Computer Science Department
114 Lind Hall
Institute of Technology
University of Minnesota
Minneapolis, Minnesota 55455

A PDP-11/40E Microprogramming Primer

by

Helmut K. Berg

Technical Report 78-8
July 1978

Cover design courtesy of Ruth and Jay Leavitt

A PDP-11/40E MICROPROGRAMMING PRIMER

by

Helmut K. Berg
Department of Computer Science
University of Minnesota

Abstract

This report is the first in a series of five reports which document the utilities of the microprogramming laboratory at the University of Minnesota. It is conceived as an introductory tutorial that adopts the terminology used in the microprogramming course offered at the same university. It is intended to familiarize the new user with the hardware and firmware structures of the PDP-11/40E that constitutes the basis of the microprogramming laboratory. It incorporates the experiences gained during the first year of operating the microprogramming laboratory and therefore may also be used as a reference for advanced users. The basic features of the PDP-11/40E are introduced by descriptions of the PDP-11/40 architecture and the architecture of its extension, the WCS 11/40. The micro-level architectures of both constituent components of the PDP-11/40E are informally described to facilitate the understanding of basic PDP-11/40E microprogramming characteristics. The idiosyncrasies of the PDP-11/40E organization are systematically presented to provide a reference to the basics needed by the beginner microprogrammer. In this presentation, emphasis is put on the PDP-11/40 - WCS 11/40 interface and the functional behavior that results from the combination of these components into a PDP-11/40E. Microprogramming characteristics that are implied by the PDP-11/40E hardware idiosyncrasies are collected, explained, and demonstrated by examples.

1. Introduction

The computer science microprogramming laboratory at the University of Minnesota is based on a PDP-11/40E. The PDP-11/40E is a standard PDP-11/40 computer that has been extended by a writable control store option, called WCS 11/40. This control store option was developed at Carnegie - Mellon University, Department of Computer Science and is manufactured and marketed by the 3-Rivers Computer Corporation. A general block diagram of the PDP-11/40E is depicted in Fig. 1.

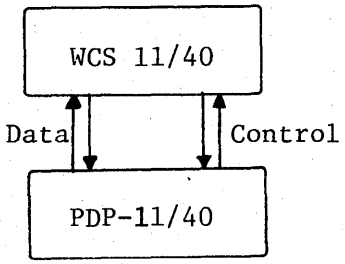


Figure 1: PDP-11/40E

The WCS 11/40 supports user microprogramming of the originally microprogrammed (ROM control store) PDP-11/40 computer and extends the data manipulation capabilities of the basic processor. The addition to the PDP-11/40 is achieved with a minimum of modification to the basic processor. The WCS 11/40 hardware fits into the space that has been reserved in the PDP-11/40 processor cabinet for external processor options (i.e., the extended instruction set, EIS, and the floating point instruction set, FIS). The functions of both options have been microprogrammed for the writable control store and hence, are still available on the PDP-11/40E. All other modifications to the PDP-11/40 processor are minor and can be accomplished by installing about 12 additional wires on the wire-wrap side of the backplane of the circuit board drawer. The data interface between the WCS 11/40 and the PDP-11/40 is organized such that data may be transferred between both modules of the extended processor. The control interface allows only the initiation of the WCS 11/40 from the PDP-11/40, whereas microprograms in the WCS 11/40 control store may execute control over all functional hardware units in the extended processor. Executing control from the WCS 11/40 has no effect on the PDP-11/40 register and control store addressing or on any of the PDP-11/40 processor options, such as memory management, stack limit register, the line frequency interrupt clock etc.

This report is intended to complement the available documentation on the PDP-11/40 computer [1], [2], [3], [4] and the WCS 11/40 [5], [6] by introducing the PDP-11/40E microprogramming characteristics with respect to our microprogramming laboratory. It is followed by a series of four reports [7], [8], [9], [10] which document the utilities of the microprogramming laboratory. Therefore, it is conceived as an introductory tutorial that, on the one hand, adopts the terminology used in the microprogramming course offered at the University of Minnesota and, on the other hand, presents the fundamental characteristics of the PDP-11/40E that are only indirectly reflected by the concepts and structures of the microprogramming utilities which are available in our laboratory. The documentation presented in this and the subsequent reports incorporates the experiences we gained during the first year of operating the microprogramming laboratory.

