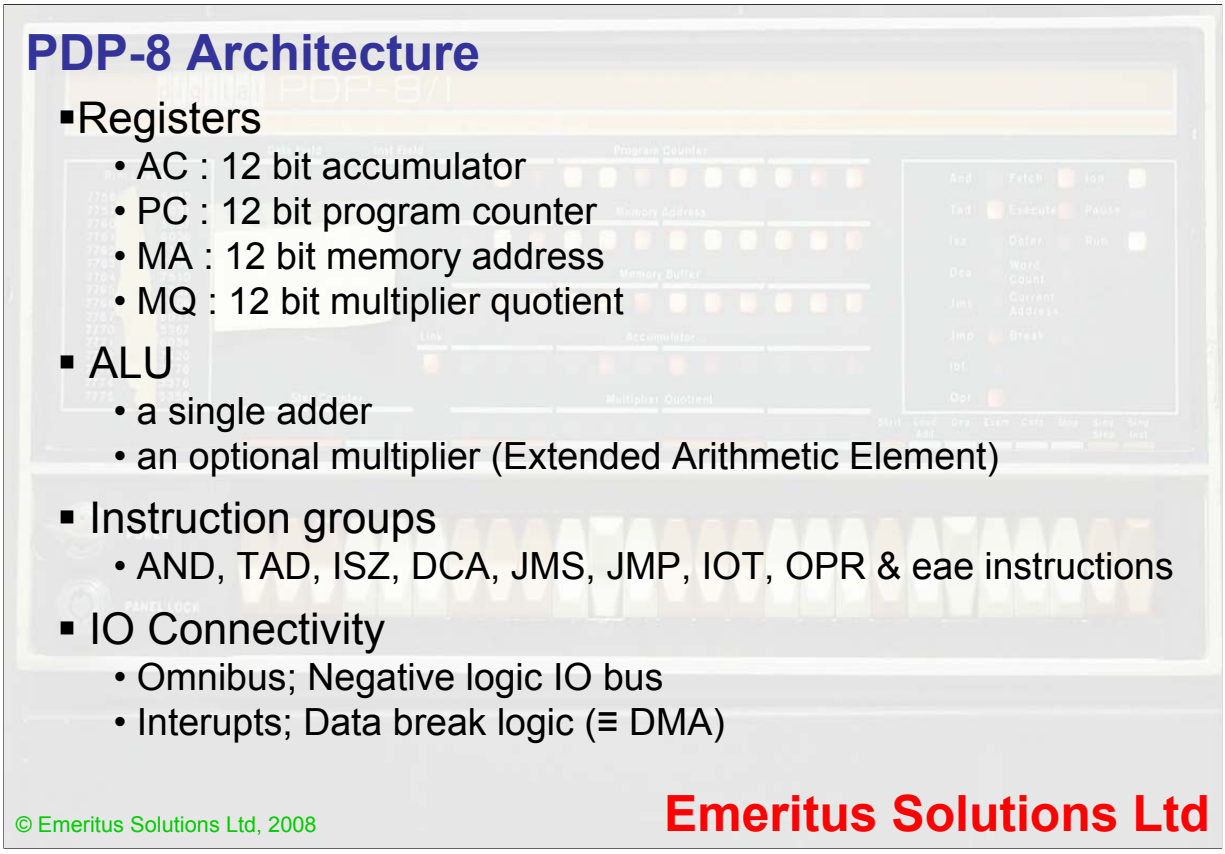
The Program Counter fulfils its usual function. The Data and Inst Field registers provide memory segmentation to support 8 x 4 ki Words of memory.

The Break, Word Count and Current Address registers hint at the machines DMA capabilities. Either the CPU or a controller could perform the word count and address computations required for what are designated Data Break transfers.

The Ion light is the status of the global interrupt enable.

The Toggle Keys

The toggle keys permit the examination and deposit of values in memory. Also, they control the processor. The granularity of control provided is quite fine, with the ability to step at both instruction level and at the processor step level : fetch, defer (indirect addressing) and execute. The panel lock is of course an operator lock out.



Registers and ALU

The inventory of registers is minimal, unsurprising as low cost was a major design driver. Similarly, the single adder in the ALU does everything : PC updates, MA updates and arithmetic into the accumulator.

The Extended Arithmetic Element (EAE) provides hardware multiply support and related arithmetic capabilities. The EAE is of course much more complex than the ALU.

