Table LINC Family Characteristics

	LINC	LINC-8	PDP-12
Project start	~1961	summer 1965	6/67
First ship	~~1964~~ 3/62	8/66 (demo)	6/69
No. produced		143	~1000
withdrawn	~50 (21 by DEC) 12/69	12/69	6/75
Price (min.)	$43,600	$38,500	$28,000
Key Designers	Wes Clark (Lincoln Laboratory) and Charles Molnar	Wes Clark, Dick Clayton (DEC)	Dick Clayton
Goals, features	Complete system for laboratory user... including file system and crt's	cost, + PDP-8 software compatibility, use 8 components	Larger scope, Bus compatible with 8/I.
Size.	69 x 32 x 30 + Tape, Keyboard and consoles	69" x 32" x 33"	76 x 35 x 33
Mp-Pc accesses. (Mega)	0.125	0.67 (PDP-8 memory)	0.67 8/I memory.
Weight	?	?	700
Power (watts)	1000	2000	< 2000
CRT	~~5"x8"~~ 2 - Oscilloscopes	2 - 5" scopes	1-7"x9" (10")

~~Part~~

Part III. The First Minicomputer

This is a separate thing.

This benchmark makes extensive use of the byte and compare features in the instruction set.

Table 3.3 shows the execution times on the LSI-11, CMU-11, and PDP-11/40 for each of the four benchmarks. From these results we see the CMU-11 is approximately twice as fast as the LSI-11 and 63% of the speed of the PDP-11/40. As expected, there is a moderate amount of variation in the relative performance of the three machines for the different benchmarks. The two dominant effects that can be seen in Table 3.3 are that the PDP-11/40 design has optimized register-to-register operations more than either the LSI-11 ~~of~~ or the CMU-11 (as demonstrated in the partial differential equation benchmark). Byte operations are more efficiently performed in the CMU-11 because of its byte-swap data path provided by the D and I buses. The last line in Table 3.3 is the data published by O'Laughlin [1975] in an article comparing the different DEC PDP-11 implementations.

It is mildly disappointing that the CMU-11, built with Schottky TTL bit-slices could not equal the performance of the PDP-11/40, built with standard TTL circuits. The next two sections will examine in detail where performance was lost (and gained) in the CMU-11 design. Before continuing with this review of the design, we turn to a brief discussion of the cost of the CMU-11.

A principle objective of the 3000 microcomputer bit-slice packages is to simplify the design of processors like the CMU-11. Table 3.4 is a summary of the complexity (measured in integrated circuits) of the CMU-11. There are two columns in Table 3.4: a simple count of the number of integrated circuit packages used in the CMU-11, and a column that converts the design to "16-pin equivalent" packages ~~to~~ to give a measure of the board area ~~a measure of the size of the design in a standard unit~~ required ~~to build~~ to build the processor. Table 3.6 gives a breakdown of the actual cost of the CMU-11 at January, 1976 prices.

CONSTRUCT

LINC (Laboratory Instrument Computer)

The LINC was designed at the MIT Lincoln Laboratory ~~and MIT~~ in collaboration with the Communication Biophysics group of the MIT Research Laboratory of Electronics. (Clark and Molnar, 1964)

~~Since the LINC was designed outside of DEC, it doesn't precisely fit the constraint as being about DEC's computers. However, since the~~ ~~because~~ The LINC was initially built from DEC Systems Modules, and strongly affected the PDP-4 and PDP-5 designs (see Ch. 00), ~~we feel somewhat justified. A stronger tie is perhaps that~~ ~~its inclusion is somewhat justified~~ Additionally, DEC went on to manufacture the LINC directly from the LINC (see Fig. LINC) design. Subsequently, Wes Clark with Dick Clayton designed ~~both~~ the LINC-8 (see Fig. LINC-8), a two-processor machine (LINC + PDP-8) which executes both ISP's in parallel; ~~and~~ Dick Clayton designed the PDP-12 (see Fig. PDP-12), a single physical processor that executes either PDP-8 or LINC instructions sequentially (by switching modes).

Insert A, next page

~~Also LINC affected the PDP-4 design as we discuss in the 18 Chapter 0.~~

Some of the characteristics of the LINC Family ~~machines is~~ are given in ~~the following table.~~ Table 00. ~~(Table #)~~ Note that the size has remained essentially constant at one cabinet over the computer's life.

switch positions w/ next #1

Packages, Power, Interconnection, and Consoles.

Changes in Organization

1. Packaging ~~P~~ (Boxes and Cabinets)
2. Interconnections: Modules and Backpanels.
3. ~~Inter bay~~ ~~Inter pa~~ Inter backpanel inter
4. ICs (interchip)
5. Power.
6. Consoles.

Testing, Design, Standardization, Metrics, Evolution, Design Aids, etc. (machine readable serial #)

Two: Gonzales & Schalke. | non-verbal, but he knows. Others are verbal, don't know anything. or do anything.

Others are beaucrats

Pile layers of crap: Committees

Table LINC Family Characteristics

	LINC	LINC-8	PDP-12
Project start	1961	summer 1965	6/67
First ship	3/62 (demo)	8/66	6/69
Withdrawn	12/69	12/69	6/75
No. Produced	50(21 by DEC)	143	1000
Price (min.)	$43,600	$38,500	$28,100
Key designers	Wes Clark & (Lincoln Lab and MIT) Charles Molnar ~~(Lincoln Lab)~~ (AF Cambridge Research Lab.)	Wes Clark (MIT), Dick Clayton (DEC)	Dick Clayton
Goals, features	complete system for laboratory user. (including file system and CRT's)	cost, + PDP-8 software compatibility, use 8 components	larger scope, bus compatible with 8/I
Size	69 x 32 x 30 +	69 x 32 x 33	76 x 35 x 33

LINC DEC-8 PDP-12

	tape, keyboard, and console		
Mp-Pc accesses	0.125	0.67	0.67
(Mega)		(PDP-8 memory)	8/I memory
Weight	? fill in wt.	?	700
Power (watts)	1000	2000	< 2000
CRT	2 - ~~oscillo~~scopes	2 - scopes	1 - 7" x 9" ~~(10")~~

? where does this go? Could it be LINC?

~~Have this ¶ follow 1st LINC ¶ OR use as concluding ¶ to section starting "The LINC ..."~~

A insert

~~There are other reasons to include the LINC as a member~~ Family ~~s of the DEC 12-bit Family. The LINC strongly influenced other PDPs (including PDP-4 and -5).~~ The LINC tape unit and system ideas permitting a user to have personal files has been carried over directly in the DECtape design and programs. This device made the complete personal computer for the researcher possible at a reasonable economic level. ~~We have also included the LINC because the paper is especially well written, even though it does stress applications and minimally discusses the construction, structure, and other engineering aspects of the design.~~

This should be at the beginning of the LINC section where you provide rationale for including it. Follow with ¶ some charact... are given OR at end of section

The systems aspect of LINC (circa 1962) were really quite significantly ahead of their time ~~at DEC,~~ and ahead of other systems including the timesharing work in terms of providing a clean, simple, yet powerful console (two CRTs) and file system interface (Fig. 2). Note the DECtape library system came two years later on PDP-5. ~~We~~ DEC had been stressing design flexibility and modularity and provid~~ed~~ing many ways for interconnecting computer components together in order to have any type of structure. Also, LINC is quite constrained (only 1 Kword or 2 Kwords of primary memory were available) with two LINC tapes and two CRTs.

this detracted from having a base system configuration complete with software.

