This material appears in *Computer Engineering: A DEC View of Hardware Systems Design* by C. Gordon Bell, J. Craig Mudge, and John E. McNamara.

8

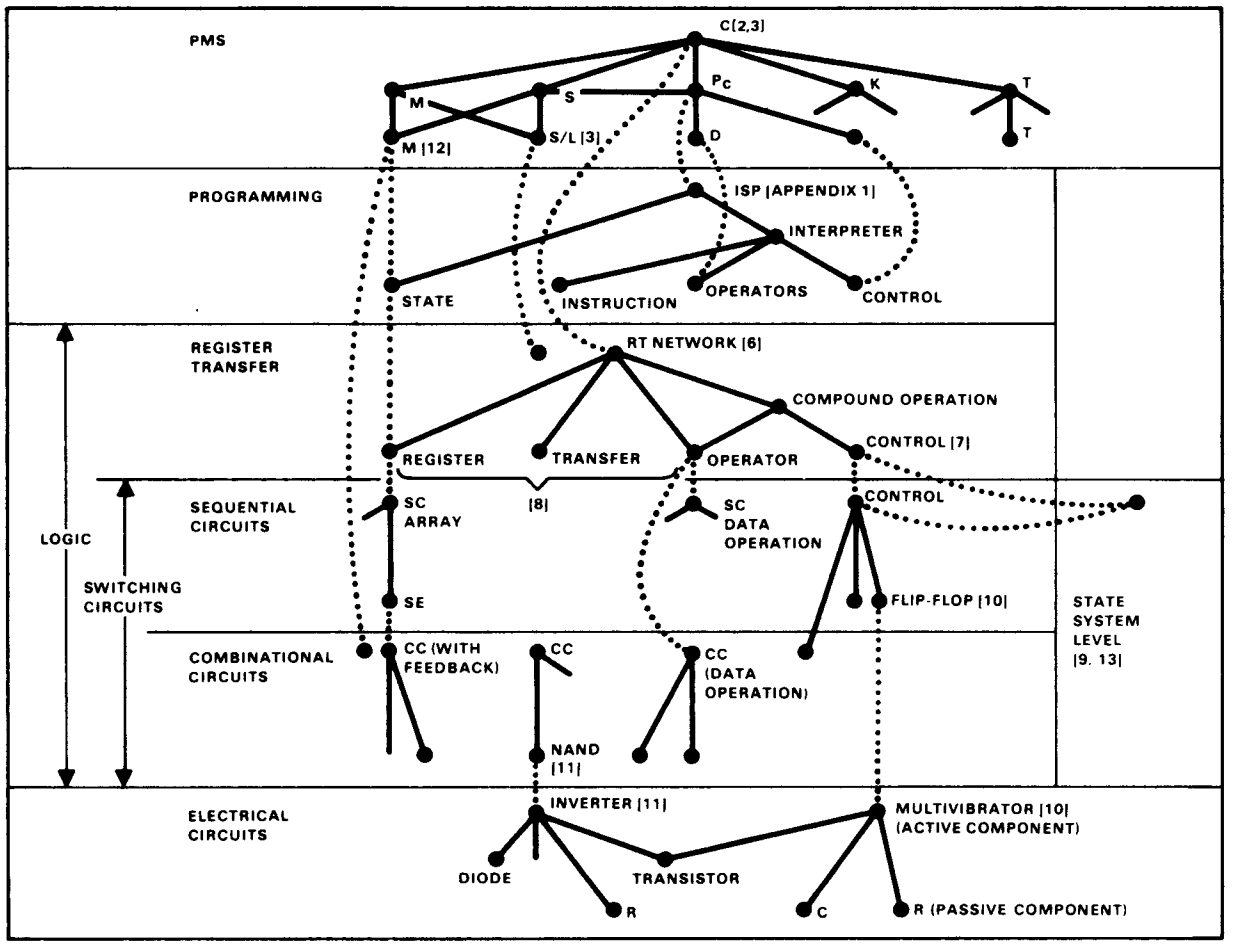
Structural Levels of the PDP-8

C. GORDON BELL, ALLEN NEWELL,
and DANIEL P. SIEWIOREK

The history of the DEC 18-bit and 12-bit computers, summarized briefly in the previous two chapters, was basically that of a recursive process in which new technology was applied and re-applied to the same basic designs to obtain improved price/performance ratios. In the late 1960s, the availability of relatively inexpensive integrated circuits made logic cost a less pressing concern. Computer engineering, and architectural issues of elegance, flexibility, and expandability, grew more important as the importance of architecture to total system price/performance became more evident. The PDP-11 papers in Part III elaborate on these issues, but first the hierarchical nature of computer systems design will be explored by examining the PDP-8 from the top down to lay the basic groundwork for future architectural discussions. The description of the PDP-8 will use some of the processor-memory-switch (PMS) and instruction set processor (ISP) notations introduced in *Computer Structures* [Bell and Newell, 1971]. These compact and straightforward notations are useful in comparing and analyzing computer architectures, and their use in the PDP-8 context should be helpful to the reader when encountering these notations in other papers.

