PDP-11 MICROPROGRAMMING

J.F.O'Loughlin

I. PDP-11 ARCHITECTURE (BACKGROUND)

- A. UNIBUS Interface
  - 1. Separation of Machine Elements
  - 2. Strength of a Common Interface Specification
- B. Instruction Set
  - 1. Multiple Address Modes
  - 2. Multiple Register
  - 3. Multiple Word Instructions
- C. System Stack
  - 1. Subroutine and Instruction Usage
  - 2. Interrupt Usage

II. MICROPROGRAMMING

- A. General Comments
- B. Specific Application to PDP-11 Processors

III. DESIGN GOALS

- A. PDP-11/20
- B. PDP-11/40
- C. PDP-11/05
- D. PDP-11/45

IV. MICROPROGRAMMING DESIGN TOOLS

- A. Breadbroad
- B. Software
  - 1. Debugging
  - 2. Documentation
- C. Simulator for Floating Point

V. IMPLEMENTATION

- A. PDP-11/20 } Data Path & Micropath together
- B. PDP-11/40 } Number of general registers, states
- C. PDP-11/05 } Commentary on microword relationship to
- D. PDP-11/45 } data path!
- E. Summary Charts
  - 1. Processor State Complexity
  - 2. IC Count
  - 3. Microprogramming Characteristics

VI. MICROPROGRAM EXPANSION

- A. Integral - PDP-11/40
  - 1. Expansion Instruction Set
  - 2. Floating Instruction Set

B. Separate - PDP-11/45 FPP

VIII. SUMMARY & CONCLUSIONS

A. Design
   1. Procedural (Segmentation)
   2. Developmental (Data Path & Flow Diagram)
   3. Documentation

B. Product Hardware
   1. Microprogramming Relationship to Design Goals
      a. PDP-11/40
      b. PDP-11/05
      c. PDP-11/45
   2. Size - Module Number
   3. Production
      a. Uniqueness of PROM's & ROM's
         1. Trade off between the two.
         2. Vendor delivery problem.
      b. Engineering changes
      c. Training

MICROPROGRAMMING THE PDP-11's

J.F.O'Loughlin

INTRODUCTION:

The recent microprogramming of three PDP-11 machines provides a unique opportunity to review the effects of microprogramming on a fixed architecture machine. Three separate design groups were involved (PDP-11/05, PDP-11/40, and PDP-11/45) each with specific design goals. Central to each of these efforts was the absolute requirement for common (or subset) software, common peripherals, and a background of common corporate resources and hardware. Comparisons are made to the original non-microprogrammed PDP-11/20 machine.

A total hardware development and implementation viewpoint is attempted. Microprogramming may reduce the size of a machine, alter its data paths or increase its speed; it can also alter the techniques of project documentation and development. The microprogramming impact is placed in perspective; it is a major element in the design of the new machines but it is not the only element. The continued development of denser integrated circuits is important; so also is the altering application of the machine which in turn affects design goal. The minicomputer is not theoretical; it is pragmatic both in application and design. This paper is so presented. The choices made for the PDP-11's may be right for them, but may have other meanings in another or future design.

PDP-11 ARCHITECTURE:

The fixed architecture under consideration is that of the PDP-11; three salient features should be noted: the UNIBUS[R]; the Instruction Set with multiple address modes and registers; and the System Stack for subroutines and interrupts.

R Registered Trademark of Digital Equipment Corporation

The UNIBUS provides the interconnection between components of the computing machine in Figure 1. This interconnection with its attendant specifications provides both a physical and logical discipline among components. Physically the UNIBUS provides signals consisting of: data; address; data control; bus ownership control; and power on-off control. The data transfers are asynchronous with interlocking control signals; four types of transfers can be initiated by a bus master device (DATA IN, DATA IN PAUSE, DATA OUT and BYTE DATA OUT). Note that bus mastership is required for data transfer and can be provided for any element on the UNIBUS , not just the processor. The processor, however, does have a slightly greater than equal role on the UNIBUS , it contains the logic that arbitrates the bus ownership. Two classes of non-processor ownership are provided: Non-Processor Request (Direct Memory Access) which occurs throughout processor operations; and Bus Request (usually effecting Interrupts) which occurs only at instruction completion and depends upon peripheral request priority versus program priority.

The discipline of the UNIBUS specification requires some logic cost within each system component; this cost, however, is well returned in the generality of the component interfaces. Specialized I/O instructions are not needed as I/O device registers (control and storage) occupy normal UNIBUS address space. Direct Memory Access transfers occur directly between the participating devices (disk and memory, for example) without processor interaction. Subsets of UNIBUS control are allowed; some devices are only master (processor registers) or slave (memory); others devices only acquire bus control for INTERRUPT operations. This flexibility accommodates the need for minimum logic and cost in some peripherals and allows power and speed in others. The various new processor accommodates the UNIBUS specifications in different ways - each meets the specification but implementation and speed may differ.

The Instruction Set of the PDP-11 machines is relatively strong and diverse. Over 400 instructions exist with their major characteristics as follows: multiple address modes; multiple general purpose registers, double and single operand instructions, byte or word operation, processor status word, and a predilection for stack operations. Figure 2 indicates the double and single operand instructions in reference to the address modes and general registers. Instructions can involve Register to Register, Register to Memory, Memory to Register, or Memory to Memory operations. Of special value is the inclusion of the Program Counter (PC) and the processor Stack Pointer (SP) within the General Registers; this allows relative and system stack ordered instructions. Most instructions provide for word or byte operation with the processor monitoring byte operations and odd address utilization. Bus errors (odd address, stack limit or non-existent memory) result in specific trap responses. A portion of a Processor Status word provides for a number of Branch Instruction conditions by storing characteristics of the last instructions data; the word also contains program priority to determine bus ownership upon peripheral bus request.

A system stack accommodates both subroutine response and processor interrupt response; processor trap instructions also utilize this stack. The specific sequence of operation for the Jump to Subroutine, Return from Subroutine and Interrupt response are noted in Figure 3. The Subroutine instructions use of the linkage register (RX) provides for arguments or addresses of arguments; the use the system stack (SP) allows nesting of subroutines and interrupts. The Interrupt response indicates the transfer of both the Program Counter (PC) and the Processor Status (PS) word to the system stack. The new PC and PS comes from the peripheral transmitted vector address and allows a change in program operation and the priority of program operation.

MICROPROGRAMMING:

Microprogramming as a concept for machine implementation was noted in 1951 by Professor Wilkes. Initial implementations were upon the larger machines with a variety of control store devices. At Digital, the medium sized PDP-9 machine used a magnetic E core implementation for its control store. Widespread use of microprogramming in the minicomputer field came only after Large Scale Integrated (LSI) circuits provided Read Only Memory (ROM) and the related Programmable Read Only Memory (PROM). These devices allow a direct, practical application of the economical advantages of LSI without the design time and cost requirements of custom LSI. Only the ROM pattern needs to be custom, the basic integrated circuit layout is standard and uniform. These devices provided the starting point for microprogramming the PDP-11 machines.