In this report we introduce the fundamental hardware and firmware structures of the PDP-11/40E. We begin (section 2) with a brief discussion of the PDP-11/40 architecture. To continue, the micro-level architecture of the PDP-11/40 is described in section 3. In section 4, we introduce the architecture of the WCS 11/40 microprogramming option. The description of the PDP-11/40 micro-level architecture is continued in section 5 with a detailed discussion of the WCS 11/40 micro-level architecture. In section 6, the micro-level organization of the PDP-11/40E is described. This description refers to the PDP-11/40 - WCS 11/40 interface, the timing characteristics of the PDP-11/40E, and the microinstruction sequencing and execution. To conclude, some microprogramming techniques which are characteristic to the micro-level architecture and organization of the PDP-11/40E are presented in section 7. The effect of hardware and firmware idiosyncrasies on PDP-11/40E microprogramming are demonstrated by examples.

2. PDP-11/40 Architecture

The PDP-11/40 is a minicomputer manufactured and marketed by Digital Equipment Corporation [1], [4], [11], [12]. It is a member of the PDP-11 computer family which provides the following features.

- Direct addressing of 32K of 16-bit words or 64K of 8-bit bytes allows for efficient handling of 8-bit characters as part of the 16-bit machine word.
- Hardware sequential memory manipulation facilitates the handling of structured data, subroutines, and interrupts through stack processing.

- 16 internal registers, eight of which are machine language visible general purpose registers, may be used for accumulators, address generation, and internal, operational tasks.
- The flexible and powerful set of machine instructions includes single and double operand instructions.
- An automatic priority interrupt system permits grouping of interrupts according to response time requirements. Interrupts are vectored to provide fast interrupt response without device polling.
- Direct memory access (DMA) allows multiple devices to perform direct data transfers to and from main memory.

The conceptual structure of PDP-11 computers is characterized by the UNIBUS concept. As shown in Fig. 2, all components of the computer system communicate via the UNIBUS. The communication via the UNIBUS is asynchronous and based on an interlocked master-slave relationship between the communicating devices. A typical example of this principle is the CPU, as master, directing main memory. The communication protocol for all devices on the UNIBUS is identical. Device registers, main memory locations, and processor registers are assigned unique UNIBUS addresses [1]. Thus, peripheral device registers can be manipulated as flexibly as main memory locations or processor registers. The PDP-11 computer is controlled by the 16-bit general purpose processor. It performs the arithmetic and logic operations, decodes machine instructions, controls the time allocation of the UNIBUS, and controls the interface to the programmer's console. The overall operation of the system is conditioned by the processor status information in the processor status word, PS (UNIBUS address 777776_8).

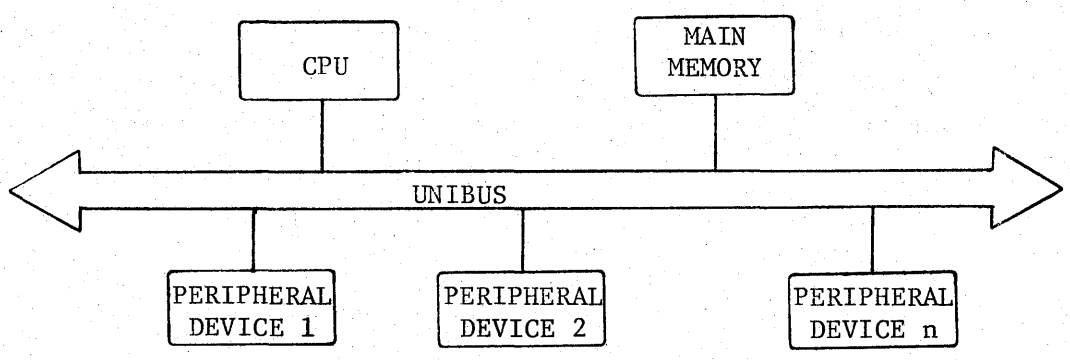


Figure 2: PDP-11 System Block Diagram

The processor registers R[0],...,R[7] are machine language visible. The registers R[7] and R[6] are normally used as program counter (PC) and stack pointer (SP), respectively. SP points to the last entry in a common, temporary storage area in main memory with "last-in first-out" organization. The processor registers R[8],...,R[15] are used by the control unit for the temporary storage of internal values.