Instruction Groups

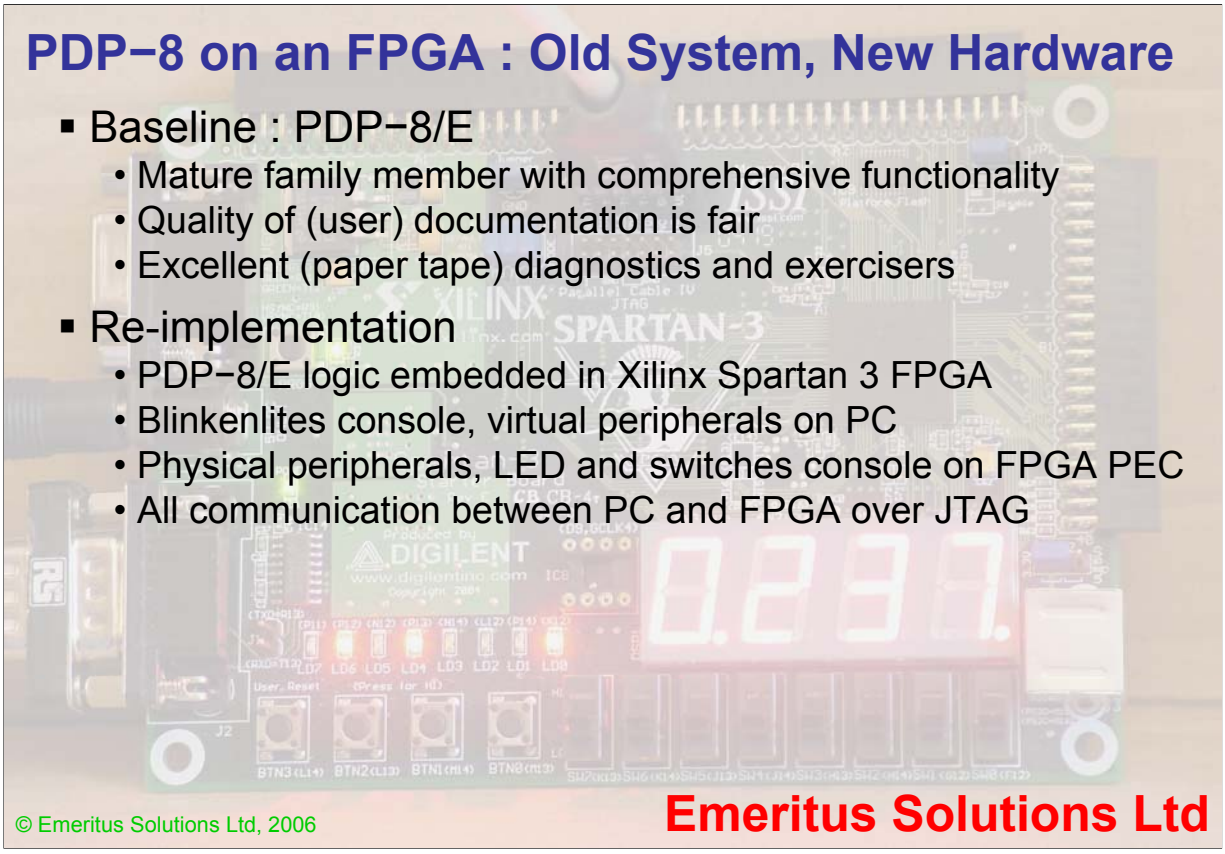
The instructions provided are simultaneously both very simple and surprisingly complex. The arithmetic operations provided are all between a memory location and the accumulator : AND – logical and, TAD – two's complement addition, ISZ – increment and skip if zero, and DCA – deposit and clear accumulator. The program control operations provided are : JMS – jump to subroutine, and JMP – jump to address.

The nine bit address field provided in these instructions provides an 7 bit address, a page selection bit and an indirection bit. The seven bit address is either applied to the current page, implied by the high order bits of the PC, or to page zero. Additionally if an indirect access to page zero locations 10_8 .. 17_8 is made the location is post-incremented.

If the addressing seems byzantine the IOT and OPR instructions are even more so.

IO Connectivity

The provision of interrupts and DMA (called data break) is basic with each capability having only a single level of arbitration. However, a bus for attaching standard or custom peripherals is provided. And, the peripheral's response to an (IOT) instruction is loaded into the accumulator. A very flexible concept which is very applicable to an inexpensive machine when bus and processor speeds were similar.



Baseline : PDP-8/E

The PDP-8/E is the “classic” model of the PDP-8 family.

The PDP/8-E is well documented at the user, architectural and circuit levels.

Paper tape diagnostics and exercisers are available for the PDP-8/E in both binary (loader) format and as (pdf) listings. Source code would be nice …

Re-Implementation

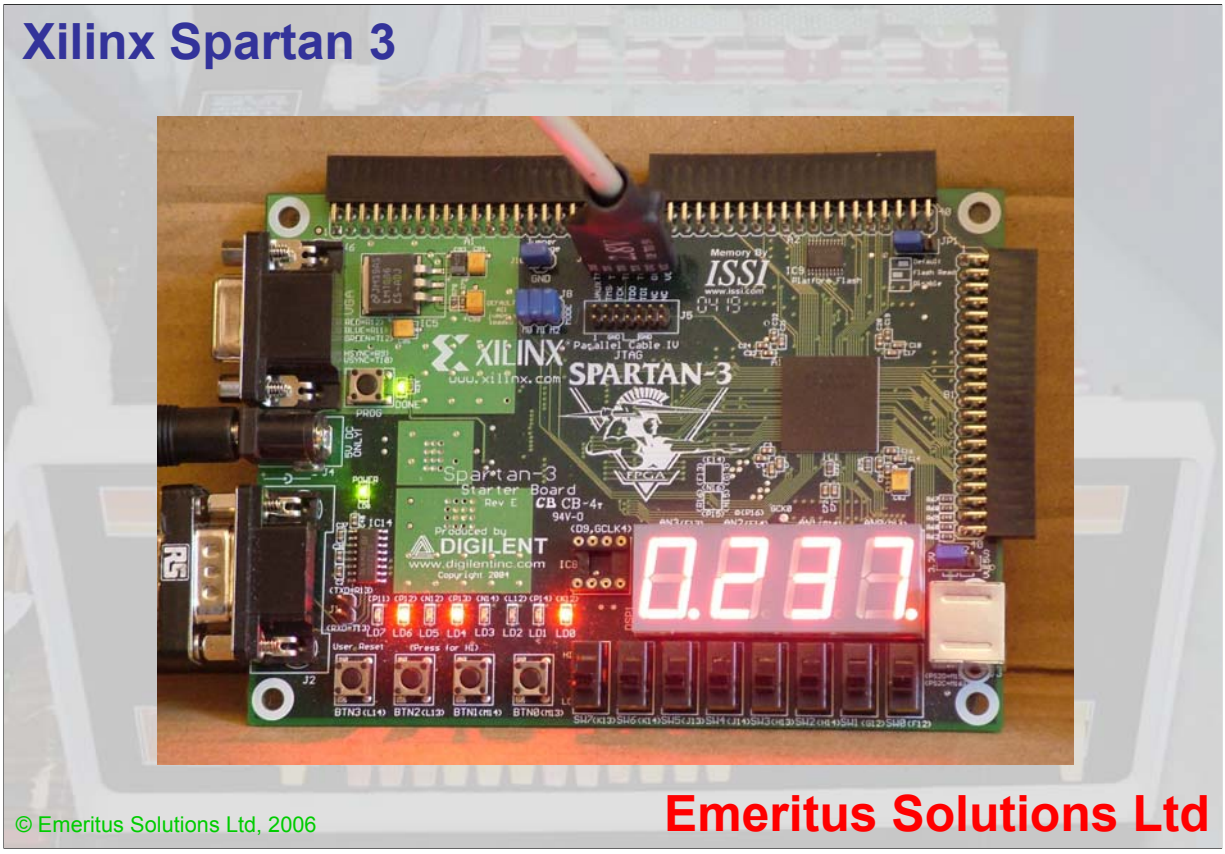
A Xilinx Spartan 3 FPGA was used as it supports the implementation of user communication over JTAG, possesses ample on chip memory and we required a shakedown project for that FPGA and Xilinx’s ISE tool chain..