Given the means for economical microprogramming, should PDP-11's be done that way? The PDP-11 machines are strong yet complex, with this complexity providing the threshold for consideration of a microprogramming design. The Instruction Register operation code is not specifically laid out for microprogramming, but generalities exist in the format for the address modes. The asynchronous nature of the UNIBUS data and ownership transfers appears neither for nor against microprogramming; however, the separation and generalization of UNIBUS elements with defined interfaces is very characteristic of the microprogramming design concept. The need for optional, incremental expansion lends itself to microprogramming.

DESIGN GOALS:

The design goals for the machine provide a necessary reference for interpretation of the existing designs and the microprogram application to those designs. Note that microprogramming is not noted as a goal, it is an unspecified technique.

The PDP-11/20 primary design goal was for a fault free implementation of a lasting architecture. The implementation utilized off-the-shelf components. No major use was made of LSI or MSI components.

The PDP-11/40 is the follow-on machine for the PDP-11/20. Its major goals are improved price (less than a PDP-11/20) and performance (Register to Register operations in less than a microsecond) with optional, incremental expansion.

The PDP-11/05 is a minimum logic implementation with the belief that such an implementation is low cost. Some original effort was directed at a single module board design of a PDP-11 subset or a non-PDP-11 machine. Software and peripheral costs, however, require a PDP-11 machine and the present two board machine resulted with low cost, and consequentially low speed, the criteria.

The PDP-11/45 is a state of the art machine specifically designed to provide raw computational power. Both MOS and Bipolar semiconductor memory are provided with a separate Floating Point Processor. Cost was not an originally specified item with multilayer logic boards and high speed Schottky logic always considered necessary.

MICROPROGRAMMING DESIGN TOOLS:

A major microprogramming design tool is the computer driven Read/Write Control Store of Figure 4. A bipolar scratch-pad type memory interfaces to the processor breadboard; the word width and depth is that of the control store for each machine. The processor breadboard provides the micro address and control for the read only interface to the breadboard. Input and change for the control store comes through the UNIBUS interface to the PDP-11/20 system. This system is supported by Teletype, Paper tape and software.

Two types of software are provided: on-line debugging programs; and documentation programs. The on-line debugging program allows examination and modification of the breadboard control store. Input commands and data are checked, paper tape output is provided for generating Programming Read Only Memories (which then can be used as masters) and for generation of machine and IC documentation. Machine documentation is generated on a PDP-10, System 40 machine. Programs allow: the conversion of the PDP-11 papertape to a standard ROM (256 words X 4 bits) format; the generation to complete microflow for machine documentation; and the ability to edit for engineering changes.

A design tool enhanced by microprogramming was a simulator of the PDP-11/45 separate Floating Point Processor. With the data path and control strongly characterized by the microprogram flow diagrams, each register could be assigned a program memory location and each microtransfer directly simulated.

IMPLEMENTATIONS:

PDP-11/20 - The block diagram of Figure 5 indicates the simple nature of this original PDP-11 implementation. The Adder has latch registers on each input with a separate output register for Bus Address storage. The Scratch Pad general registers receive their inputs from the Adder output and provides output to each Adder input. A figure eight data path exists between the UNIBUS , the Adder, and the Scratch Pad.

Control for the data path comes from combinational logic decoding shift registers containing machine states. Four types of state information are utilized: major machine state (Fetch, Execute, etc.); instruction states within the major machine states (ISR1, ISR2, etc.); bus control states within instruction states (BSR1, BSR2, etc); and the Instruction Register decode. Usually four to six gate levels exist between the shift registers and the control signal to the data path.

PDP-11/40 - The block diagram of Figure 6 directly concerns itself with speed, generality and some economy. Register to Register transfers that require the loading of both Arithmetic Logic Unit (ALU) inputs with Scratch Pad Memory information and UNIBUS data are accommodated by direct paths. The need for a UNIBUS address to quickly initiate bus operation is accommodated by direct path to the Bus Address Register from the ALU output and the Scratch Pad Memory. The location of the Scratch Pad Memory between an outgoing data bus (DMUX) and an ingoing data bus (BUS D) allows the further addition of other memory or processing elements; this important generality allows expansion. The data struction while more complicated than the PDP-11/20 still is economically tuned to data interaction with core memory on the UNIBUS.

Direct microcontrol is exerted upon the Data Path for the multiplexors and Scratch Pad Memory selection; a single logic gate level is required for clocking signals. Only a small portion of the microcontrol word is used indirectly through decoding, these fields relate to constant generation or alteration of Arithmetic Logic Unit response as a function of Instruction Register decode. The minimum loop time (140 nanoseconds) for microword control (address, microdata output, buffer loading) matches the time for the most cirtical data path transfer (Scratch Pad Memory to the B Register at the ALU input). Different microtiming cycles are also provided for arithmetic operation through the ALU (200 nanoseconds), and for complete data cycles from the Scratch Pad Memory, through the ALU and back to the Scratch Pad Memory (300 nanoseconds). The microword is buffered so that next word look-up can occur during present word operation. The next microword base address is provided completely by the present microword; branching information inputs to the NOR gates prior to the microword buffer. This branching input location requires that branch conditions be specified by the microword, one microword before the branch.

PDP-11/05 - The block diagram of Figure 7 reflects economies in the number of data paths and data registers. Operations performed directly in the PDP-11/40 and PDP-11/45 often require two or more cycles. The Scratch Pad Memory must pass data through the Arithmetic Logic Unit (ALU) and output multiplexor to load the B Register on the ALU inputs. Byte operations for data justification require several rotate operations instead of a single swap operation. The UNIBUS interface is similar to that of the PDP-11/20. A separate Bus Address Register is provided, but data outputs require the holding of data through the ALU and its input registers. The emphasis is on economy with time being the major sacrifice.

The microcontrol of the PDP-11/05 continues these economies. The microword is used directly from the Read Only Memory; no buffer register is provided for the data protion of the microword. The microword access time is accommodated with the data operation time in the total microcycle time (300 microseconds). Fewer data path elements result in fewer microcontrol bits.

PDP-11/45 - The block diagram of Figure 8 is complex with multiple interconnections and multiple registers for speed. Two copies of the Scratch Pad Memory allow direct and simultaneous input to the Arithmetic Logic Unit (ALU) for register to register, double operand instructions. Other duplicate registers are also provided for the Instruction Register, the Program Counter and the data registers, B Register and B Register A on the ALU input. Multiplexors allow the direct transfer of data usually requiring multiple data cycles in other machines; hardware is exchanged for time. Some of the multiplexors receive and transfer data to the semiconductor memory bus with most of the data path optimized to the cycle time of these high speed memories.