A map of the PDP-8 design hierarchy, based on the Structural Levels View of Chapter 1, is given in Figure 1, starting from the PMS structure, to the ISP, and down through logic design to circuit electronics. These description levels are subdivided to provide more organizational details such as registers, data operators, and functional units at the register transfer level.

The relationship of the various description levels constitutes a tree structure, where the organizationally complex computer is the top node and each descending description level represents increasing detail (or smaller component size) until the final circuit element level is reached. For simplicity, only a few of the many possible paths through the structural description tree are illustrated. For example, the path showing mechanical parts is missing. The descriptive path shown proceeds from the PDP-8 computer to the processor and from there to the arithmetic unit or, more specifically, to the Accumulator (AC) register of the arithmetic unit. Next, the logic implementing the register transfer operations and functions for the *j*th bit of



[X] indicates figure number of instance.

Figure 1. PDP-8 hierarchy of descriptions.

the Accumulator is given, followed by the flip-flops and gates needed for this particular implementation. Finally, on the last segment of the path, there are the electronic circuits and components from which flip-flops and gates are constructed.

ABSTRACT REPRESENTATIONS

Figure 1 also lists some of the methods used to represent the physical computer abstractly at the different description levels. As mentioned previously, only a small part of the PDP-8 description tree is represented here. The many documents which constitute the complete representation of even this small computer include logic diagrams, wiring lists, circuit schematics, printed circuit board photo etching masks, production description diagrams, production parts lists, testing specifications, programs for testing and diagnosing faults, and manuals for modification, production, maintenance, and use. As the discussion continues down the abstract description tree, the reader will observe that the tree conveniently represents the constituent objects of each level and their interconnection at the next highest level.

THE PMS LEVEL

The PDP-8 computer in PMS notation is:

```
C('PDP-8; technology:transistors; 12 b/w;
   descendants:'PDP-8/S, 'PDP-8/I, 'PDP-8/L,
   '8/E, '8/F, '8/M, '8/A, 'CMOS-8;
   antecedents: 'PDP-5;
   Mp(core; #0:7; 4096 words; tc:1.5 μs/word);
```

```
Pc(Mps(2 to 4 words);
   instruction length:1|2 words;
   address/instruction:1;
   operations on data:(=, +, Not, And, Minus
   (negate), Srr 1(/2), Slr 1 (×2), +1)
   optional operations:(×,/,normalize);
   data-types:word,integer,Boolean vector;
   operations for data access:4);
P(display; '338);
P(c; 'LINC);
S('I/O Bus; 1 Pc; 64 K);
Ms(disk, 'DECtape, magnetic tape);
T(paper tape, card, analog, cathode-ray tube)
```