The design requirement was for :

• a basic blinkenlites console on the hardware platform, especially step / run / stop logic and hardware register displays for low level debugging.

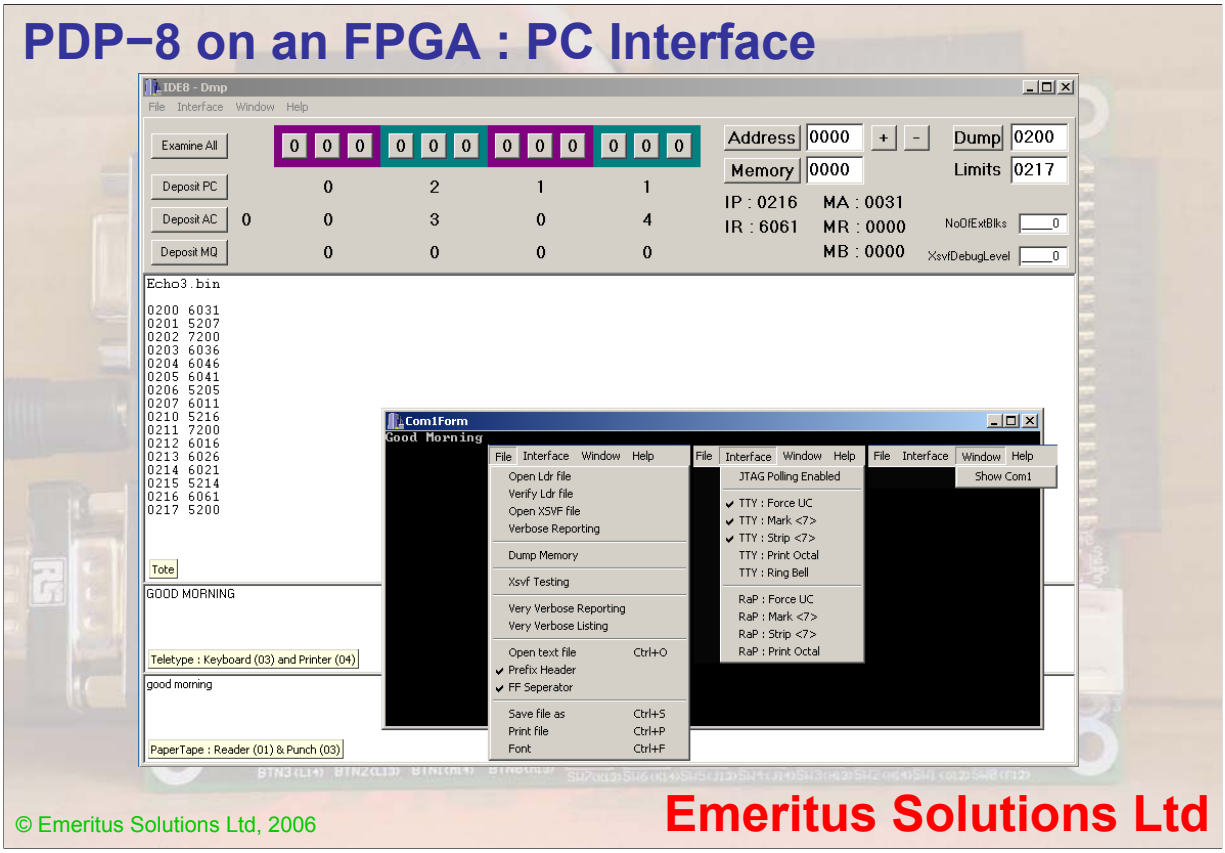
• a virtual console on a PC providing the traditional register displays, memory peek and poke, and virtual peripherals (console teletype, and paper tape reader and punch).

• All communications between the PC and FPGA to be over JTAG, and for memory and register access to be available when the PDP-8 processor is running.



Xilinx Spartan 3 Board

The $99 Digilent Spartan 3 Starter Board provides sufficient resources; the JTAG lead is included in the price. In addition to a small (~200k gate, 15 x 15 mm) FPGA with a JTAG interface, the board provides seven segment displays, slide and push button switches, an RS 232 port, and further unused capabilities.



Blinkenlites Interface

The blinkenlites interface is inspired by a traditional hardware console of the PDP-8 era. However, it is significantly augmented to provide modern IDE (Integrated Development Environment) capabilities melded with the look and feel of switches and lights.

The switch register is classically implemented with mouse clicks substituting mechanical toggle switches. The “loadable” registers (PC, AC and MQ) have individual deposit buttons – they take their value from the switch register.

The Examine All button uploads all of the significant processor registers : PC, AC, MQ, IP (instruction pointer), IR (instruction register), MA (memory address), MR (memory – write – register) and MB (memory – read- buffer).