With the uniform UNIBUS addressing scheme for device registers, main memory locations, and processor registers, all (over 400) PDP-11 instructions are accomplished by one class of instructions. There is no distinction between memory reference instructions, operate instructions, I/O instructions, etc. The major PDP-11 instruction formats are depicted in Fig. 3. For a detailed

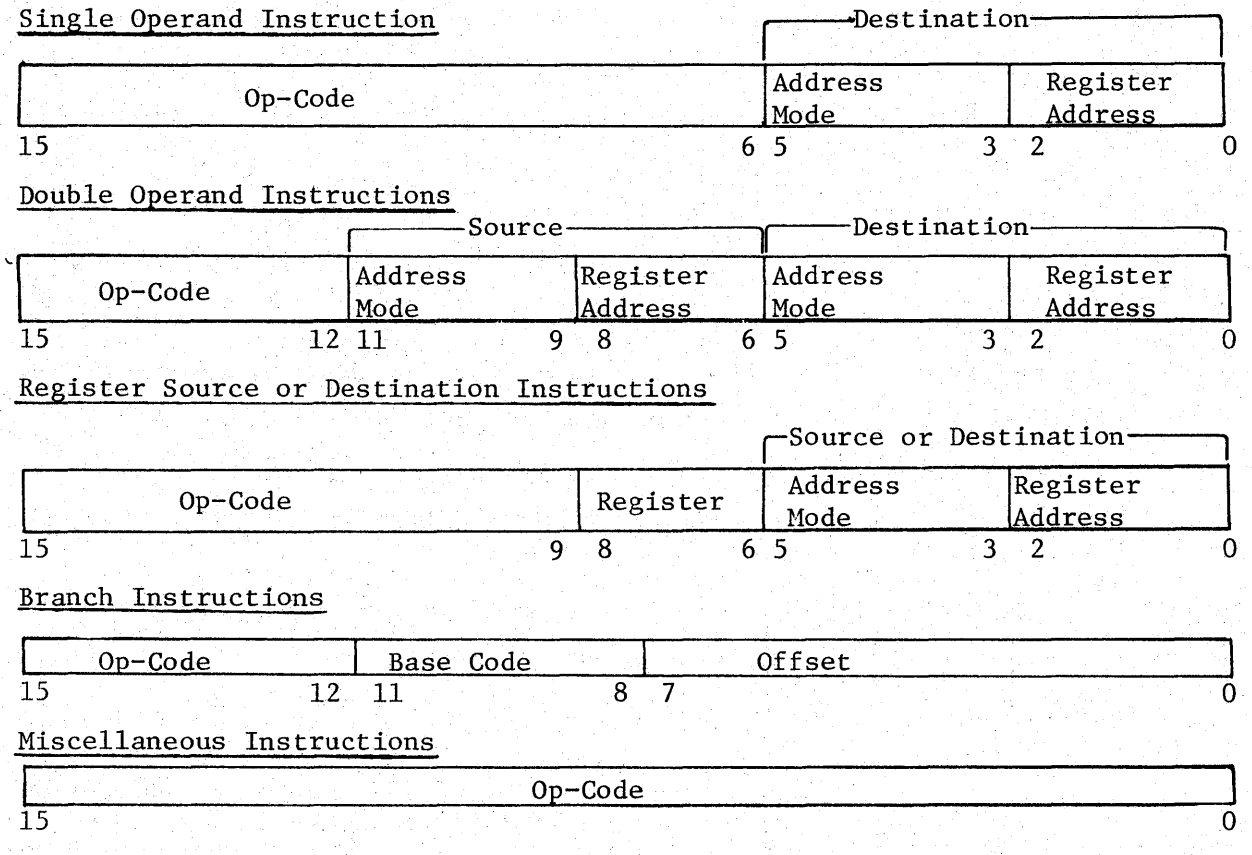


Figure 3: Major PDP-11 Instruction Formats

description of the PDP-11 instruction set, refer to [1]. Much of the power of the PDP-11 instruction set is derived from its wide range of addressing capabilities. As shown in Fig. 3, operand sources or destinations are specified together with an address mode. When the address mode identifies a source or destination by a 16-bit UNIBUS address, this address is stored in the memory word immediately following the corresponding machine instruction. When a general purpose register is identified as source or destination, the register address is stored in the appropriate 3-bit field of the machine instruction. The address mode also distinguishes between byte and word instructions. The addressing scheme for bytes and words in registers and main memory locations is shown in Fig. 4. Low bytes are stored at even-numbered locations and high bytes are stored at odd-numbered locations. Words always start at even-numbered locations.

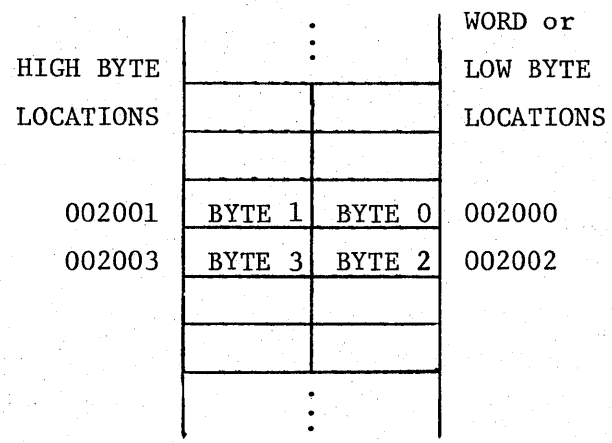


Figure 4: Word and Byte Addressing

The multi-level automatic interrupt system permits the processor to respond to conditions outside the CPU. Every device that is capable of interrupting the processor has two main memory words reserved for its interrupt vector. The first word contains the location of the device service routine and the second word contains the processor status word that is loaded for the execution of the service routine. Before an interrupt service routine is executed, the processor status word, PS, and the current program counter, PC, of the interrupted program segment are automatically saved. Hence, interrupt servicing can be nested in order to serve higher priority interrupts that occur during the servicing of an interrupt with lower priority. Maximum priority is given to DMA devices. The associated non-processor UNIBUS requests do not affect the processor status.