The extra logic elements in the data paths would require excessive time were it not for the use of Schottky integrated circuits. These high speed gates reduce throughput delay time, but require the use of multilayer circuit

board with internal ground planes, restricted signal length, and terminators on some signal lines.

The microcontrol section of the PDP-11/45 processor utilizes a buffer register on the Read Only Memory (ROM) for speed. The microbranching network alters the base microaddress before the buffer register; because of multiple timing states the branch is called within the word preceding the branch. The CONTROL BUFFER and the ADRS BUFFER are clocked at different times within a single microword timing state.

SUMMARY CHARTS are provided for the details of implementation and the effects of microprogramming on the various machine designs:

PROCESSOR STATE COMPLEXITY - (Figure 9) indicates the number of control states and data storage required within each processor.

PROCESSOR IC COMPLEXITY - (Figure 10) indicates the number and types of Integrated Circuit (IC) packages used in the processor. The availability of Medium Scale Integration (MSI) devices puts into perspective the contribution of the microprogramming control of the Large Scale Integration (LSI) devices.

PROGRAM CHARACTERISTICS - (Figure 11) compares the use of the control store in each of the processors.

EXPANDABILITY:

All the processors can be expanded along the UNIBUS with peripherals and through UNIBUS switches with other processors. The PDP-11/45 within its own backplane provides a separate microprogrammed Floating Point Processor in addition to the Semiconductor Memory Bus. Only the PDP-11/40 processor provides for expansion of the microprogram directly in terms of words and width. Figure 12 notes the expansion of the microstore for an Expansion Instruction Set, and for a Floating Point Instruction Set. The microword length is also extended in each case to provide control for the extended data path logic. This logic is

placed across the DMUX outgoing bus and the BUS D in going bus noted in the Block Diagram of Figure 7; the microcontrol store is expanded by module interconnecting cables on the wired-or ROM outputs. Within this expansion, the microcontrol bits for basic machine control which are unnecessary to the option, are ommitted.

SUMMARY AND CONCLUSIONS:

Very real advantages existed for each machine in the design phase. The precise order of the microcontrol provides for a direct, specified interface to the data path. Machine operation can be almost completely defined by the flow diagram and data paths. Some initial data path and microlength changes occur early in the design, but by the time the machine is breadboarded, the changes usually concern the microword. Microwords are added, control bits within the words are changed; the debug of the machines reflected this changes and the ease of making them.

Product hardware also benefitted from microprogramming. Each machine directly used microcontrol to meets its design goals. The PDP-11/05 consists of two module boards only because of the LSI influence of the ROM's used for microcontrol and decoding. The major advantage of the PDP-11/40 consists of speed with direct control of the data path reducing machine cycle time. The majority of credit for fewer IC's in the PDP-11/40, however, must go to the general use of MSI and LSI integrated circuits independent of microcontrol. Completely unique and dependent upon microprogramming is the PDP-11/40 expansion possible for the Expansion Instruction Set and the Floating Instruction Set. No other technique, except microprogramming, would allow these simple single module expansions. The PDP-11/45 used microcontrol for speed and some economy in control logic in addition to a segmentation of design. On any large machine, the ability to define segments of design, so as to apply more people effectively, is important.

The only concern for microprogramming comes from production considerations.

Production benefits derive from the lower module counts and the ability to replace modules in the PDP-11/05 and PDP-11/40 machines in production and in the field. Greater effort is then exerted at the module test and repair level (a lower level of production) rather than at the machine test level. The order of the microcode and of the machine also allows some simplification in training.

The only reservation on microprogramming comes from a production aspect of the initial reason for microprogramming. The advantages of standardized Large Scale Integration (i.e. noncustomized) exist mostly at the design and engineering prototype level where the application of a pattern to a Programmable Read Only Memory in an engineering atmosphere is immediate, and faster than changing logic and its associated etch boards. In the Production environment, however, the micropattern in a ROM from a vendor <u>is</u> a custom design. No other customer can use the design, delivery becomes a matter of IC yield with the vendor reluctant to have an inventory of your design. Careful long range purchasing is necessary to insure that all ROM patterns are present at module build. Some slippage can be accommodated by the use of Programmable Read Only Memory but only at a cost in time and money. Given this production-vendor relationship, one of the often mentioned advantages of microcode dissipates: Engineering changes will not be made against microcode, except as a last resort.

REFERENCES:

Processor Handbooks, Digital Equipment Corporation

PDP-11/20/15/r20 Processor Handbook, 1972

PDP-11/40 Processor Handbook, 1972

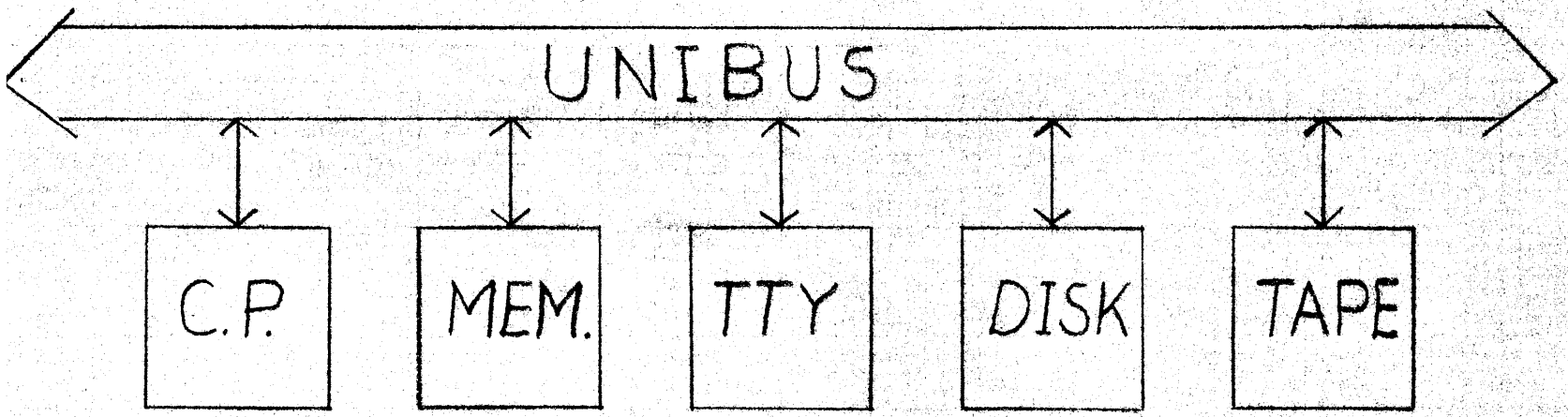
PDP-11/05 Processor Handbook, 1972

PDP-11/45 Processor Handbook, 1972

Microprogramming, Principles and Practices, Samir S. Husson, 1970, Prenctice-Hall