The Address, Memory, + and – buttons permit the peek and poke of PDP-8 memory.

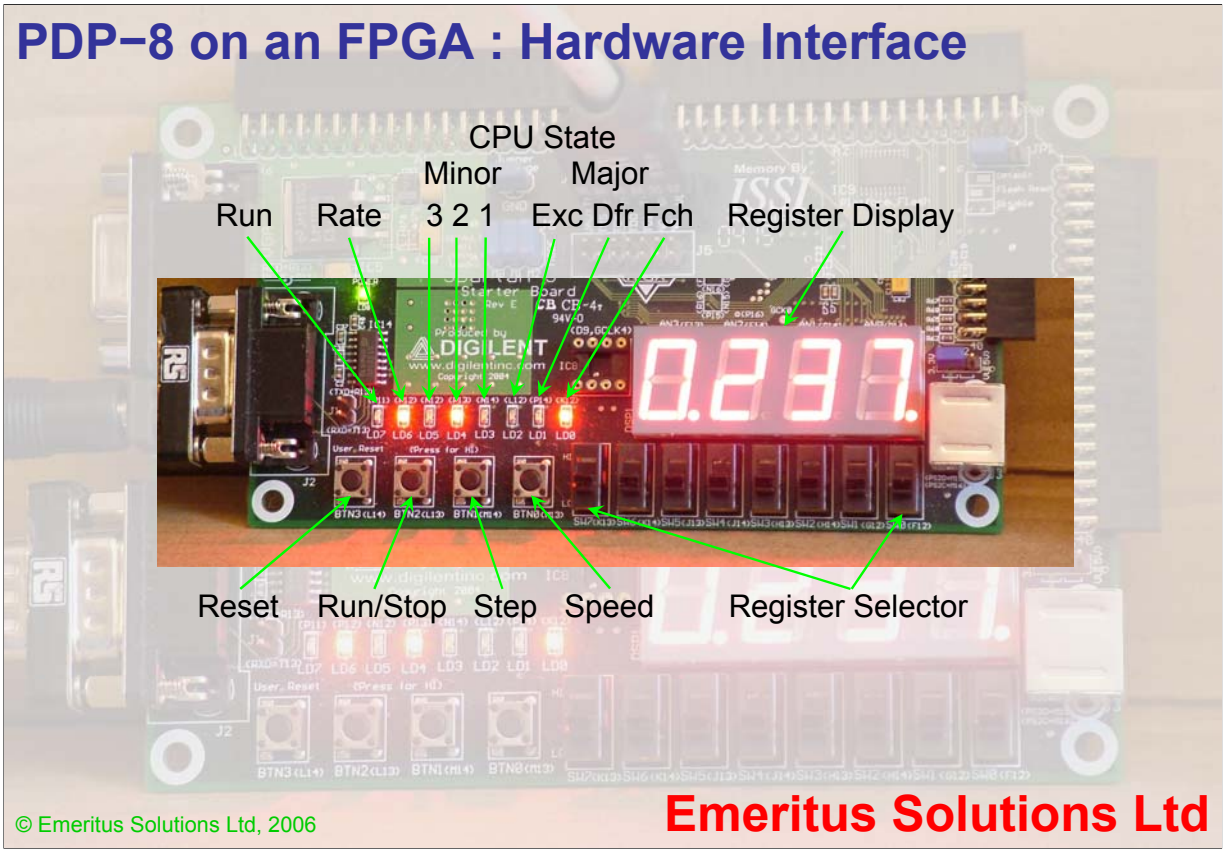
The Dump Limits capability implements a PDP-8 memory snapshot capability.

Embedded Peripherals

The (paper tape) reader, punch, (teletype) keyboard and printer are virtual peripherals which communicate over JTAG. The Tote is a physical peripheral connected to the PCs RS 232 Com1 port. Byte formatting of the TTY and RaP streams are controlled from the main menu bar.

Loader

The main menu bar provides both load and verify capabilities for standard PDP-8 binary file formats.



State LEDs

The discrete LEDs display the PDP-8/E's processor state. The run LED provides a run/stop indication. The Rate LED flashes slowly (1 Hz minor state execution), quickly (1 kHz) or stays on (1 MHz); the FPGA clock runs at 50 MHz and the combinatorial delays are less than 20 ns – the PDP-8/E could zip along. The CPU state LEDS indicate Fetch / Defer / Execute and their (internal) minor states; as the four minor states of a PDP-8 memory cycle are not required by the FPGA's double ported memory only the functionally necessary minor states are implemented.