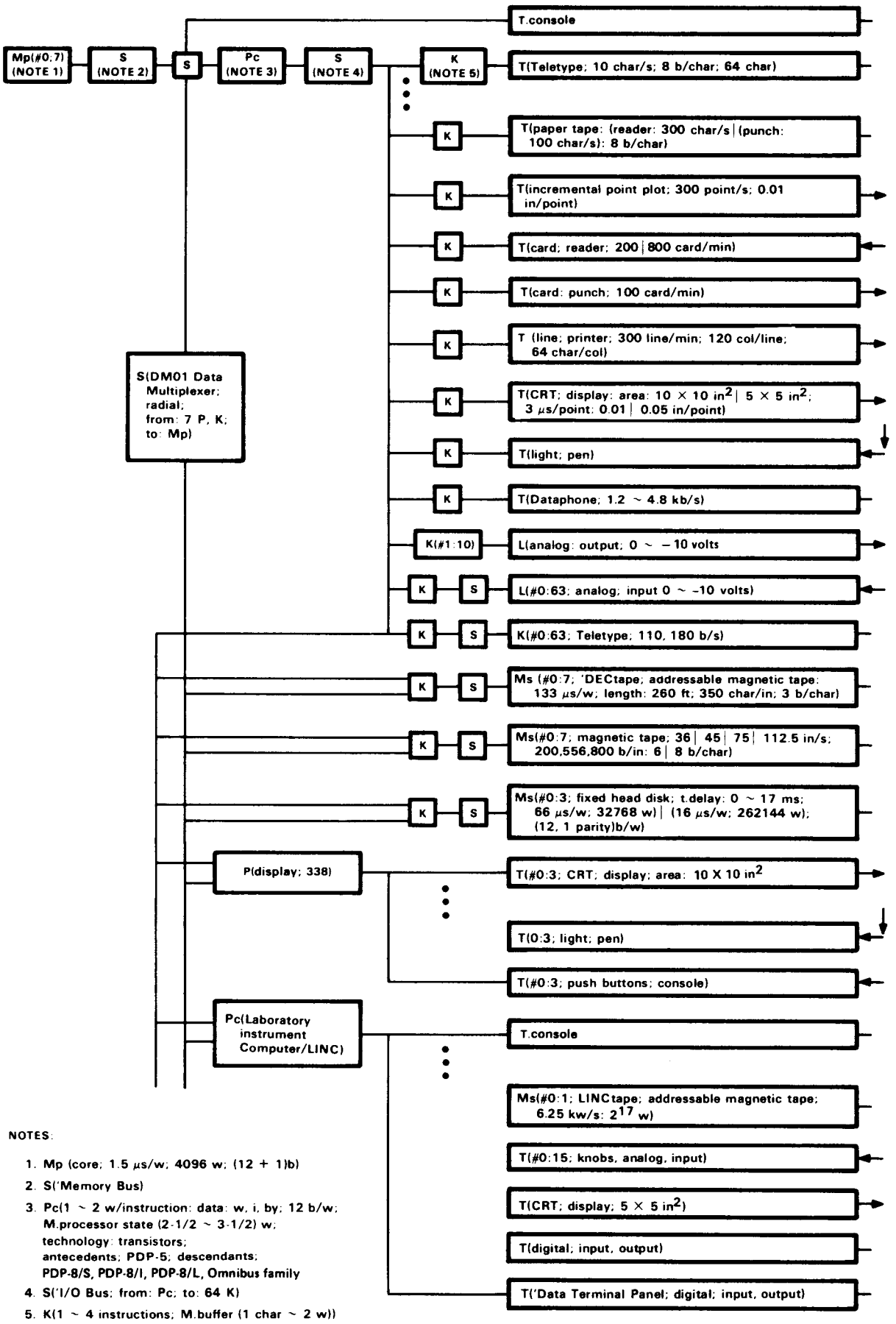
As an example of PMS structure, the LINC-8-338 is shown in Figure 2; it consists of three processors (designated P): Pc('LINC), Pc('PDP-8), and P.display('338). The LINC processor described in Chapter 7 is a very capable processor with more instructions than the PDP-8 and is available in the structure to interpret programs written for the LINC. Because of the rather limited instruction set being interpreted, one would hardly expect to find all the components present in Figure 2 in an actual configuration.

The switches (S) between the memory and the processor allow eight primary memories (Mp) to be connected. This switch, in PMS called S('memory Bus; 8 Mp; 1 Pc; time-multiplexed; 1.5 μs/word), is actually a bus with a transfer rate of 1.5 microseconds per word. The switch makes the eight memory modules logically equivalent to a single 32,768-word memory module. There are two other connections (a switch and a link) to the processor excluding the console. They are the S('I/O Bus) and L('Data Break; Direct Memory Access) for interconnection with peripheral devices. Associated with each device is a switch, and the I/O Bus links all the devices. A simplified PMS diagram (Figure 3) shows the structure and the logical-physical transformation for the I/O Bus, Memory Bus, and Direct Memory Access link. Thus, the I/O Bus is:

S('I/O Bus duplex; time-multiplexed; 1 Pc; 64 K; *Pc controlled, K requests*; t:4.5 μs/w)

The I/O Bus is nearly the same for the PDP-5, 8, 8/S, 8/I, and 8/L. Hence, any controller can be used on any of the above computers provided there is an appropriate logic level converter (PDP-5, 8, and 8/S use negative polarity logic; the 8/I and 8/L, positive logic). The I/O Bus is the link to the controllers for processor-controlled data transfers. Each word transferred is designated by a processor in-out transfer (IOT) instruction. Due to the high cost of hardware in 1965, the PDP-8 I/O Bus protocol was designed to minimize the amount of hardware to interface a peripheral device. As a result, only a minimal number of control signals were defined with the largest portion of I/O control performed by software.