Computer Structures: Readings and Examples, Gordon Bell and Allen Newell, 1971, McGraw Hill

FIGURE 1: UNIBUS



BIDIRECTIONAL
ASYNCHRONOUS, INTERLOCKED
DYNAMIC OWNERSHIP
I/O ADDRESSABLE AS MEMORY
DATA SIGNALS, OWNERSHIP SIGNALS
VECTORED INTERRUPT
STANDARD SPECIFICATIONS

FIGURE 2: INSTRUCTION FORMAT

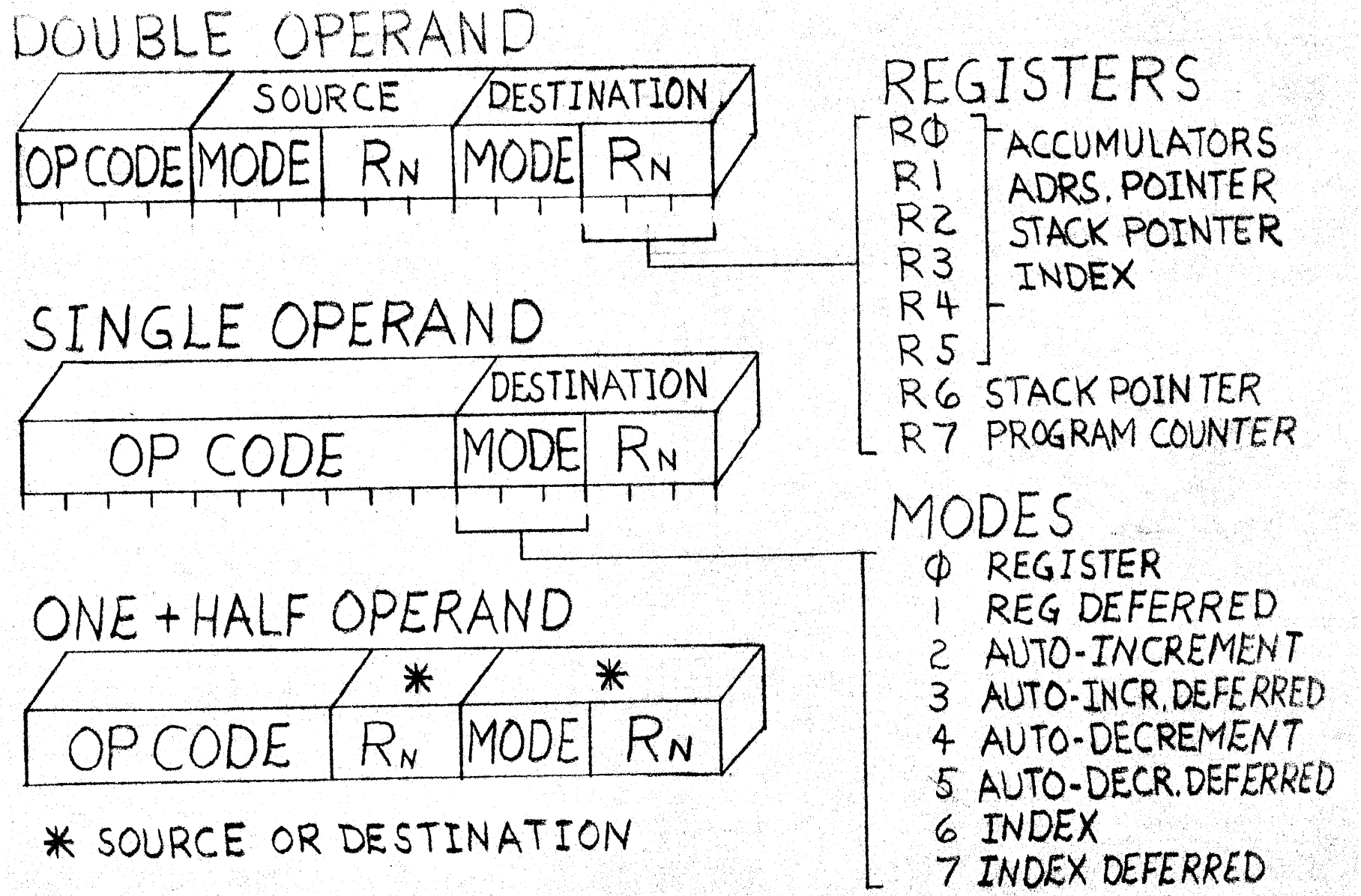
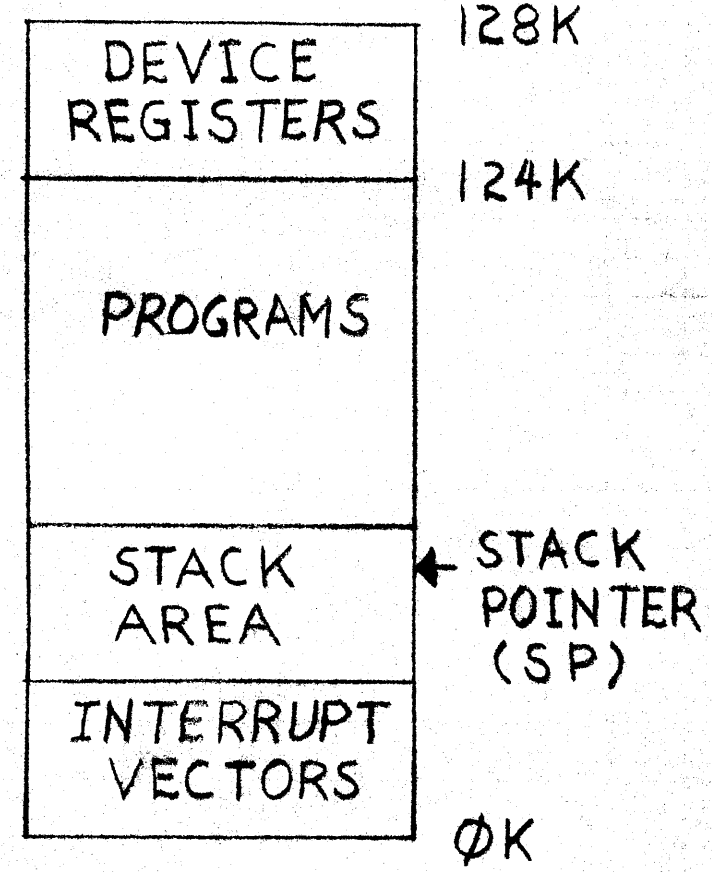