Most all other characteristics including the I/O interface were quite constrained. By bounding the system in this way, it ~~is~~ was possible to provide a complete environment including software.

console. This was the machine's only patent. Ultimately, the logic was too complex; when the console failed, it was harder to fix than the processor.

Improved Priority Interrupt Latency

We were not able to reduce interrupt latency to the two microsecond goal. With parity, memory protect and relocate options implemented, and with adder and synchronizing delays, a four nsec. time was achieved, which was acceptable in the system.

One Cabinet Package

The CPU, I/O processor, console and 32 Kword memory were packaged in one cabinet, but it was close. There were virtually no spare module slots in the CPU.

I/O Bus Length

The I/O bus was 75 feet by switching to a terminated bus. The PDP-9 used an unterminated transmission line with diode clamps. The PDP-15 has cleaner signals and no reflections. A new set of bus driver/receiver modules were designed which were faster and

(shown in Fig 0, page 00)

DEC 338 DISPLAY COMPUTER

The 338, (a variant of the PDP-8,) ~~is included mainly~~ might have been included for its historical importance (see Chapter 25, Bell and Newell, 1971) ~~reasons and because it is a~~ variant ~~of the PDP-8 Family~~. ~~It~~ clearly shows how the application need drives a complete change in hardware ~~so as~~ in order to interpret the necessary data type (a graphic picture). ~~We believe~~ it is one of the very earliest display processor based computers--if not possibly the first. ~~The question of being first is clouded by the state of the evolution as we discuss below.~~ necessary?

The 338 idea was extended and applied to PDP-9 (in the form of the 339), ~~and then~~ ~~greatly~~ extended ~~in~~ for the PDP-15 ~~'s~~ VT15 display processor, and also applied to the PDP-11 series of display processors. Although the 338 has the right general capabilities, it doesn't did not have the refinements that later display processors for PDP-15 and PDP-11 ~~have~~ had (GT40 and GT60). ~~Also, there is a question of having a balanced computer.~~ Since Because the display processor is ~~so~~ relatively expensive, ~~then the host processor~~ the cost of the host processor should be adjusted accordingly: it should not be a performance bottleneck, nor ~~must it be~~ need it be significantly cheaper than the display part. ~~The DEC display processors have tended to extend the 338, whereas there have been departures from this structure (CMU thing?).~~ Transition doesn't follow ? What kind ?

An observation that display processors and ~~perhaps~~ other specialized processors evolve in a fashion ~~akin to~~ called the wheel of reincarnation (Myer and Sutherland, 1968) as diagramed in Figure 3. The Figure ~~Figure 3~~ shows how one starts with a very simple, basic design (here) to have graphics picture output for a computer. The ~~project~~ trajectory around the wheel follows.

Point-plotting

1. The computer includes a single instruction display controller which can plot a ~~point~~ picture on a point-by-point basis under command of the central processor. For most all displays, except storage scopes, the processor can barely calculate the next point fast enough to keep the scope refreshed. Hence, the system is processor bound, and the display may be idle. The original PDP-1 Type 30 scope is typical of this position. A scope of this type is offered on most DEC minis.

Such scopes (e.g. VT50) are not integrated with the computer, but are connected via a serial communication link.

Vector-plotting

2. By adding the ability to plot lines (i.e. vectors), a single instruction to the display processor will free some of the processor and begin to keep all but the fastest scope busy.

Character-plotting and alphanumeric

3. ~~Having realized~~ With the realization that characters are mainly displayed, ~~then~~ commands to display a character are added, further freeing the processor. Many of the point-plotting displays were extended to have character generation plotting. The alphanumeric scopes have this capability ~~(less~~ and usually do not plot vectors; ~~plotting)~~. Special scopes such as the ~~VT8/E an~~ VT8E and VT14 have been added to PDP-8 to ~~do this~~ carry out this function. ~~Similarly~~ in a tightly coupled fashion

~~The Conventional alphan~~

4. In reality, a picture doesn't consist of just characters and vectors; ~~but~~ each element of the picture is actually a string of characters to be displayed at a particular point and a set of closed or open polygons to be displayed starting at a particular point. By providing the control display with a Direct Memory Access channel, it (? antecedent?) can fetch each string of text and generate polygons with no central processor interaction.

General figure and character display

Display processors

5. Now with the ability to put up sub-pictures with no processor

intervention, it is easy for the whole picture to be displayed by linking the elements together in some fashion. This merely requires jump and subroutine call instructions ~~(~~so that common picture elements don't have to be redefined~~)~~. ~~(This is roughly the capability of~~ the 338 and other ~~most~~ display processors.~~)~~ ~~why parenthetical?~~ have roughly this capability

Cap ? ital.

Integrated display and central processor are

6. Now, all the data paths and state ~~is~~ present for a fully general purpose processor~~,~~ so ~~such~~ that the central processor need never be called on again. This requires a slightly more general purpose interpreter. There is full generality by minor perturbations, and the processor design can be refined in such a way as to execute the same ISP as the original host because the cost of incompatibility is too great. Two processors require two compilers, diagnostics, manuals, and support for use. This state ? ~~Finally, we~~ ~~returned~~ provides ~~have arrived back to~~ the same capability ~~as we~~ ~~from which~~ as the initial starting point 1. ~~started~~. The original processor is completely free, and there is a display processor with the capability of executing both the original ISP plus the display ISP.

~~The~~ Two ~~processors for~~ computer structures

7. Alternatively, ~~we can take this~~ the processor ~~and~~ can be isolated ~~it~~ as a separate computer ~~which we~~ and reconnected in some fashion ~~back~~ to the original's central processor-primary memory pair. Such a structure is just a basic computer plus ~~position four~~. state 4.

offered less bus loading. The penalty paid was higher power consumption and greater power supply cost.

Minimum Multi-User Protection Scheme

The least expensive multi-user protection scheme was provided in the form of a relocation register and a boundary register. Because it was marketed as an option, it degraded the machine more than necessary. However, the minimum machine cost was kept within the target.

Better Maintainability

A means to look at more than 400 signal points was provided in the logic. This, coupled with the single step feature, allowed troubleshooting of the machine from the console without an oscillisope. As it turned out, this feature was used infrequently because of the required training.

PDP-9 Compatibility

PDP-9 instruction compatibility was achieved, but with three minor exceptions. These caused no customer or DEC software

THE CACHE-BASED PDP-8[1]

~~Chapter 7 describes~~ a ~~design for a~~ PDP-8 with cache was designed and constructed by Professor David Casasent at CMU. Initially, the project was proposed as exploratory

~~The cache~~ (Bell, Casasent and Bell, 1974; Casasent and Bell ?)

advanced development on the desirability and feasibility of using Emitter Coupled Logic (ECL) logic for designing ~~very fast, minicomputers~~ fast computers (including PDP-10s). As the investigation proceeded, the need for a large, ~~very~~ fast memory emerged. Such a memory turned out to be so costly that ~~such~~ a computer would not be feasibly marketed ~~feasible from a market standpoint.~~ The easiest way to build a cost-effective, fast minicomputer turned out to be using a cache structure in order to reduce the cost of primary memory. Also, instead of designing a very fast PDP-8, which ECL logic would have provided, the goal was modified to be ~~more modest~~ less fast, much less costly and potentially marketable. This caused TTL Schottky ~~was~~ to be used in the design, even though the logic family was just beginning to evolve.