A detailed structure of the processor and memory (Figure 4) shows the I/O Bus and Data Break connections to the registers and control in the notation used in the initial PDP-8 reference manual. This diagram is essentially a functional block diagram. The corresponding logic for a controller is given in Figure 3 in terms of logic design elements (ANDs and ORs). The operation of the I/O Bus starts when the processor sends a control signal and sets the six I/O selection lines (IO.SELECT<0:5>) to specify a particular controller. Each controller is hardwired to respond to its unique 6-bit code. The local control, K[k], select signal is then used to form three local commands when ANDed with the three IOT command lines from the processor. These command lines are called IO.PULSE.1, IO.PULSE.2, and IO.PULSE.4. Twelve data bits are transmitted either to or from the processor, indirectly under the controller's control. This is accomplished by using the AND/OR gates in the controller for data input to the processor, and the AND gate for data input to the controller. A single skip input is used so that the processor can test a status bit in the controller. A controller communicates back to the processor via the interrupt request line. Any controller wanting attention simply ORs its request signal into the interrupt request



NOTES:

1. Mp (core; 1.5 μs/w; 4096 w; (12 + 1)b)
2. S('Memory Bus)
3. Pc(1 ~ 2 w/instruction: data: w, i, by; 12 b/w; M.processor state (2-1/2 ~ 3-1/2) w; technology: transistors; antecedents; PDP-5; descendants; PDP-8/S, PDP-8/I, PDP-8/L, Omnibus family
4. S('I/O Bus; from: Pc; to: 64 K)
5. K(1 ~ 4 instructions; M.buffer (1 char ~ 2 w))

Figure 2. LINC-8-338 PMS diagram.