Register Display

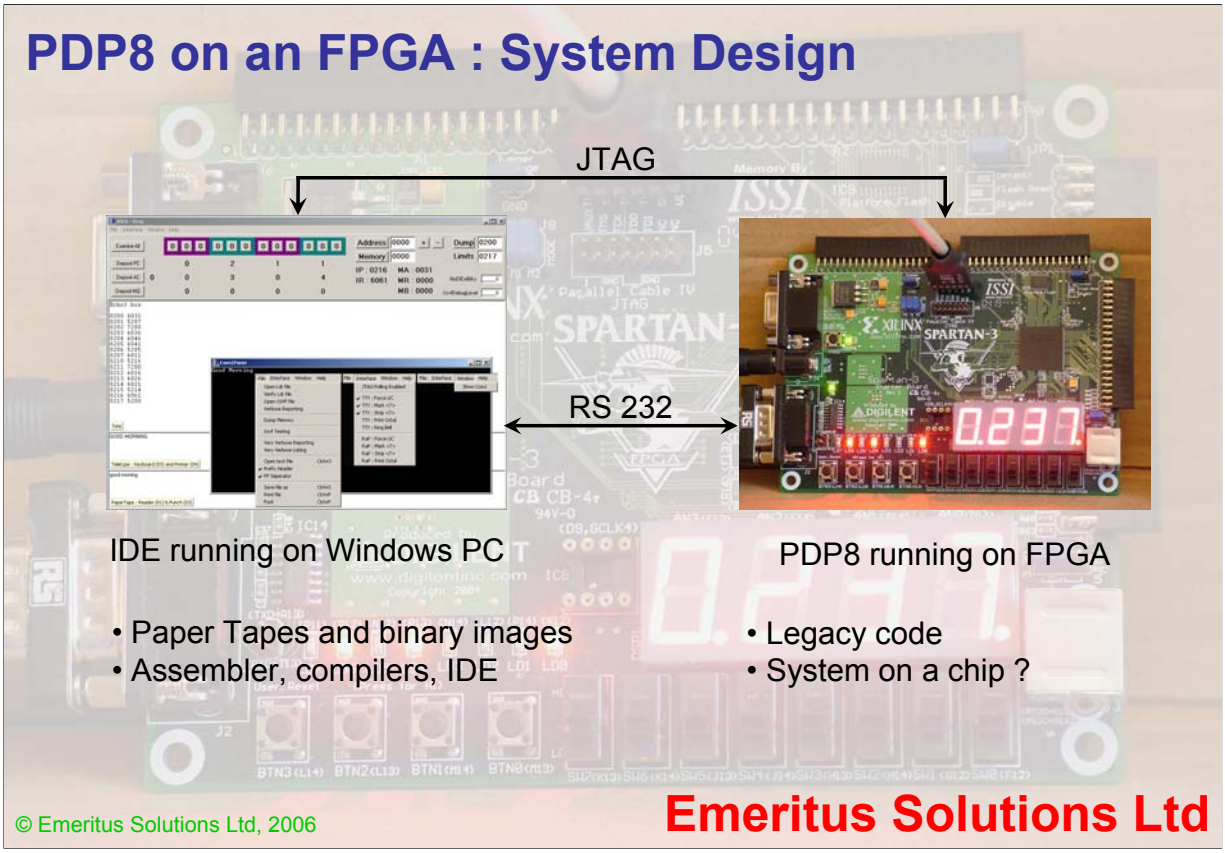
The register display outputs the selected register in octal; except that the link bit is displayed in hex as the 4th bit in the upper nibble. The decimal points output the Skip, IRq pending, IRq enable and Uart IRq enable bits.

Register Select Slide Switches

The register displayed is determined by the last register selector switch transition or takes the reset default. In general the combinatorial and clocked register values are both available, for use in design verification. An indication of the current register is not provided – if you can't remember, you slide the switch.

Push Button Switches

Reset of the PDP-8 to it's default state is effected by the left hand push button. The FPGA has another reset push button (out of view) which resets the FPGA and attempts to load it from flash memory. The other push buttons perform their annotated functions, the speed push button cycling through the available speeds – the default is the fastest.



The System

The PDP-8/E runs on the FPGA, it could boot the FPGA from Flash memory and run.

The IDE runs on a windows PC, communication with the PDP-8/E on the FPGA is entirely over JTAG. This link provides: virtual peripherals; load, verify, dump and peek / poke access to memory and registers.

An RS 232 link, for example, is available from the PDP-8/E on an FPGA. The connection to the IDE is simply for convenience in testing. The 232 uart is “modern” and has a FIFO between the PDP-8/E registers and the UART.