FIGURE 3: STACK OPERATIONS

MEMORY MAP



SP = STACK POINTER
R_N = GENERAL REGISTER
PC = PROGRAM COUNTER
DEST = DESTINATION ADDRESS
PS = PROCESSOR STATUS
VECTOR ADRS FROM PERIPHERAL

STACK ORDERED OPERATIONS

JUMP TO SUBROUTINE

$\downarrow$(SP) ← R_N
R_N ← PC
PC ← DEST

RETURN FROM SUBROUTINE

PC ← R_N
R_N ← (SP)$\uparrow$

INTERRUPT SERVICE

$\downarrow$(SP) ← PS
$\downarrow$(SP) ← PC
PC ← (VECTOR ADRS)
PS ← (VECTOR ADRS +2)

RETURN FROM INTERRUPT

PC ← (SP)$\uparrow$
PS ← (SP)$\uparrow$

FIGURE 4: MICROPROGRAMMTG BREADBOARD

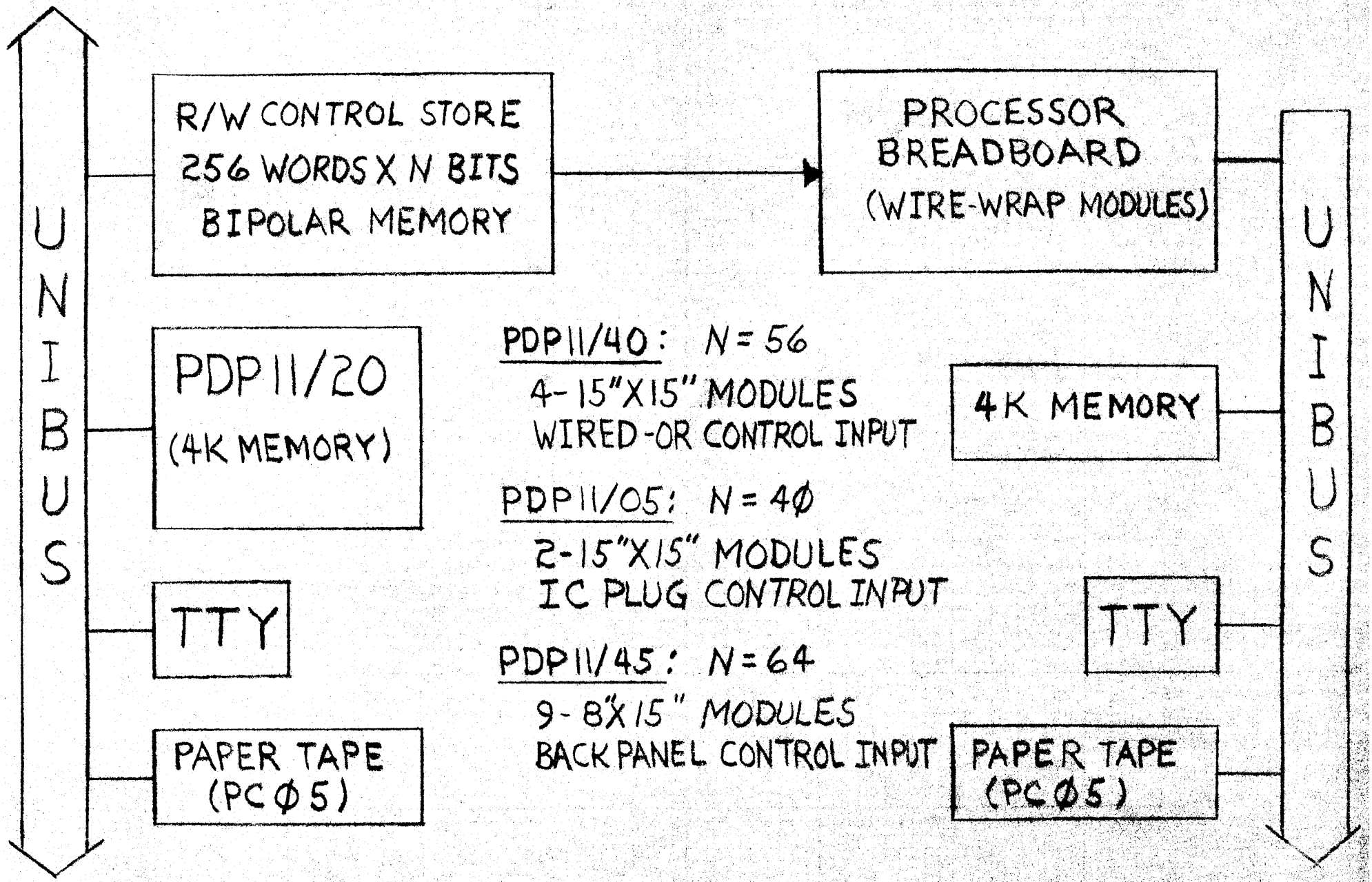
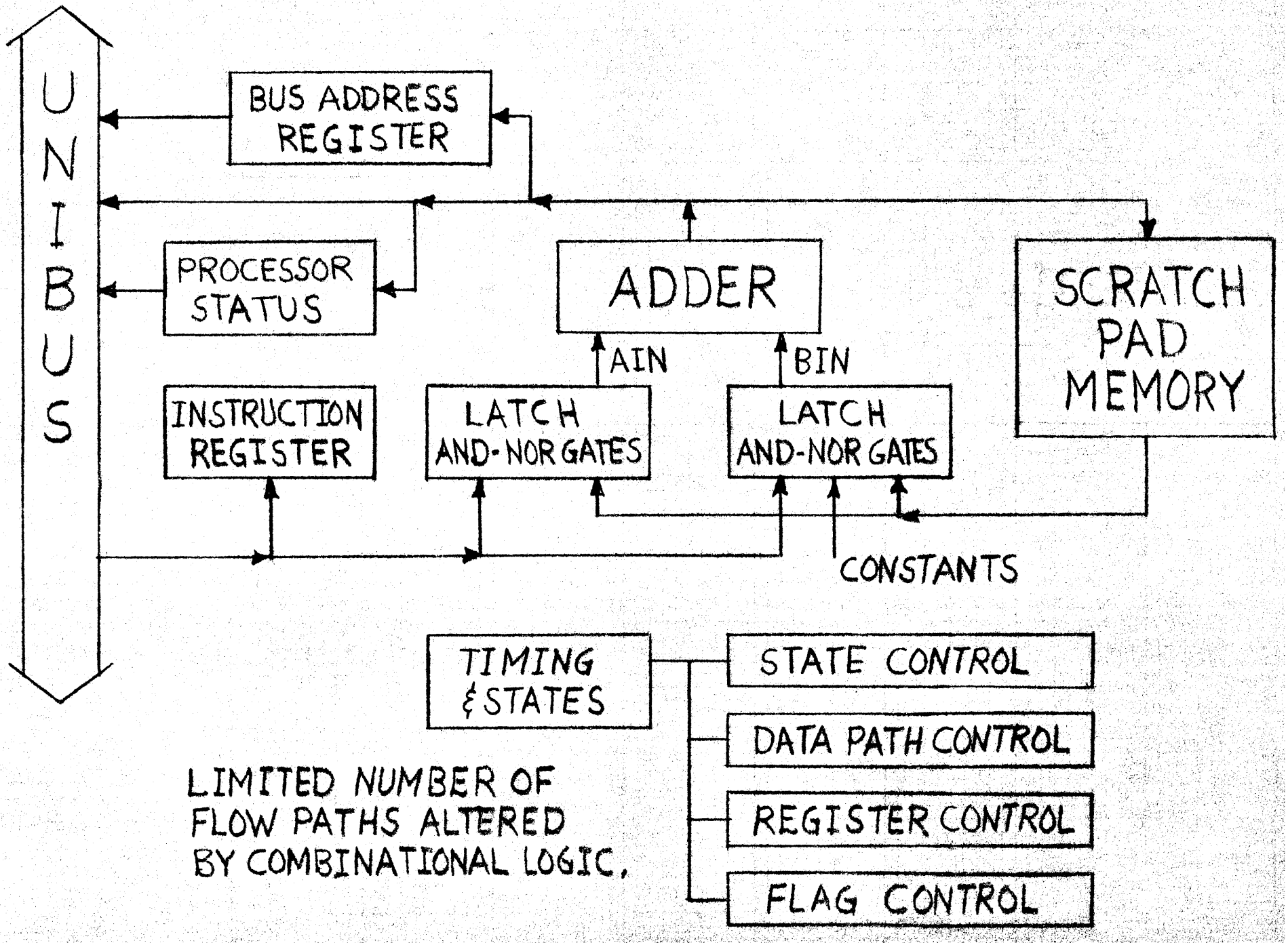