In order to make the prototype more nearly marketable without ~~a~~ completely redesigning it ~~redesign~~, the project ~~design~~ was ~~also~~ constrained to ~~be compatible with most of~~ use the 8/E Omnibus backplane and parts. The prototype did not become operational as quickly and cleanly as possible, and was therefore not used to stimulate a market. ~~hence the use of the prototype to stimulate a market did not occur.~~ Throughout the development of ~~All through~~ the design there was little interest in its ~~the~~ marketability ~~of the design~~. It was carried forward in order to test the cache applicability, circuitry, ease and techniques for building a fast 8. As is true with ~~in~~ most yet to exist ~~new~~ products, the fact a market is not there is clear. The logic: ~~We~~ were not losing sales to very fast, non-existent minicomputers, hence there is ~~was (and is)~~ no market for very fast minicomputers. Furthermore our customers are of little or no help because, unless faced with the widescale availability of significantly faster ~~such a~~ machines, the applications can't be defined and ~~generally don't~~ emerge.

1 This structure uses a small fast memory to hold localized primary memory references. The structure ~~It~~ has been subsequently used in the KL10, 11/70 and 11/60 ~~KL10 (chapter 00) and PDP-11/70 (cho~~ (see chapters 0, 0 and 0).

A number of things were learned from ~~doing the work~~ the research. ~~It~~ The Cached-8 did influence the 11/45 although the designers rejected the cache notion (Chapter 00) ~~since~~ because they believed that the cache concept clearly didn't not apply to PDP-11 program behavior. Eventually Strecker persevered and convinced other 11 designers to build a cache-based larger scale minicomputer 2 and this became the 11/70. Fortunately, the 11/45 processor was designed to operate with all bipolar, fast memory. The ~~paper~~ work "Alternative Cache Organizations" illustrates the use of the 8 as an experimental vehicle for understanding a design in terms of program behavior. It also allows one to measure the time flow of ideas from research, advanced development, and finally ~~into~~ to production of machines. It also shows how, by rather minor perturbations, a design can be kept compatible with its predecessors while providing rather dramatic performance and performance/cost ratio improvements. as shown in ~~the~~ Table 0.

~~What's the difference?~~

We decided that a block size of 1 was adequate, given the PDP-8/E 1 word memory width constraint. The design had 512 words ~~or~~ with no separation of the cache ~~for bet~~ between data and instructions. ~~The results~~ A 100 nanosecond PDP-8 with 512 cache ~~perfor~~ had the ~~following~~ characteristics shown in the following table. (use from P351 of the attached paper)

Improve Producibility

Margin testing of the PDP-15 was planned using a combination of varying logic timing and temperature. Special test equipment was provided for the production line to allow rapid head cycling of CPUs and memories. In addition, a fast program loader system was designed around a PDP-8 with multiple DECtape units. Diagnostic programs could be loaded into the memory of a unit being tested by merely pressing a button. This saved considerable time in computer check-out.

Subassembly replacement was planned at the final assembly and test group. The concept was that if a processor, memory, power supply or whatever failed to work when it was integrated into a system, the whole subassembly would be replaced and sent back to its appropriate test line rather than doing the repair in the area. This process proved impractical, as the production line was never filled with enough material to allow this to take place.

Early First Customer Ship

The first PDP-15 was to be shipped eighteen months after commencement of the project. We ran into a number of difficulties during the project, including personnel turnover,

The PDP-14 was designed expressly as a controller for complex electromechanical machinery (such as transfer machines, conveyors, simple milling). The problem first originated at General Motors ~~as~~ via the need for a controller for a large, transfer machine which handled engine blocks by a sequence of operations (transfers). ~~Since the first implementation, it often replaced or supplemented large relay sequential circuits.~~ The notion of a controller evolved from the use of standard DEC K-series industrial modules, (see Chapter 00. for sequential circuits. The K-series modules provided increased reliability ~~over the~~ and replaced electromechanical control (i.e. relays) components by using solid state sensing and switching. The 14 represents a further cost reduction over the K-series modules by using time-multiplexing so that one

control structure (in memory) could serve as the interconnection (and processing) structure versus physically interconnecting the modules together. Such a system is shown in Fig. 3A.

The Basic Design Problem

Such a controller should:

1. be lower priced (with lower life cycle costs) than existing control alternatives--and ~~it~~ should ~~be~~ easy to operate;

2. solve the control problem and be programmable by users who have solved problems using a different representation (e.g., relay ladder diagrams);

3. operate in the high electrical noise environment;

4. operate in the poor physical industrial environment characteristic of the machine it controls;

5. have the appropriate I/O interfaces to sense contacts and ~~output~~ to control power relays.

Although a PDP-8/I might have been programmed to carry out the task, it was either not considered or else rejected because:

1. the cost was perceived to be too great;

2. there was some perception that a conventional stored program computer could not solve the problem;

3. the circuits did not appear to have adequate noise margins to operate in the environment;

4. the physical environment constraints (heat, dust, liquid, etc.) couldn't be met;

5. there was inadequate I/O.

Even with little arithmetic capability, the machine structure and processor state ~~of the machine~~ is roughly equivalent to a PDP-8 design. Ultimately, the ~~state~~ processor state was extended beyond that of a PDP-8, but ~~this aspect~~ these extensions will not be discussed when communication was required with host processors as the problem changed (eg.

The Design

As a result, the PDP-14 (see Fig. 4) was proposed and designed expressly to solve the problem and cost less than the 8/I which was just going into production. For example, the 14 had no data operations except on a single boolean value using a 1-bit accumulator called TEST. ~~Indeed, it did not even have a subroutine CALL instruction.~~

1. In order to solve the boolean equations that a conventional relay controller solves in parallel, the 14 must solve equations sequentially at a rate of approximately 30 hz, giving the illusion (which is quite adequate) that the equations are solved in parallel. The 14 has instructions for taking data from input sensors, solving the equation, and outputting the result to a relay.

2. The circuit logic was slowed down in order to have improved noise margins. It was felt that core memory did not have adequate noise susceptibility; therefore, ~~hence couldn't be used.~~ A braided, read only memory was used. hyphenate?

3. The unit was housed in a dust proof enclosure.

4. Appropriate I/O interfaces were designed.

The ISP of the PDP-14 is given in Fig. ~~x~~ 5. From the ISP description, ~~we~~ note that it is the smallest, most trivial ISP that one would find. Technically, it ~~must~~ could be called a computer because it can perform

computation in the same way a Turing machine can -- without an arithmetic unit. However, It encodes the Boolean data operators for which it was designed more efficiently than nearly any other computer--provided the equations are simple.

There are four instructions to take values from input switches or the current state of the outputs (relays), and to compute new output values (the right hand side of a boolean equation). Therefore, it also can simulate a Sequential Machine (state diagram or flow chart). Two instructions sense the value of intermediate results (stored in TEST) and are used ~~so as~~ to eliminate the need to completely evaluate an equation each time. Two instructions set output values either to 0 or 1.

There ~~is one~~ are four instructions, Jump, Skip, Jump to Subroutine (a single level) and Return from Subroutine, to direct program flow.

Another way to understand the 14's operation is to start with the application. Figure ~~X~~6a shows ~~how we start with what is~~ a combinational, relay logic network that ~~simply~~ evaluates a boolean expression (in parallel). When this is network implemented with the 14, the inputs and outputs are simply connected, and the program forms the interconnection (Fig. ~~X~~6b). Figure ~~X~~6c gives the boolean expression for the network in Fig. ~~X~~6a. To ~~implement (evlauate)~~ evaluate this equation using a 14 requires a sequential program (see Fig. ~~X~~6d). This program requires between 120 microseconds and ~~180~~ 200 microseconds to compute ~~an~~ the output value, ~~y6~~ y8, since each instruction requires 20 microseconds.