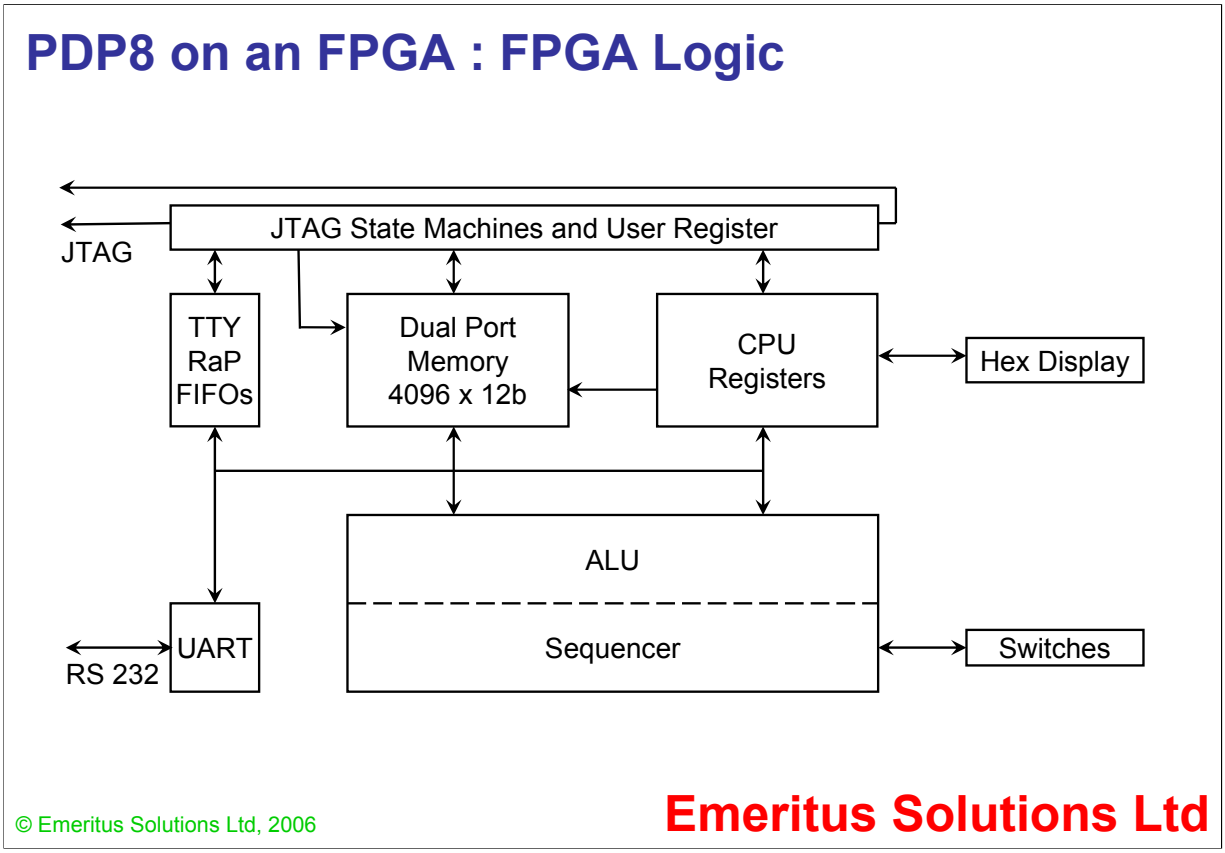
Applications

Running legacy code – maybe not for a PDP-8, but most processors can be FPGA’d.

System on a Chip – new applications can be based on this core.

Support Tools

The original operating software, compilers, assemblers and loaders for the PDP-8 are available on the internet. Also, several PC simulations of the PDP-8 are available, and a freeware PC cross assembler is available on the internet



JTAG

JTAG provides a simple, customisable link between the PC host and the target FPGA. Use of the Spartan 3's user register permits the definition of a custom scan path. For the PDP8 on an FPGA design the user register provides access to virtual peripheral FIFOs, the PDP-8's 4 ki Word memory, and the CPU registers.

The physical PC to JTAG interface can be effected over a printer port or via a suitable USB adapter. The PC driver software is readily available from the internet.

Dual Port Memory

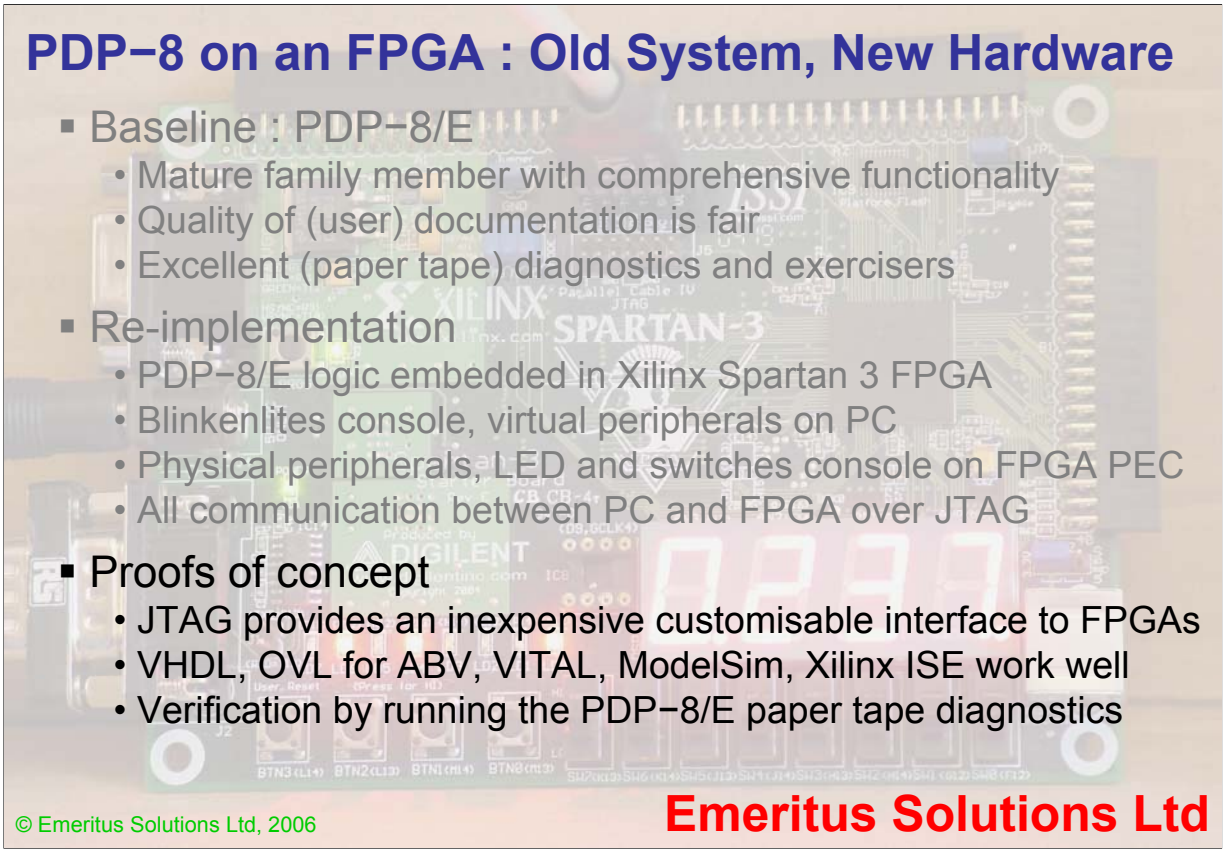
The FPGA has a considerable quantity of on-chip, fast, dual port memory. This design used only a modest part of the available memory on a small FPGA. The dual port memory structure was used to simplify memory interfacing, and to provide a baseline implementation to validate its use. The design of an arbiter to share the interface used by JTAG with another bus master would not require a great effort.

Sequencer

The sequencer which drives the ALU comprises a few synchronous state machines.

ALU

The ALU, which passes the PDP-8/E CPU diagnostics, is based on an unoptimised behavioural description of the PDP-8/E instruction set architecture.