FIGURE 5 : PDP 11/20 BLOCK DIAGRAM



LIMITED NUMBER OF FLOW PATHS ALTERED BY COMBINATIONAL LOGIC.

FIGURE 6: PDP 11/40 BLOCK DIAGRAM

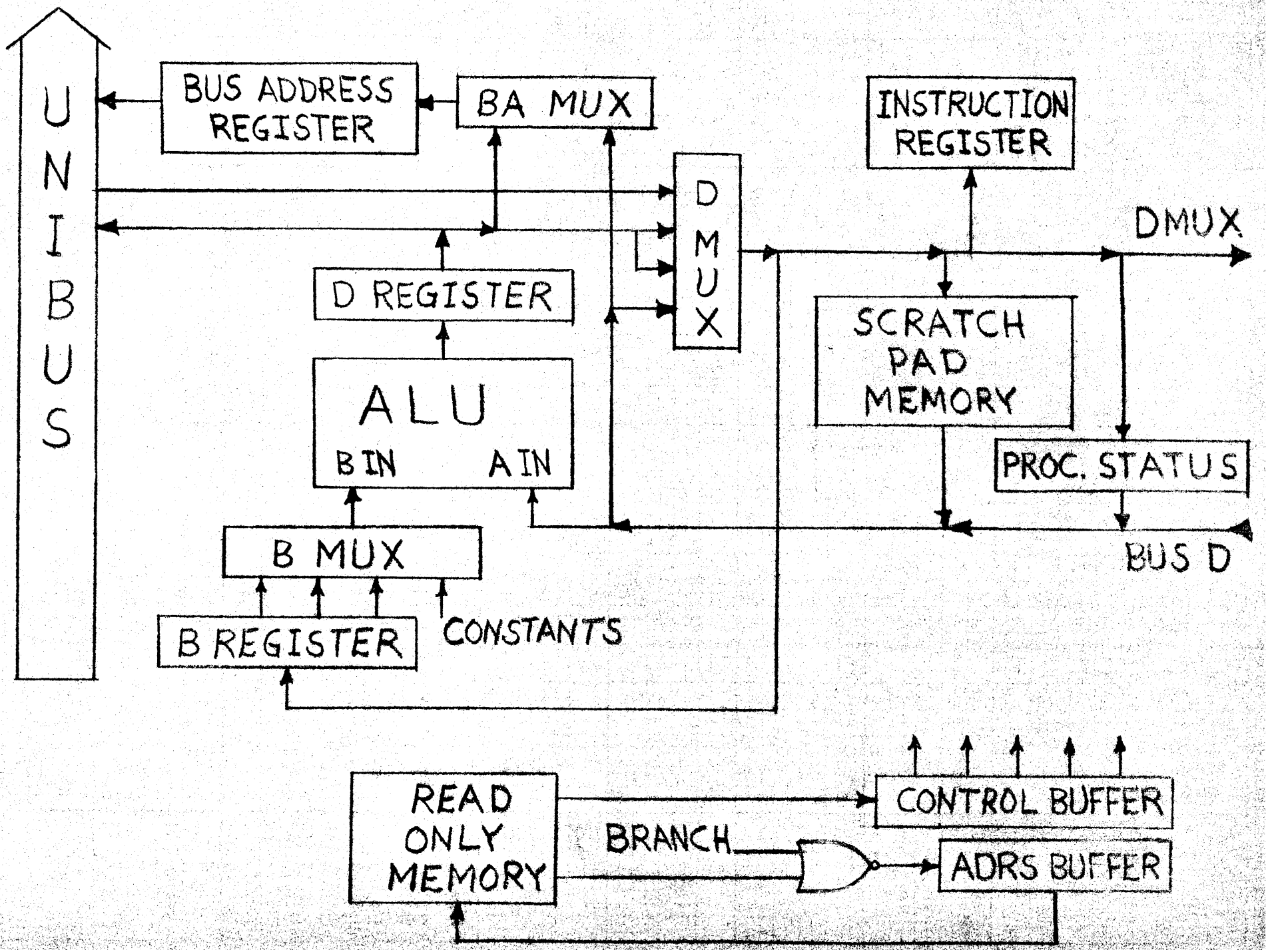


FIGURE 7: PDP11/05 BLOCK DIAGRAM

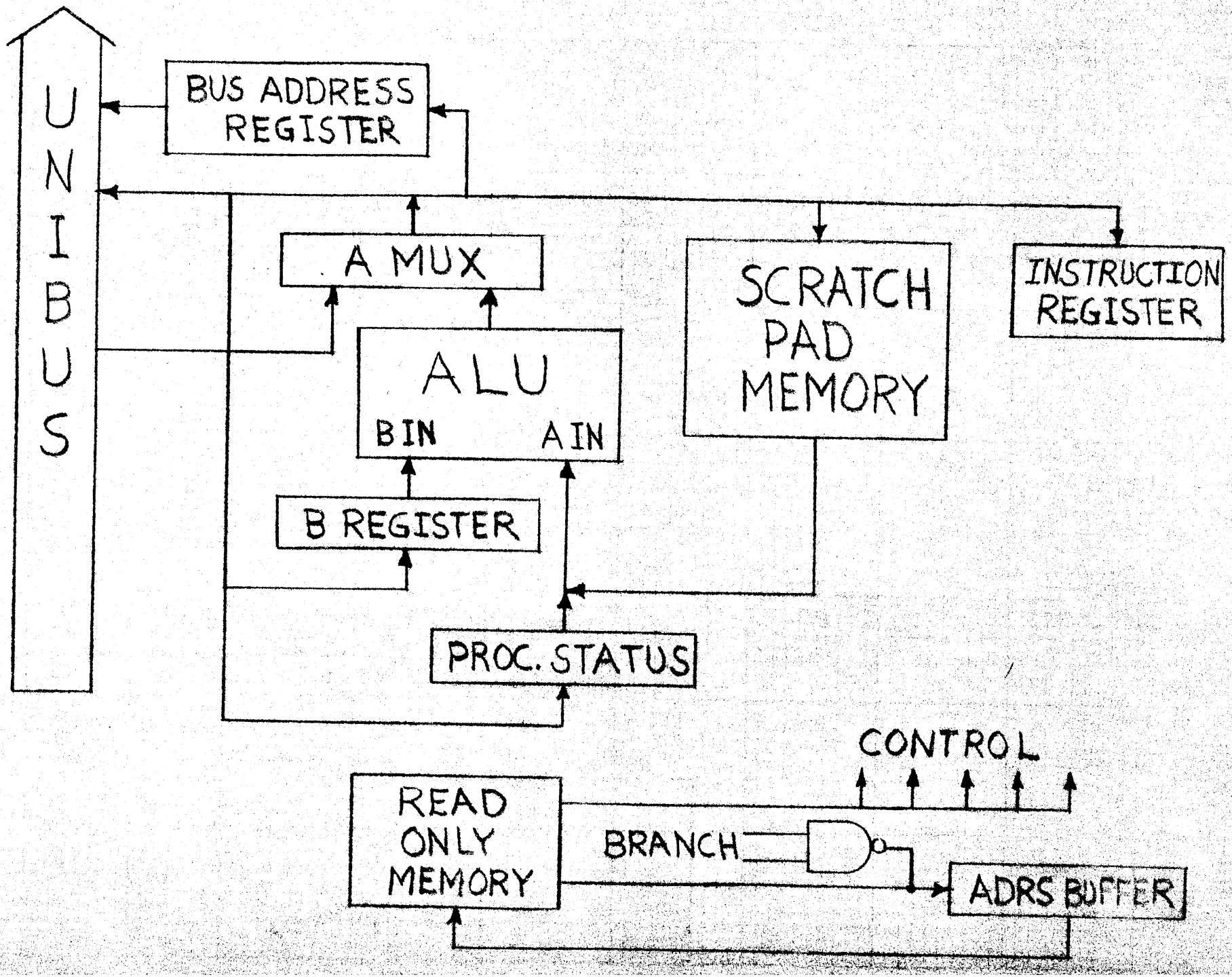


FIGURE 8: PDP 11/45 BLOCK DIAGRAM

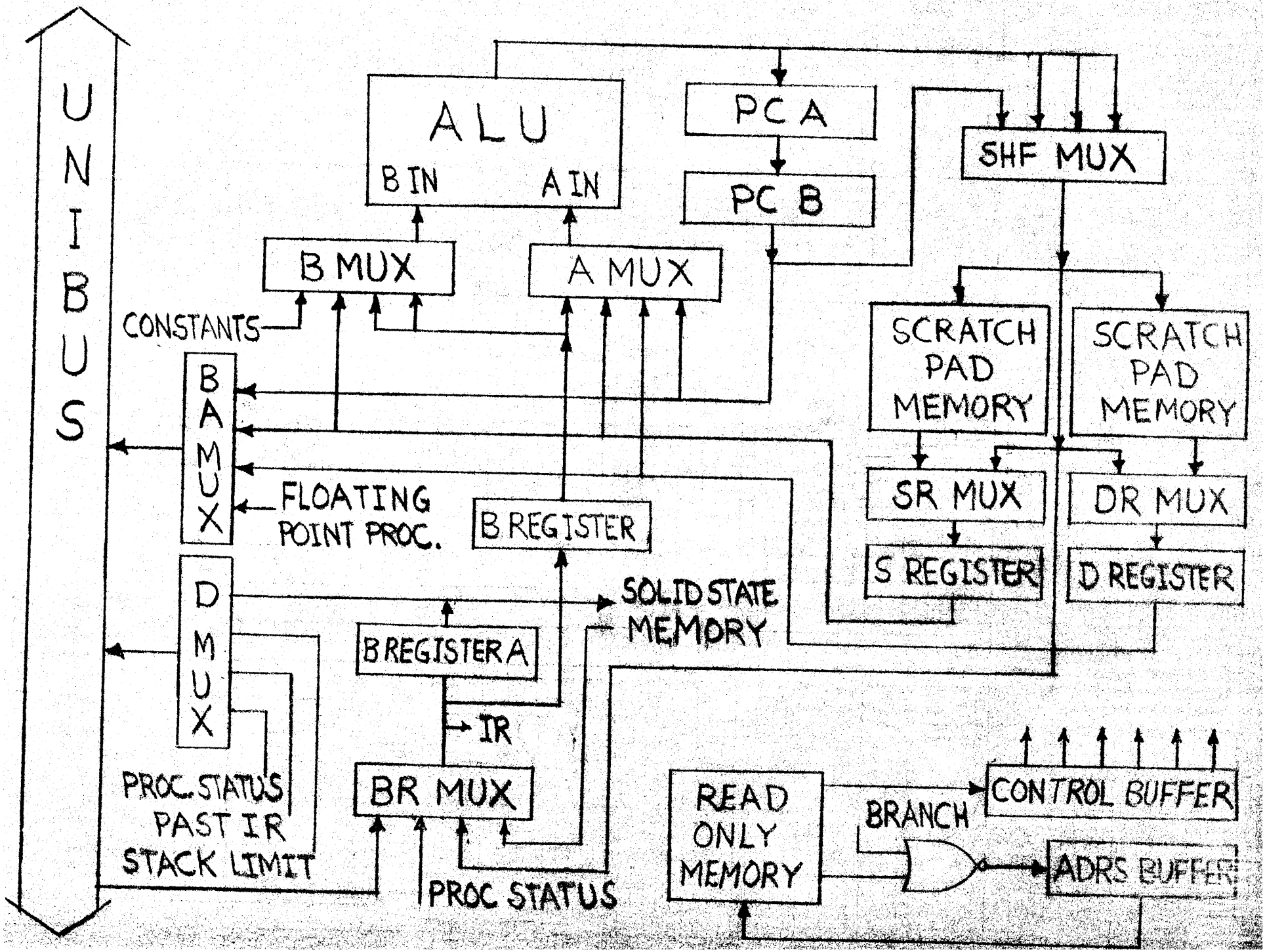


FIGURE 9: PROCESSOR STATE COMPLEXITY

MACHINES:	PDP11/20	PDP11/40	PDP11/05	PDP11/45
CONTROL				
SINGLE FF'S	67	54	65+9*	119
ROM'S (DECODE)	–	–	18 †	3 †
ROM'S (CONTROL)	–	56 BITS X256 WDS	40 BITS X256 WDS	64 BITS X256 WDS
WORD USUAGE		251 WDS	249 WDS	256 WDS
STORAGE				