It should be clear to the reader that as the number of equations to solve increases, the time to solve the set decreases. Since heavy duty industrial control relays typically

operate at a 30hz rate (33 milliseconds) and each equation with 4 terms takes approximately 150 microseconds, then 222 such equations can be evaluated by such a system. Also note that this equation required twelve locations; and hence, a memory full of such equations requires about 2.5 Kwords.[1]

[1] Here, it is instructive for the reader to compare this memory need with that implied using Amdahl's constant ~~we discuss in~~ (see Chapter 00).

A number of 14s were built and installed in the intended applications over the period 1970-1972. Programming was carried out in languages supported by compilers that operated on PDP-8. The languages allowed users to write:

1. ordinary assembly programs (resembling PDP-8 programs);

2. directly as a set of boolean equations;

3. express ladder diagrams (in effect, these are a set of boolean equations);

4. a program as a flow chart; i.e., as a sequential machine that goes state by state and tests and branches on various input values to create output state. This permits both combinational (boolean equations) and sequential circuits to be implemented;

5. The user could also simulate the behavior.

The Evolution

As the 14 and its competitive, contemporary machines were used, a demand for a second generation controller was created. By 1972, the additional requirements included:

1. better cost and speed due to competitive similar products, including the companies whose relay products the 14 displaced;

2. ~~There were~~ many more functional demands including ~~having~~ an ~~easy to change~~ easily changed read only memory with the ability to load programs via a communications line or connected console. In addition, the controllers should connect in a network fashion and report back status and results to a supervisory computer at the next level of a hierarchy. The controller should also be capable of recording events ~~(e.g.,~~ including counting the number of parts it processed and. It needs timers which ~~can~~ could be ~~set~~ used as part of the control equations. ~~The controller needs to keep track of whether a part of the machine is full, or empty (i.e., a shift register representation is used).~~

shift register (queue) representation of objects

TRANSITION

3-5. The unit should operate over an even wider environmental range, with a more complete set of I/O.

Thus, the PDP-14/30 evolved (see ~~Photo Y~~ Figs 7 and 7a). The initial core memory was replaced by an 8 Kword core memory. In this way, the programs could be easily changed rather than having to be returned to DEC for manufactur~~e~~ing. ~~Since~~ Because the original 14 was so slow compared to its inate, circuit capability, the instruction time was reduced from 20 microseconds to 2.5 microseconds to have better frequency response and to handle a larger number of equations. ~~A primative subroutine calling instruction was added, giving it one level of subroutines by using a single register to hold the caller.~~ ~~Since~~ Additionally, because a large number of special registers were added to hold numeric values (the shift registers, timers and counters), an arithmetic unit was added to ~~it~~ the 14/30 in an ad hoc fashion. All these additions forced the ISP ~~op~~ to change. The 14/30 extensions were unable to be made in such a way as to have binary compatibility; thus, ~~the~~ software changes were also required. Another critical part of the 14 includes ~~a~~ the VT14 for a programming, program load and observation console. ~~the VT14, which~~ This consists of a CRT and PDP-8 mounted in a portable housing. Since the 14/30 can also report ~~back~~ the status of its input and output variables, the VT14 also has the ability to display the status of the ladder diagram (i.e., ~~whether a relay is on or off and the contact position~~ relay and contact position). The display of the screen is shown in Fig. ~~?~~ 8. At the time when the 14/30 was proposed, I urged that it not built because:

K punct.

1. A standard PDP-8 computer ~~is~~ was cheaper to build, and we would get more production volume and lower costs by not ~~building~~ constructing a special unit. In fact, ~~a~~ the 14/30 ultimately ~~[illegible]~~ required more space than an 8 processor ~~[illegible]~~ module -or
2. The PDP-8 could be extended trivially to have the original 14 ISP. Thus, the PDP-8 ISP would be available to do the tasks that would inevitably evolve to:

a. do self-diagnosis;

b. do more extensive counting and timing functions;

c. deal with non-boolean data (e.g., time, non-discrete events including angular position);

d. handle general control in both the analog and digital domain;

e. communicate with computer networks requiring much better protocol control for intelligent and error-free communication; and

f. use algorithms to encode the control function instead of the relatively large program state with no ability to perform computation.

3. ~~The~~ Since the 14/30 and PDP-8 were built using the same circuit ~~PDP-8 circuits had adequate~~ electrical noise margins of the 8 were adequate. family, by definition the

4. The 8 could be packaged in the proper cabinet for the physical environment.

5. There could be adequate interface parts.

6. ~~Most of the work~~ The 14/30 required an 8 anyway; ~~hence, it was more expensive to have two computers than one!~~ two computers were obviously more expensive than one!

All of these enhancements would have greatly improved the marketability of PDP-8 for all industrial applications.

Although I was unsuccessful in convincing the design group that the 14/30 should become an 8 with appropriate extensions and packaging, some of the 8 parts were used in the 14/30 design. At this time, some of the initial conjectures about the 14 ~~was~~ were correct. The product was eventually downgraded as a mainstream product primarily because it required high, specialized support; ~~hence, dropping~~ thus, not continuing it as a mainline product represented a strengthening of purpose. ~~Control is~~ The Control ~~has evolved~~ problem has evolved to require a computer, even though the 14 ultimately evolved to be a complete, general purpose machine

While it is easy to dismiss the 14 as a trivial special purpose controller that should never have existed (especially when one looks at the poorly evolved 14/30 architecture), this is to say that the problem being solved should not exist. Such a conjecture flies in the face of the adage "that the customer is always right".

The problem is worthwhile and should have been solved with the full generality of a computer. The ultimate state (several next generations) of the controller is clear: a computer is needed to handle the full generality of the problem as it was, as it slowly evolved (when the 14/30 was designed) and as it will clearly continue to evolve ~~to~~. The solution was, and still is, to adapt and evolve a general purpose computer by a combination of hardware instructions and software to be the best controller. Such evolution ~~can~~ ~~easily surpass~~ surpasses ~~any~~ the fixed control structure that ~~would have evolved~~ evolved and was continuing to evolve, not evolving the 8 to the application by just adding ~~the~~ a few special ~~Boolean~~ instructions* 1 given in Fix. X ~~was~~ is the error.

5

or the 11 (and other microcomputers or minicomputers)

1 ~~Left~~ The extensions are left as an exercise to the reader.

The Families

~~The~~ This part on the ~~Mi First~~ First minicomputer includes ~~all~~ the descriptions of ~~all~~ the 12-bit computers that DEC has built and been associated with that ~~e/aff~~ effected the PDP-8 family design.

Figure 2 shows both the family tree of the~~se~~ 12-bit machines ~~and~~ with significant events ~~both inside~~ outside DEC and the corresponding internal response events, that made up the evolution (eg technology use, first ~~ship~~ shipment of a compiler)

The family tree postion shows that there are ~~basically within the extended PDP-8 famil extended family, there are 5~~ five nuclear groupings (or sub-families) within the complete family. There are two distant relative ~~a~~ families, the LINC~~s~~ family — which contributed some ~~the formed some of the~~ base ideas ~~for~~ to the original PDP-5. ~~An~~ and The Industrial Controller Family, PDP-14 ~~series, as~~ which also had a 12-bit word and ~~cert cet certainly bare~~ bears some relationship to the 8 Family. Later members of the 14 used parts from the Omnibus family, ~~for its construction...~~ including the Omnibus and memory components. ~~The LINC family Later members of the 1 LINC~~

Later members of the LINC family were made of up of ~~am~~ major subassemblies physical and software that were part of the 8 Family, and ~~of course~~ the the ~~DEC 8~~ original LINC was built from DEC Systems Modules.