JTAG

The PDP-8 on an FPGA offers a convincing demonstration that JTAG can provide an inexpensive customisable interface to FPGAs. The ability to run virtual peripherals over JTAG, together with memory and register access deliver useful hooks for every stage in the project life cycle. All this for no marginal cost.

Xilinx Toolchain

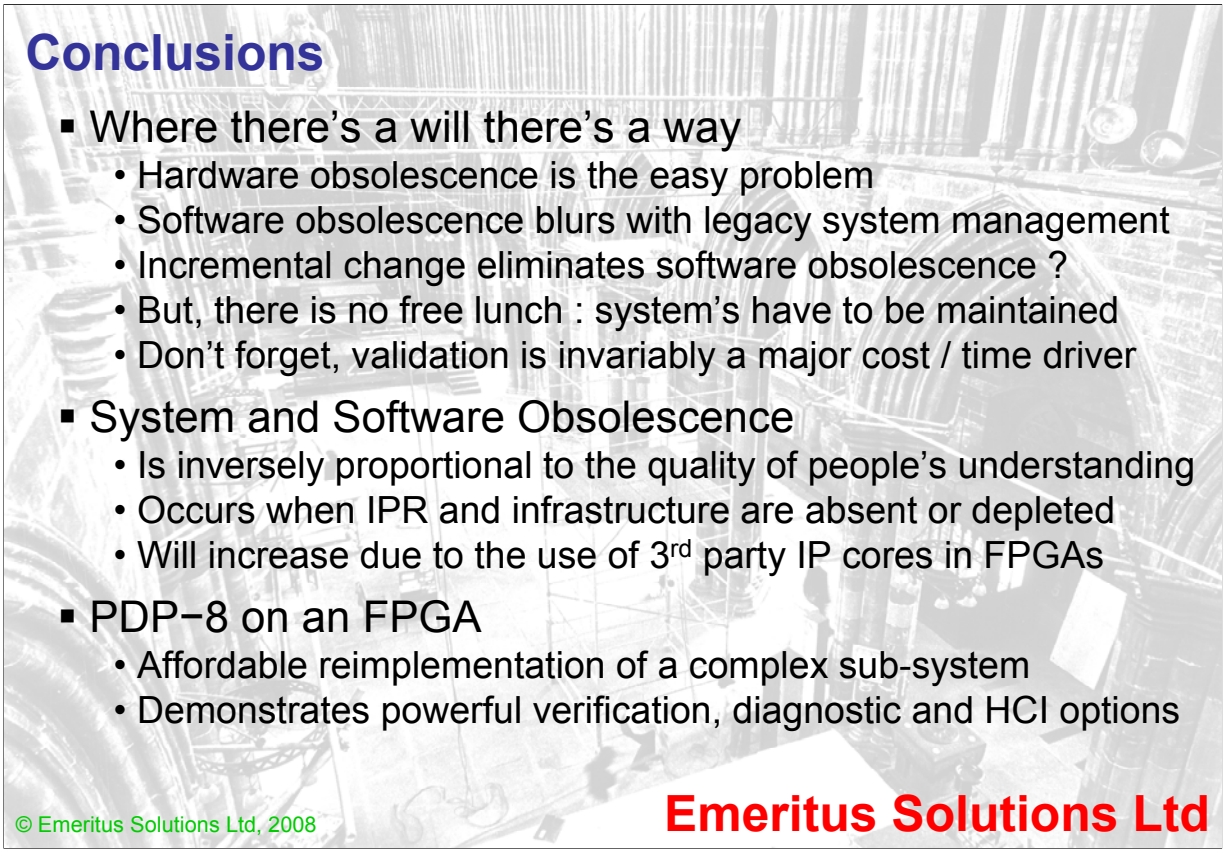
The Xilinx toochain was wholly satisfactory for the PDP-8 on an FPGA. Confirming that fairly substantial FPGA projects can now be performed without “investment”.

PDP-8/E Paper Tape Diagnostics

The Paper Tape Diagnostics for the PFP-8/E are almost certainly the true heroes of this tale, as they permitted the testing and validation of the PDP-8/E on an FPGA..

Lessons Learnt

A key lesson (which we knew already) is the critical importance of accessible, usable, valid IP, documentation and media when working on (obsolete) systems.



Where There's a Will There's a Way

However, averting software obsolescence is like painting the Forth Bridge – an ongoing project. Ongoing software updating with occasional platform refresh is if affordable or necessary a good option. If platform refresh from the usual channels is impractical, a re-design of the hardware is demonstrably an option. Significantly, validation invariably a major cost driver. Validation is likely to be significant cheaper and less risky for a new hardware platform, rather than porting software and interfaces to a new platform.