CONTROL	16X1 (IR)	16X1 (IR) 56X1 WD	16X1 (IR)	16X2 (IR) 64X1 WD
DATA PATH	20 WDS	20 WDS	19 WDS	25 WDS

* THE PDP11/05 INCLUDES ASR INTERFACE.

† ROM'S ARE 8 BITS X 32 WDS.

FIGURE 10: PROCESSOR IC COMPLEXITY

IC'S (PACKAGES) \ PDP'S	11/20	11/40	11/05	11/45
SIMPLE LOGIC (GATES, FF'S)	504	332	129	519
MSI LOGIC (REGISTER, MUX'S)	19 (4%)	63 (15%)	37 (18%)	146 (21%)
LSI LOGIC (ALU'S, ROM'S)	—	22 (5%)	37 (18%)	31 (4%)
UNIBUS	105 (20%)	86 (21%)	50 (25%)	78 (11%)
TOTAL IC'S	523	417	203	696
IC TYPES	27	53	60	78

FIGURE 11: MICROPROGRAM CHARACTERISTICS

ITEMS \ MACHINES	11/05	11/40	11/45
SPEED RATIO*	0.8	1.85	2.04 CORE 3.58 MOS 4.92 BIPOLAR
WORDS	256	256	256
BUFFERED	NO	YES	YES
BITS	40	56	64
DIRECT	40	49	49
ENCODED	4‡	7	15
TIMING	157 NS † 315 NS 630 NS †	140 NS 200 NS 300 NS	150 NS

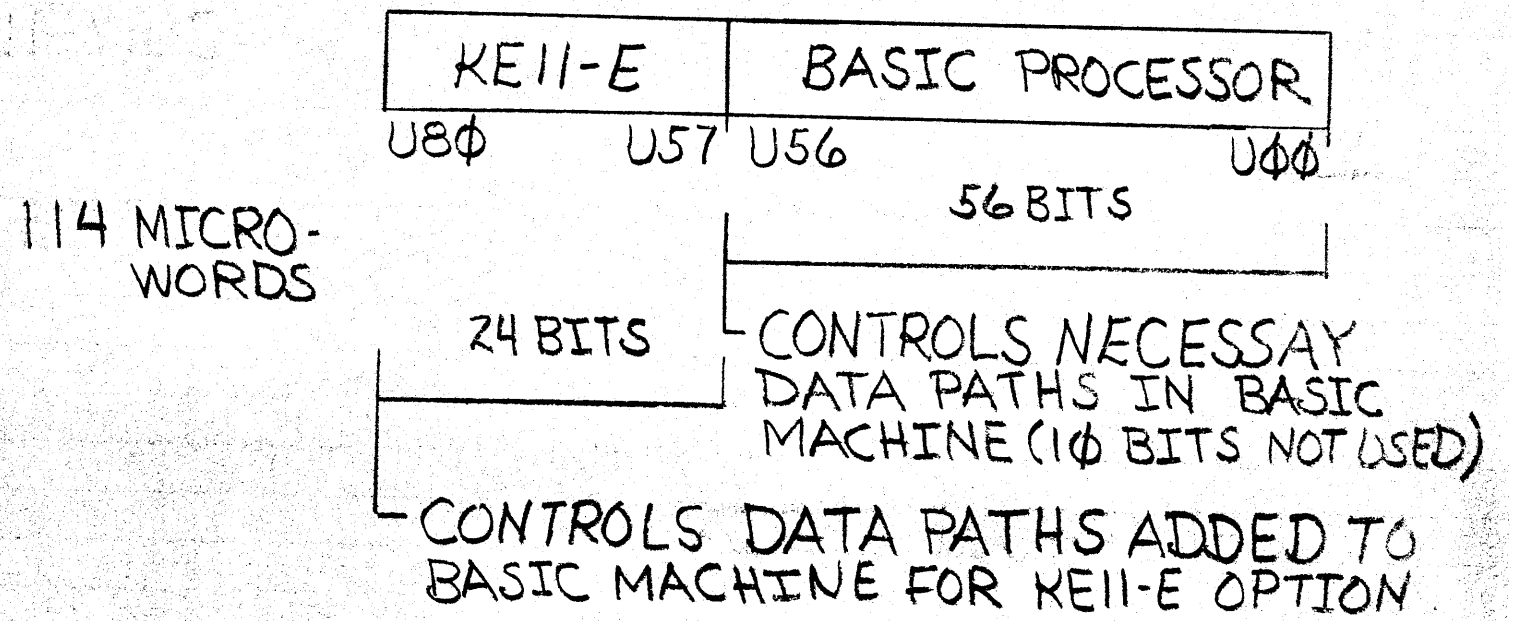
* RATIO OF MACHINE SPEED RELATIVE TO A PDP11/20, COMMON INSTRUCTION MIX, SAME CORE MEMORY.

‡ FOUR MICROPROGRAM BITS HAVE DOUBLE USUAGE.

† SPECIAL CYCLES FOR SHIFT ON BYTE OPERATIONS (157 NS) AND FOR CONDITION CODE ALTERATION (630)

FIGURE 12: PDP 11/40 MICROWORD EXPANSION

KE11-E: EXPANSION INSTRUCTION SET
(MULTIPLY, DIVIDE, ARITHMETIC SHIFTS)



KE11-F: FLOATING INSTRUCTION SET
(FLOATING: ADD, SUBTRACT, MULTIPLY, DIVIDE)