It is clear that the CDC 160 was the first commercail 12-bit computer and as such ~~it se formed the bas~~ contributed ideas and organization to both the LINC and 8 Families, as did the DEC 18-bit Family (as we describe in Chapter). ?

The Procesor-on-a-chip Family is built to~~t~~ally outside of DEC by two other organizations, Intersil and Harris Semiconductors Our ~~attempt to~~ efforts to build the 1 chip computers only lasted for about a year, before the~~y~~ efforts were turned to the PDP-11 family. ~~It is significant to use~~ The existence of this family is particularly significant as it forms the basis for the ~~Computer~~ one-board-computers and the computer-in-a-terminal.

The title is ~~overly~~ slightly presumptuous because the main contribution of the 8 Family is ~~probably~~ the number produced ~~(used)~~ (in use) — the notion that significantly cheaper machines can always be built and applied to new applications — hence the name Minicomputer, for mini(mal) computer. The 8 Family also clearly either directly or indirectly contributed to the name, Minicomputer — not an insignificant contribution in a world that ~~identifies~~ ~~or~~ often must have a simple way (name) to identify a concept. The early history of the Minicomputer, can is more accurately, represented in Fig. 1 which shows the interrelationship of early computers

Packaging

~~The right side of the figure shows~~ part of the
~~Figure~~ The time line figure is divided into basic semiconductor and

module packaging technology , computer languages and computer Operating systems.

On ~~In~~ the packaging ~~aerea~~ and semiconductor line we note there is ~~something of a 3 year~~ a 1 to 5 year delay

~~gap~~ between when an~~d~~ idea is available and when it is applied to a machine

(note the application of Silicon transistors and TTL ~~logica0~~ logic). ~~Ip~~ For example, It is

possible that the 8 could have usesd TTL logic and been ~~evn~~ even further ~~h ahead of~~ out

~~its time~~ on the state of the art line. ~~(It shoul~~ — negating need for 8/I and 8/L). Since the packaging

was different for the 8, ~~it~~ and ~~was~~ done with automatic ~~wr~~ wirewrap machinery,

this change was enough for a single machine. ~~On the other hand, the~~ also, TTL logic, as initially

marketed, was significantly ~~mroe~~ expensive ~~that~~ than the discrete logic circuits that

the 8 was built from. It took five years to adopt the automatic wirewrap machinery

that was firt used in the ~~8~~ IBM 7090 . This ~~is a~~ was partially due to the

fact that ~~DEC~~ it reperesented both a major packaging change and a major ~~expense~~

capital expenditure to DEC (sales were about $10 M then). The various read only and read write memories have

~~are d~~ not been essential to the ~~8s~~ designs, and since the ISP is so simple.

~~, although having a~~ large register file and low cost rom for microprogramming makes th the floating point processor and its

control possible (note FPP versus semiconductor memory availabili~~y~~t in the 8 price range).

Software ~~Several languag~~ Fortran, in 4 Kwords, is probably the most significant

accomplishment in terms of langages for the 8. It was designed for and operated

on the PDP-5. Eventually ~~a~~ both BASIC and FOCAL interpreters were

written to provide better interactive languages. ~~that the single user~~

~~could have for scientifica computation and~~ . These ~~were used are~~ are

used extensively ~~in~~ as a calculator, and for real~~t~~ time control ~~and for~~ including

laboratory experiment control and industrial. ~~Bo~~ The ~~lay~~ lanugaes were both extended for graphiscs display.

The ~~Since the~~ 8 ~~was~~ is also used extensively ~~f in~~ for ~~c instruction, they They Te These~~ teaching about computers and computer aided instruction

~~lanugages were u were used extensively in pedagogical I instructional.~~

In 1970, a ~~la~~ business language interpreter, DIBOL was introduced. Its

roo~~p~~ts ~~go back to~~ include COBOL, but ~~since it is~~ as an interpreter, it was designed

to provided faster response to the writing and debugging of programs

and to ~~be~~ make more efficient ~~with~~ use of memory space. These are excellent tradeoffs for the single ~~user~~ user sytesm whose problems are usually ~~bo~~ disk performance bound vis a vis records accesssing ~~accesses~~. A n especially powerful editor and langage was introduced in ~~1~~ 1976 in the form of WPS 8. for the word processing market It also ~~forms the basis of the barplica~~ is the forcing ~~drive~~ function to have very low cost stand alone computers...in this case the VT78 resulted.

The Operating Systems

~~A similar story can be~~ The operating system line shows that the influence for software came from two ~~corners~~ areas: the original LINC, which was designed to provide a single user with his own environment and the timesharing work which was designed to provide a user with ~~a~~ ~~an apparent computer c~~ private, apparent computer with his own filing and processing system.

~~All the~~ ~~The third force,~~ A recent ~~no~~ product, DECnet, a method of interconnecting computers over greater physical distances, came from the ~~growth & posulation~~ minicomputer posulation growth, which in turn forces the computers to communicate with one another in some form. It is interesting to note, that DECnet 8 is the first application of DECnet within DEC. This has quite possibly occurred because ~~of the simplicity of the~~ basic ~~design and~~ operating system is clean, as such ~~that~~ the complexity of DECnet can be understood and integrated more easily. This much of the basic work of DECnet was derived from the ARPAnet reseearch project

The various single user and real time operatings sytems are quite unequalled ~~on even thoush s theh there has~~ even though ~~there~~ have been comparatively smaller efforts devoted to their desing and implementation. ~~Perhaps more import~~ Another by-product of the single user designs ~~w~~ has been the understanding developed for the creation of the ~~RT-11~~ single user operating system, RT-11, which for the PDP-11. ~~it tends to be teh most used.~~

A system that the one of us (Bell) has an especially attachment for is TSS8, the time sharing sytem for the PDP-8. Only ~~several~~ a hundered or so system and per (per users)

systems were sold, ~~but it rem~~ while technically it is the ~~highest performance~~ lowest cost and ~~best~~ highest performance/cost ~~with the best cost and cost/p~~

timesharing sytem we provide. The reaon for the lack of commercial success,

I ~~hope~~ suspect is simply attributed to the fact that the 11 came along and the devlopment and marketing

effort were redirected ~~was turned to its development.~~ On the other hand, TSS/8 ~~si a complex system~~

which requires some understanding, and the time to sell ~~and~~ such a system

may be greater than we have ~~allocated~~ allowed for (ie it may be underpriced in

terms of software complexity). A major side be benefit of TSS/8 was the

training of one of the implementors , Nate Teicholz, who went on to ~~The Nate went directly~~

~~on~~ to propose and first implemnt the RSTS timesharing sytem based ~~on a~~ on the

~~single user~~ language, ~~a~~ BASIC. ~~(or rather an extended version). At the first~~

~~implemntation, I argued that it should~~ the 11 should ~~be general purpose, rather than~~

~~single l a single language interpreter, but~~ the experience and complexity

of based on TSS/8 made them tru turn to an apparently ssimpler strucutrue

(TRSTS ultimately evolved to support gnerality and multiple langages.