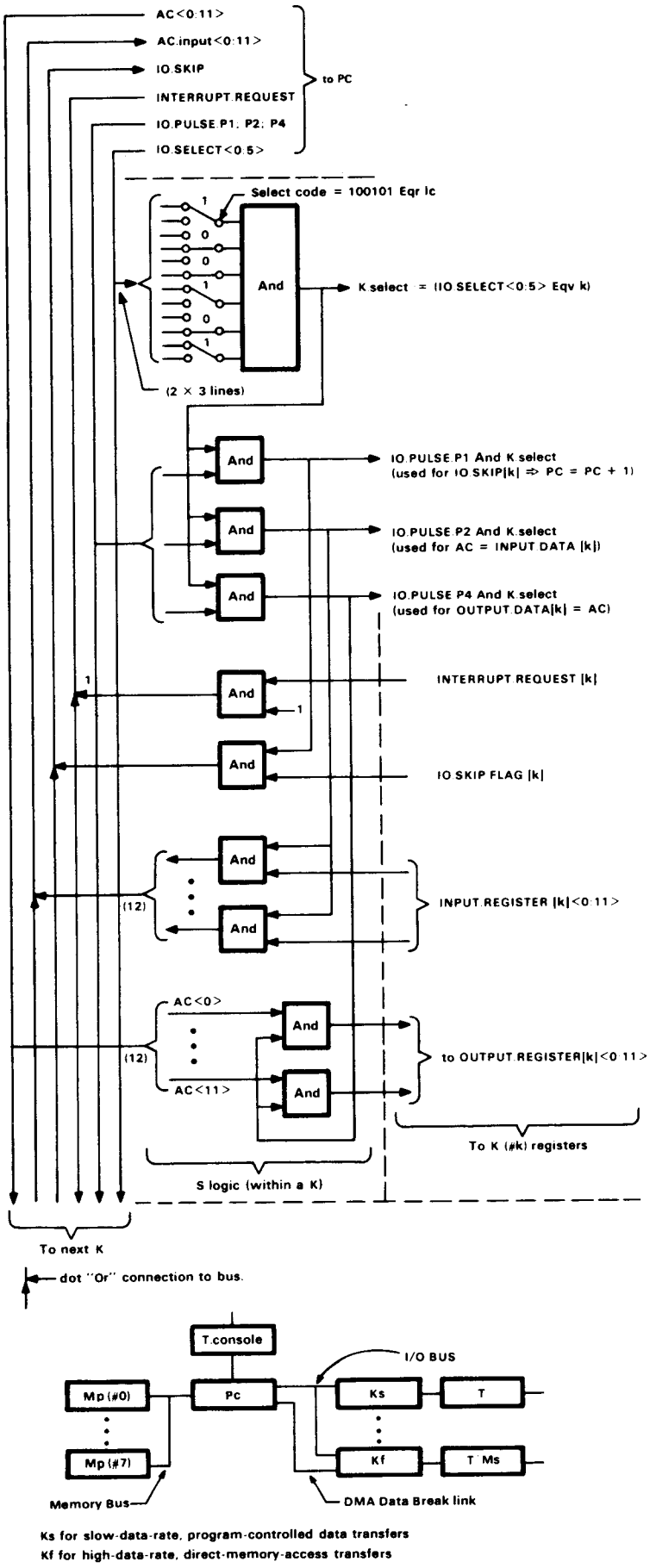


Figure 3. PDP-8 S('I/O Bus) logic and PMS diagrams.

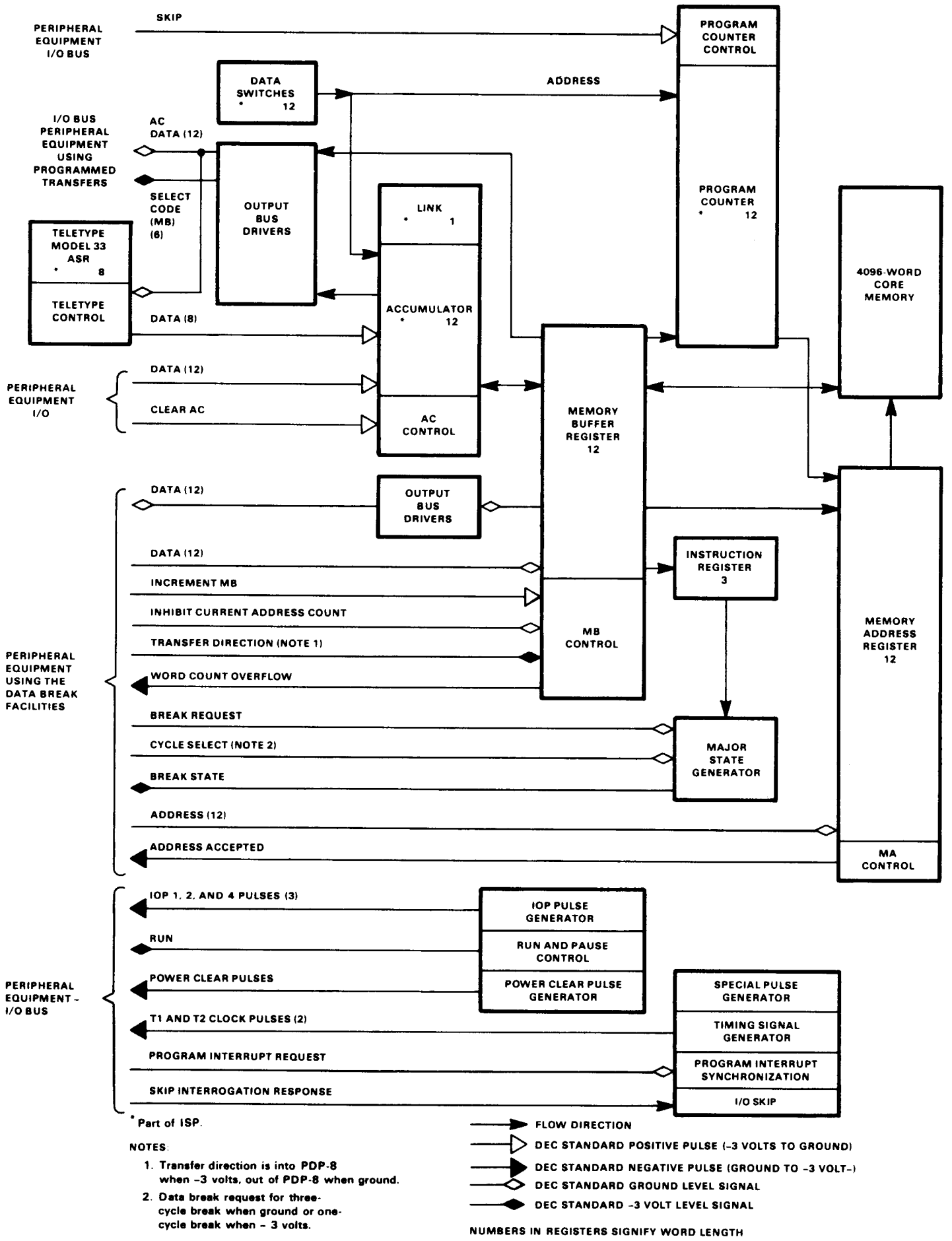


Figure 4. PDP-8 processor block diagram.

signal. Normally, the controller signal causing an interrupt is also connected to the skip input, and skip instructions are used in the software polling that determines the specific interrupting device.