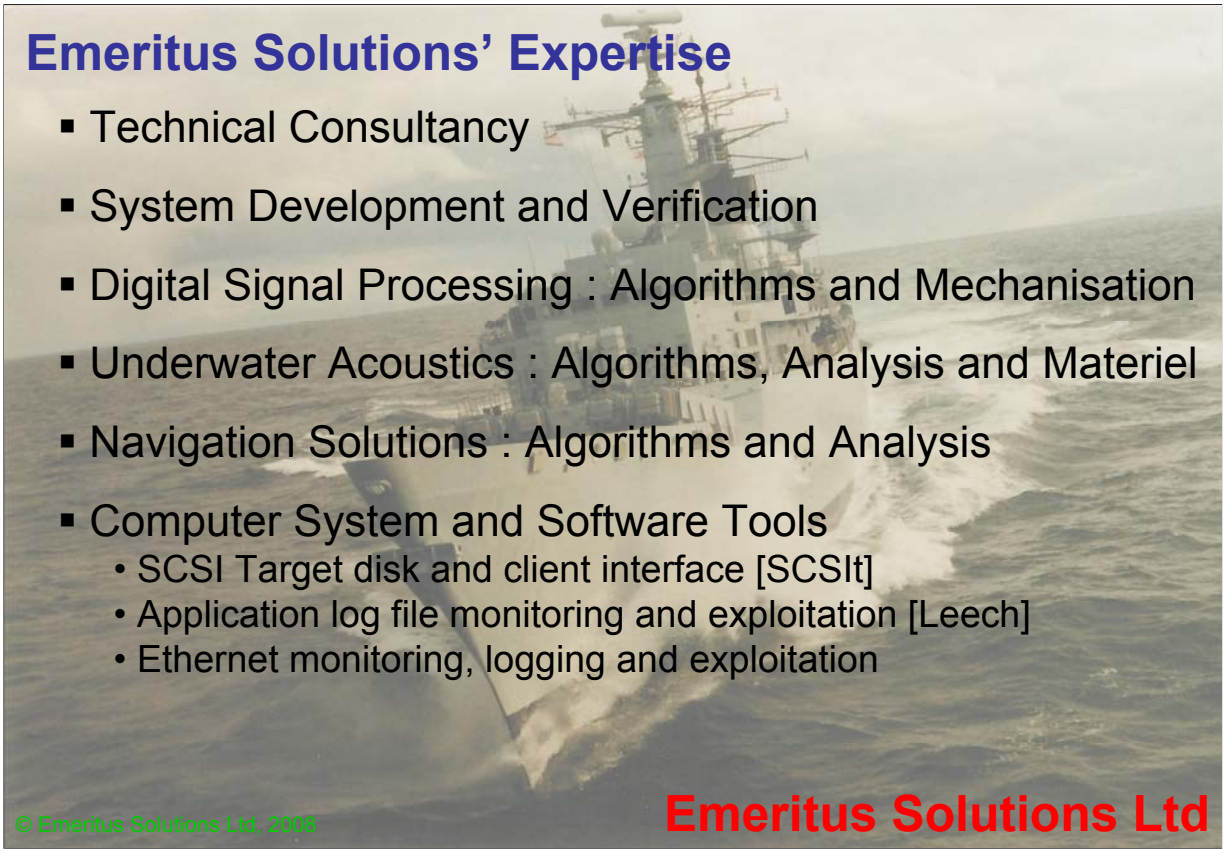
System and Software Obsolescence

Knowledge is the key to averting system and software obsolescence. However, if a frozen system is not acceptable someone has to comprehend it fully. There is no cheap software maintenance, only expensive damage by bargain personnel. Software maintenance has to be performed within the limits of people's expertise.

Long term support of 3rd party IP cores has the potential to be a (new) source of obsolescence within a few years. A range of issues will doubtless emerge from cores being retired, through old cores being incompatible with new tools, to licensing issues. The only reasonably certain strategy is to obtain the VHDL / Verilog source with adequate IP rights and to confirm that it is not brimming with logic comprising FPGA (vendor) specific primitive elements.

PDP-8 on an FPGA

Demonstrates both what can be done, although much more complex requirements could be fulfilled, and what is required to fulfil them. Reflection on how many projects have left as rich a legacy as the PDP-8 give one cause for pause. Perhaps there is no alternative to re-inventing the wheel, but that is a much riskier path than re-implementing a capability.



Emeritus Solutions' Expertise

Technical Consultancy is frequently the mechanism by which Emeritus Solutions' expertise is initially accessed. Specimen subject domains are: Oil and Gas, Sonar, Navigation, Communication and Computing.

Equally, system development and verification whether from a clean or well smudged sheet of paper are core competences with expertise available to identify current and necessary conformance to requirements. Typical systems will commonly include a mixture of PCs, embedded processors, microcontrollers, FPGAs and Digital Signal Processors (DSPs).

In DSP (Digital Signal Processing) Emeritus Solutions has extensive expertise in algorithm design and implementation for time series processing. This expertise encompasses filter analysis, design and mechanisation, complex signal (I+Q) representations and hetrodyne design, spectrum (fft / narrowband) and correlation (broadband) processing, non-linear processing, beamforming, signal measurement and of course signal conversion, transmission and A to D & D to A conversion.

A profound understanding of underwater acoustics, sonar systems and algorithms is available. Our expertise encompasses a broad spectrum from propagation, through sonar equation evaluations, to system implementation and grooming.

Our navigation expertise is based on a profound understanding of underwater tracking systems. Specifically, Emeritus Solutions has sharp tools for the analysis and implementation of trilateration tracking solutions. Our tools for the analysis and synthesis of tracking systems while tailored to sub-sea requirements are applicable to the analysis of GPS, Loran-C and mobile phone systems. Additionally we also have extensive experience of GPS and highly accurate clocks.

In Computer Systems our expertise extends from machine code to productivity tools.

Finally, in Electronics and everyday engineering our expertise is of course extensive.

Biography

Eur Ing Dr Martin Bishop is a Chartered Engineer and a Fellow of the Institution of Electrical Engineers. Martin is an Engineering Consultant with expertise assimilated during over two decades of involvement in defence R & D with the MoD, ultimately as a QinetiQ Fellow, and more recently in Offshore Oil and Gas. His current interests include Digital Signal Processing (from algorithms to implementations), platform design (from FPGAs through HDLs to digital hardware), embedded system design (from DSPs through microcontrollers to OOLs), underwater positioning (using trilateration techniques), underwater acoustics (from propagation to signal and system design), RF processing (including RFID) and the myriad uses of PCs.

Contact Details

Eur Ing Dr Martin Bishop CEng FIEE
Technical Consultant
Emeritus Solutions Ltd
22 Herringston Road
Dorchester
Dorset
DT1 2BS

+44 (1305) 262806 Tel / Fax

Mjd.Bishop@Emeritus-Solutions.com

www.Emeritus-Solutions.com