~~Also, since the PPD-11 did not have adequate memory protection, I wanted to stet the~~

~~this specifiec before anothe processor was implementeed.~~

TSS/8 was ~~in re occurred~~ designed while I was at Carnegie Mellon, University, by a graduate student Ad van der Goor, as ~~and was~~ a reaction

to the incredibly high cost ~~and cincredibly~~ poor performance and poor reliability that

IBM was ~~providing on the 3 TSS/39 t~~ providing in the TSS/360 product (~~a~~ Modle 67 design).

Although ~~IBM~~ TSS/360 eventually worked, it wasn't marketed, and at most

~~formed a few~~ contributed some ideas ~~a~~ and trained ~~several probaly~~ thousands of

programmers for IBM. At CMU, a TSS/8 operated until 1974 when the

special ~~drum we~~ swapping disk we had used ~~finally~~ gave out. ~~and~~ The cost per user

or per job tended to be about 1/20 that of the TSS/360 system they run.

(Reference?)

The ~~Papers~~ Chapters

The following section describe the motivations for in~~x~~cluding the ~~various~~ chapters, ~~and~~ how ~~these~~ the machines ~~realte~~ relate to the family, and their contribu~~i~~tons to minicomputers. The final section describes ~~teh~~ the PDP-14 ~~and~~ ~~w~~ how it ~~a~~ came to ~~inot~~ being, ~~what ist is~~ and my own version of what it could (should) have been.

~~The DEC 338 Display Processor~~

The PDP-8

~~The~~ This ~~ca~~ chapter is a ~~substat~~ ~~re a~~ substantial ~~exp~~ extension to the original ~~of the chapter o f on the PDP-8~~ ~~that~~ first appearing in Bell and Newell. ~~The extensions are the obvious to include the machines after Bee~~ The extensions include the Omnibus and CMOS-8 ~~s~~ families which came out ~~f~~ after the ~~first article ws~~ chapter awas written. ~~In additions to the~~ Now ~~The chapter also now includes~~ enough time has passed to ~~get~~ gain a historical perspective, In this regard ~~history to~~ we look at how ~~the~~ technology~~s~~ has shaped the evolution and resulting structures from a price, paerformance, and packaging characteristics over the last 15 years.

The ~~bulk of the~~ main part of the ~~ca~~ original chapter remains in tact ~~from Bell and Newell~~ and is present ~~simply~~ to show how a mcahine can be understood by decomposing it into a number of levels whcih correspond to its physcial structure. ~~The chapter is also included for its historical value as being~~ ~~The news~~

s. aside from being

The LINC

~~Although~~ Since the LINC was designed outside DEC, it doesn't precisely fit the constraitns ~~of~~ the other chapters meet as being about DEC's computers, ~~and~~

~~and how they've evolved (or others have evolved them) by us and by our us useers and other researchers). It is included more weakly~~ However, since ~~this~~ the LINC was initially built from DEC Systems Modules, we ~~it could have been included.~~ feel somewhat justified A stronger tie is perhpas that DEC went on to manufacture ~~a version of~~ the LINC ~~based on~~ directly from the LINC design. Subsequently ~~we~~ Dick Clayton (in conjunction with Wes Clark) designed both the LINC-8, a ~~[illegible] processor~~ two-processor machine (LINC + PDP-8) which ~~would exe~~s executes both ISP's, ~~finally, we~~ and the PDP-12 a single physical processor that ~~would~~ executes either PDP-8 or LINC instructions. in parallel sequentially (by switching modes)

The ~~Hsitors~~ with some of the ~~critical events andd~~ characteristics of these LINC Family machines is given in the following table.

Table here attached

~~A much more appro~~

Note the size has remained essentially constant at 1 cabinet over the computer's life.

There are ~~two more important~~ other reasons to include the LINC as members of the DEC 12-bit family. ~~One~~ The LINC ~~hasad~~ had a strongly influence ~~on the~~ d others ~~membresdfs~~ PDPs (including PDP-4 and 5). The LINC tape ~~fit~~ system ideas permiting a user to have personal files has been carried ~~, specificaly the PDP-5 and early mi other early computers and two, the~~ unit and over directly in the DEC tape design and programs. This device ~~has~~ made the personal computer for the researcher possible at a reasonable economic level. ~~LINC tape~~

~~A seco Other reasons for its~~ We ~~ahve~~ also included the LINC becuase the paper is especially well written, even though it does stress applications and ~~do play down the~~ minimimally ~~actualy~~ disusses the cosntruction, and structure ~~ae~~ and other engineering aspects ~~f~~ of the design.

The ~~syt~~ systems aspect of LINC (circa 1962) were really quite significantly ahead of ~~the were~~ their time at DEC, ~~for~~ We had been ~~really~~ stressing design ~~f~~ flexibilility and modularity ~~a~~ and provided ~~by having arbitrary busses and virtually unlimi an infinity of~~ a ~~number~~ large of wagy for interconnecting computer components together in order to ~~to be able to~~ have any type of strucutre. LINC is quite constrained (only 1 Kword or 2 Kwords were available) ~~the~~ with 2 LINC tapes and 2 CRTs. Most all other characteristics including the I/O interface were quite constrained. By bounding the sytem in the this way, it is possilbe to provide a ~~more~~ complete environment including software. ~~In fact, the operating~~

and of other system including the timesharing work in terms of providing a clean, simple, yet powerful console (two CRTs) and file system interface. (note Fig. timeline).

~~system equivalent did it appears that the equivalent operating system~~

~~did not appear until OS 8 was written in 19___.~~

Note the DECtape library system came 2 years later on PDP-5.

DEC 338 Display Computer

~~The 338 is included primarily because of its historical pers~~

~~because~~

The 338 is included mainly for historical reasons, and because it is a variant of the PDP-8 family. We believe it ~~was~~ is one of the very earliest ~~computers to~~ based on a ~~PDP-8~~ be a display processor based computers, ~~It is also the on There ahv have been~~ — if not the first. The question of being first is clouded by the state of the evolution as we discuss below ~~extensions~~

It clearly shows how the application need drives a complete change in ~~data struct~~ hardware so as to interpret the necessary data type (a graphic picture).

The 338 was extended and applied to PDP-9 (in the form of the 339) and ~~the display processor concept was~~ then greatly extended ~~as~~ in the ~~and are attached~~ ~~to the~~ PDP-15, VT15, ~~as a~~ display processor.

~~In essence, the~~ ~~Although we can look at a at the 338 from a general~~

~~viewpoint, it is has~~

Although the ~~PDP-~~ 338 has the right general capabilities, it doesn't have the refinements ~~(~~ that later displays processors for PDP-15 and ~~the~~ PDP-11 have (GT40 and GT60). ~~Similarly~~ The DEC processors displays ~~have~~ tended to ~~evolve in a part along~~ extend the ~~lines of the GT~~ 338, whereas there have been departures from this structure (CMU this?). ~~other displays have departed from this structure and appear to better~~ oriented to the picture definition and display by being defining pictures in terms of tables. ~~with poin which have characterst that are~~ to be displayed When the tables do exceed the limits, ~~pc picture~~ requires the definition of a subpicutre, it is defined by characters ~~which are made up of a new set~~ table, character definition table.

Also there is a question of having a balanced computer. Since the ~~338 is~~ display processor is so relatively expensive, ~~then any~~ then the host processor should not be a bottleneck, nor ~~need~~ must it be significantly cheaper than the display part,

~~In generla most of the display processor s a re operate~~

~~Ant Another ob~~

An observation ~~about the evolution~~ that ~~of~~ display ~~term~~ processors ~~was made by~~ (Evans and Myers , ~~1968~~ 19??)
and perhaps other ~~that~~ specialized processors ~~teneded to start out~~ evolve in a fashion
akin to the wheel of reincarnation. Figure Rein shows how one starts with a very simple
basic design to have ~~basic graphs ca capabilities o attached to~~ graphics output for picture a computer.