The Data Break input for Direct Memory Access provides a direct access path for a processor or a controller to memory via the processor. The number of access ports to memory can be expanded to eight by using the DM01 Data Multiplexer, a switch. The DM01 port is requested from a processor (e.g., LINC or Model 338 Display Processor) or a controller (e.g., magnetic tape). A processor or controller supplies a memory address, a read or write access request, and then accepts or supplies data for the accessed word. In the configuration (Figure 1), Pc('LINC) and P('338) are connected to the multiplexer and make requests to memory for both their instructions and data in the same way as the PDP-8 processor. The global control of these processor programs is via the processor over the I/O Bus. The processor issues start and stop commands, initializes their state, and examines their final state when a program in the other processor halts or requires assistance.

When a controller is connected to the Data Break or to the DM01 Data Multiplexer, it only accesses memory for data. The most complex function these controllers carry out is the transfer of a complete block of data between the memory and a high speed transducer or a secondary memory (e.g., DECtape or disk). A special mode, the Three Cycle Data Break (described in Chapter 6), allows a controller to request the next word from a block in memory.

The DECtape was derived from M.I.T.'s Lincoln Laboratory LINCtape unit, as indicated in Chapter 7. Data was explicitly addressed by blocks (variable but by convention 128 words). Thus, information in a block could be replaced or rewritten at random. This operation was unlike the early standard IBM format magnetic tape in which data could be appended only to the end of a file.

PROGRAMMING LEVEL (ISP)

The ISP of the PDP-8 processor is probably the simplest for a general purpose stored program computer. It operates on 12-bit words, 12-bit integers, and 12-bit Boolean vectors. It has only a few data operators, namely, =, +, minus (negative of), Not, And, Slr 1(rotate bits left), Srr 1 (2 rotate bits right), (optional) ×, /, and normalize. However, there are microcoded instructions, which allow compound instructions to be formed in a single instruction.

The ISP of the basic PDP-8 is presented in Appendix 1 of this book. The 2^{12}-word memory (declared M[0:4095]<0;11>) is divided into 32 fixed-length pages of 128 words each (not shown in the ISPS description). Address calculation is based on references to the first page, Page.Zero, or to the current page of the Program Counter (PC\Program.Counter). The effective address calculation procedure, called eadd in Appendix 1, provides for both direct and indirect reference to either the current page or the first page. This scheme allows a 7-bit address to specify a local page address.

A 2^{15}-word memory is available on the PDP-8, but addressing more than 2^{12} words is comparatively inefficient. In the extended range, two 3-bit registers, the Program Field and Data Field registers, select which of the eight 2^{12}-word blocks are being actively addressed as program and data. These are not given in the ISPS description.

There is an array of eight 12-bit registers, called the Auto.Index registers, which resides in Page.Zero. This array (Auto.Index[0:7]<0 :11>:=M[#10: #17]<0:11>) possesses a useful property: whenever an indirect reference is made to it, a 1 is first added to its contents. (That is, there is a side effect to referencing.) Thus, address integers in the register can select the next member of a vector or string for accessing.

The processor state is minimal, consisting of a 12-bit accumulator (AC\Accumulator

7. A high current path flows via the X and Y selection switches, but in an opposite direction to the read case (see item 2). If a 1 is written, no inhibit current is present and the net current in the selected core is −Iswitching. If a 0 is written, the current is −Iswitching +(Iswitching/2) and the core remains reset.
8. The inhibit and write logic signals are turned off at time tmd specified by timing in the memory module, and the memory cycle is completed.

Device Level

For a discussion of the behavior of the transistor as it is used in these switching circuit primitives, the reader should consult semiconductor electronics and physics textbooks. It is hoped that the reader has gained a sense of how to think about the hierarchical decomposition of computers into particular levels of analysis (and synthesis) and that the hierarchical approach will be of aid in the reading of Part III.

Opposite:

Top, left to right:
- PDT-11 programmable data terminal.
- VAX-11/780.

Bottom, left to right:
- Model 20 central processor.
- PDP-11 packaging showing cabinet level integration.