1. The computer includes ~~a simple~~ a single instruction ~~a~~ display controller
which can plot a point~~s~~ under command of the ~~a~~ central~~l~~ processor. For most alll (on a point by point basis)
displays , except storage scopes , the processor can barely calculate the next ~~s~~ point
fast enough to keep the scope refreshed. Hence, the system is processor bound,
and the display may be idle. The original PDP-1, ~~sco~~ Type 30 scope is typical of this position.

2. By adding ~~vector plots the~~ the ability to plot lines (vectors) a single
instruction to the display processor will free ~~much~~ some of the procesor
and begin~~s~~ to ~~do~~ keep all but he fastese t scope busy. ~~The pro~~

3. Having reralized that ~~most much of what is is displayed is~~ characters, are mainly displayed
then ~~the~~ commands to display a ~~particular~~ character is added, further
freeing the processor ~~and~~. (Many of the point plotting displays were extended to have character generation plotting, The alphanumeric scopes ~~are, in~~ have ~~essence~~ this capability. (less vector plotting).

4. In reality, a picture doesn't consist~~e~~ of ~~a~~ just characters and
vectors, but each element of the picture is ~~a~~ actually a string of characters
to be ~~ds~~ displayed ~~s~~ at a particular point , and a set of ~~co~~ closeds
or open ~~poysols~~ polygons to be displayed starting at a partiuclar point. By
~~pro~~ providing a Direct Memory Access channel, ~~to the display, the~~ it
can fetch each string of text and generate ~~closes~~ polygons ~~(subpictures~~ with no
the display control with a ~~pro~~ central processor interaction.

5. Now with the ability to put up sub-pictures, with no processor
intervention, it is ~~necessary~~ easy for the whole picture to be displayed. ~~This~~
~~is accomplished~~ by linking the elemnsts together in some fashion. This
merely requires jump and subroutine call instructions (so common
picture elemnts don't have to be redefined). This is roughly ~~point~~ the capability of the 338 and most other display processors.

6. Now, we realize that all the paths and all the state is present ~~ofr~~ for
a ~~processor~~ fully general purpose processor, such that the central processor
need never be called ~~upon the main processor~~ again. There is full genrality. This requires a slightly more general purpose interpreter,
By minor perturbations, the proceesor design can be refined in
such a way as to ~~e~~ execute the same ISP ~~of~~ as the original ~~machine~~ host,

~~Bea~~ ~~pa~~ because the incompatibility cost is too great. Two processors

require two com~~i~~plers, diagnostics, manuals, and support for use. Finally, we have

arrived back to the same capability as we started. The original processor is completely fr~~r~~ee

and there is a display procesor ~~of some sort capabile~~ with the capability of executing both the

original ~~Z~~ISP plus the ~~ISP of the~~ display ~~task~~. ISP. ~~A1~~

8. Alternatively, we can take this proceesor and isolate it as a seperate

computer which we reconnect in some fashion back to the originals

~~P~~ central processor-primary memory pair.

~~TThe~~ ;

The Cache-Based PDP-8

~~In the paper~~ Chapter 7 ~~sided~~ describes a design for a ~~computer that~~ PDP-8 ~~was~~ constructed by ~~D~~ Professor David Casasent at CMU. Inititally the project was proposed ~~in order to~~ for ~~a~~ do exploratory advanced developemnt on the desirability and feasibility of using Emitter Coupled Logic (ECL) for designing very fast, marketable minicomputers. ~~Also, since we~~ As the investigation proceeded, ~~it was clear that~~ the easiest way to build a fast minicomputer ~~was to a cost effective, fast minicomputer was to use~~ turned out to be using cost-effective a cache structure in order to reduce the cost of primary memory ~~over a fully fast, memory without cache.~~ Also, instead of ~~doing for an ultra~~ designing a very fast PDP-8, which ECL woul have provided, the ~~spr sadl~~ was modified to be more modest and potentially marketable. TTL Schottky was used in the design, even though the logic family was just beginning to evolve.

a large, very fast memory need emerged. Such a memory, ~~would be~~ turned out to be so costly that such a computer would not be ~~feasibility~~ feasible from a market standpont

~~When it came time to to apply the so beyoned the prot~~

Inoreder to make the prototype more nearly marketable, withouth a complete ~~rot~~ rethinking and redesign, the design was also ~~const~~ constrained to be compatiable with the most of the 8/E Omnibus backplane and parts.

~~A~~ The prototype did not become operational as quickly and cleanly as possible, hence the use of the prototype to stimulate a market ~~did fell short~~ did not occur. All through the design ~~,~~ there was little interest ~~or no inter~~ ~~as to the~~ the marketability of the design. It was carried forwared in order to test the cache applicability ~~and~~, circuitry, ~~and~~ and ease and techniques for ~~of~~ building a fast 8. As is true in most ~~prot~~ new products, the fact that a market is not there is clear. We were not losing sales to very fast minicomputers, hence there ~~is~~ was (and is) no market for ~~is~~ very fast computers. Furthermore

our customers are of little or no help because unless faced with the widescale availability of such a machine, the applications generally don't emerge.

~~The~~ A number of things were learned ~~about the des~~ from doing the work. It did influence the 11/45 ~~by~~ ~~forcing it to install the bipolr~~ although the designers rejected the cache notion... ~~port.~~ ~~While non~~ ~~I~~ although it ~~it was initially~~ since they believed that the cache concept clearly didn't apply to PDP-11 program behavior. Eventually Strecker persevered and convinced ~~the~~ 11 designers to build a cache-based ~~min~~ larger scale minicomputer... and this ~~ba~~ became the 11/70. as described in Chapter X. fortunately, ~~a~~ the ~~a~~ processor was designed to operate with all bipolar, fast memory.

~~some~~ other ~~11~~ 11

The paper illustrates the use of the 8 as an experimental vehicle for understanding ~~this design~~ a design in terms of ~~together with~~ ~~the~~ process ~~of modeling~~ the program behavior ~~to affect~~ ~~implement~~ It also allows one to measure the time flow of ideas from research, advanced development, and finally into production machines.

~~(by mod~~

It also shows how, by rather minor perturbations, ~~a~~ ~~most of~~ a ~~the~~ design can be ~~accompt~~ kept compatible with its predecessors while providing rather ~~dra~~ dramatic performance and performance / cost improvements.

The problem first originated at General Motors as a controller for a large, transfer machine which made engine blocks by a sequence of operations.

PDP-14

~~The PDP-14 was designed for to meet several goals that the PDP-8 series did not meet.~~ The PDP 14 was designed expressly ~~for the~~ ~~to be~~ as a controller ~~which would replace complex relay sequential circuits used as controllers~~ for complex electromechanical machinery (such as transfer machines, conveyors, non-numerical milling) ~~and other machines.~~ ~~In order to make this replacement, the d such a machine had to be somewhat cheaper, than PDP-8 (*/l and have rea relatively fixed, read only type memories.~~

~~As such,~~ It often replaced or supplemented large relay sequential circuits.

Such a controller

~~In order to be able to replace the relay based controllers and solve other more compolex digital control tasks the units shoudld:~~

1. Be ~~cheaper t to build~~ lower priced, and easy to operate (with lower ~~price~~ life cycle costs).
2. ~~Easier to use and people trained in la w designing equipment using a relay ladder diagram type of deign could use it~~ solve the control problem ~~by being~~
3. operate in the ~~the~~ high electrical noise environment
4. operate in the ~~poor fer temperature, physical, industrial~~ poor physical industrial environemnt ~~that w iscah~~ characteristic of ~~this typ~~ the ~~type of~~ machines it controls
5. Have the appropriate I/O interfaces to sense contracts and output to relays

control power

~~Engineers from General Motors contacted DEC inorder to work on this probalem.~~ Although a ~~PDD~~ PDP-8 ~~coul could~~ might have been programmed to carry out the task, it was rejected ~~(now in hindsight probably for the following reasons:~~ either not considered or else because:

1. the Cost was too great;
2. There was some perception that the ~~programming was nothing like the problem being solved.~~ a conventional stored program computer could not solve the problem;

3. The circuits ~~were not o~~ did not have ~~high enough~~ adequate noise margins to operated in the environment

4. ~~The 8 could not bmeet the~~ noise constraints. physical environment constraints could not be met

5. There was inadequate I/O

As a result, the PDP-14 was proposed and designed expressly to ~~solve the~~ solve the problem and cost less than the ~~be cheaper than the~~ PDP-8. ~~It did this by basically removing all the~~ The 14 removes has no data operations except on ~~the~~ a single bit (boolean). In fact the 14 has only a Program Counter, and a single 1 bit Accumulator. The most complex instruction is Jump to Subroutine. (or rather the 8/I — which was just going into production).

2. In order to solve the the ~~particular~~ Boolean equations that a conventional relay controller solves in parallel, the 14 ~~goes through and~~ must solves equations sequentially at a rate of ~~1 once each 30 ms~~ approximately 30 hz ~~milliseconds~~, giving the illusion (which is quite adequate) that the equations are ~~being~~ solved in ~~a~~ parallel.

The 14 has ~~a particular set of~~ instructions for taking data ~~in~~ from input sensors, solving the equation and outputting the result to a relay. ~~The instruc~~ The ISP of the PDP-14 is given in Fig X.

3. The ~~discrete circuits were~~ circuit logic was slowed ~~down~~ in order to have improved noise margins. It was felt that core ~~mem~~ memory ~~meoires~~ did not have adequate susceptiblility ~~to~~ noise, hence couldn't be used. The braided , ~~rpo~~, read only memory was used ~~in order to solve this problem~~

4. ~~A~~ The unit was housed in a dust proof enclosure

5. Appropriate I/O interfaces were designed.

A number of 14's were built and installed in the ~~various a~~ intended applications. Programming was carried out over the period 1970-1972 ~~using several~~ compilers ~~for~~ in languages supported by that operated on PDP-8. These languages allowed users to write:

1. ordinary assembly programs (resembling PDP-8 programs)

2. express ladder diagram (in effect, these are ~~by~~ a ~~this is a~~ set of Boolean equations

3. program as a flow chart; ~~to control~~ i.e. as a sequential machine

~~The machine demands and evolved~~ As the 14 ~~was~~ were used, ~~causing~~ and its competitive, ~~contem~~ contemporary machines

~~the creating~~ a demand for a second generation controller. was created.

~~By now,~~ By 1972, the additional requirements included:

1. Better cost, ~~sicne others were maiking~~ due to competitive similar products, including the companies whose relay products the 14 dispalced

2. There were many more functional demands including having an easy to change read only memory with the ability to load programs via a communications line or console ~~that could be~~ connected ~~to the computer. controller.~~ In addition the controllers should connect ~~to a in a~~ in a network fashion and report back ~~the observations~~ status and results to ~~a another~~ a supervisory computer ~~of some sort.~~ at the next level of ~~the~~ a hierarchy.

The controller shoud be capable of recording ~~certain~~ events ~~and signalling the m~~ (eg. counting the number of parts it processed). It needs ~~The controller should have~~ timers which would ~~could~~ can be set ~~such t that they~~ as ~~were~~ part of the control equations ~~to~~

~~3 and 4 the~~

~~3-65~~ 3-5. The unit should operate over an even wider environmental range, ~~and~~ with a more complete set of I/O.

Thus, the PDP-14/30 ~~evl~~ evolved as can be seen ~~t in the~~ ~~Figure~~ Y.

Another critical part of the 14 ~~30~~ includes a programming, program load and observation console, the VT14, which consists of a CRT and PDP-8 mounted in a portable housing ~~with the capability to load programs in to the 14/30. The VT14 coudl also~~

~~The~~ Since the 14/30 ~~could~~ can also ~~di~~ report back status of its input and output variables, the VT 14 also ~~had~~ has the ability to display the status of the ladder diagram (ie "whether a relays ~~was~~ is on or off and ~~what~~ the contact ~~psos~~itons ~~were~~). A diagram is shown in Fig. —

At the time when the 14/30 was proposed, Iurged that it not be built, because

1. A standard PDP-8 ~~coimputer (a PDP-8 was~~ is chaeapre) to build and we wouod get more production volume and lower costs by not building a specaily unit.

2. The PDP-8 ~~cabinetry would get modified to have the~~ could be ~~SI ISP would be~~ extended trivially to have the complete 14 ISP.

~~In addition,~~ Thus the PDP-8 ISP would be available to do ~~several~~ the tasks that

~~were evolving and~~ would inevitably evolve:

a. ~~[illegible]~~ self-diagnosis

b. more extensive ~~[illegible]~~ counting and timing functions

c. dealing with ~~the occasional bit of~~ non-boolean data

(eg. time, non-discrete events ~~such as~~ including analog position)

d. general ~~[illegible]~~ control in both the analog and digital ~~[illegible]~~ domain

e. ~~[illegible]~~ communicate with computer networks requiring much better protocol ~~error and protocol~~ control ~~lines~~ for intelligent communication and error-free

3. The PDP-8 circuits had adequate electricical noise margins,

4. the 8 ~~would~~ should be packaged in the proper cabinet for the physical environment

5. ~~and the~~ There would be adequate interface parts

~~6. The sys system could also handle all the non-con other types of control that a simple, boolean a equatin solver could not handle.~~

Although I was ~~[illegible]~~ unsuccesful in convertins the 14/30 to become an 8 with appropriate extensions and packaging some of the 8 parts were used in the 14/30 design. At this time, ~~it is would appear that~~ some of the initial conjecture about the 14 was ~~partically~~ correct. The product was discontinued primarily because it required ~~[illegible]~~ ~~more~~ intensive support that was outside the mainstream resourdces, hence ~~it represen~~ dropping represented a strenghtening of purpose.

While I ~~at one level the 14~~ it is easy to dismiss the 14 as a trivial controller special purpose ~~computer~~ that should never have existed, this is to say that the problem being solved should not ~~have~~ exist. ~~and that is~~ Such a conjecture ~~not only presumptious, but~~ flies in the face of ~~against~~ the adage "that the custoer is always right." ~~What is reasonable to say What I do I~~

The problem is worthwhile and ~~[illegible]~~ and should have been solved with the full generality of ~~the~~ a computer. The ultimate (several next generations) ~~controlle state~~ state of the controller is clear: a computer is needed to handle the full genrality of the problem as it was, as it slowly evolved (when the 14/30 was designed) and as it will clearly evolve to. The solution was to adapt and evolve ~~the 8~~ by a general purpose computer

~~a comp~~ a combination of hardware and software to be ~~an acceptable~~ and still is

the best ~~what can easily~~ controller ~~that~~ Such evolution can easily surpass

any fixed ~~type~~ control strucutre that would have evolved ...not evolving the

8 was the error.

to the application