The processor priority (as stored in PS) can be set under program control,thus interrupts can be masked.

Optionally, the PDP-11/40 processor may be equipped with memory management [12]. It is designed to facilitate complete memory management for systems with main memories that are larger than 32K of 16-bit words and to support protection and relocation in systems with multi-user, multi-programming environments. The memory management option has been optimized towards a multi-programming environment and allows the processor to operate in kernel mode (complete control) and in user mode (restricted control). The hardware of the memory management option provides means for dynamic run-time memory allocations upon demand, relocatable page assignments to individual user programs, and control of unauthorized access to pages of user programs.

3. PDP-11/40 Micro-Level Architecture

3.1 KD11-A Processor Structure

A simplified register-transfer diagram of the KD11-A processor of PDP-11/40 computers is shown in Fig. 5 (control paths are not displayed). The components and data paths in the KD11-A processor may be divided into four parts.

- The instruction processor includes the read-only (ROM) control store which provides the microinstructions, the branch logic for microinstruction sequencing, the instruction decoding logic, and the basic timing control logic. It generates the control signals for the data processor, the UNIBUS interface, and the programmer's console and interprets the status bits generated by these controlled functional hardware units.
- The data processor includes the arithmetic and logic unit (ALU), the processor registers, and the data paths with the associated multiplexing and shifting logic. It performs data storage, data transformations, and routing functions as directed by the instruction processor.
- The UNIBUS interface includes drivers and receivers for UNIBUS signal lines, the data transfer control logic, the timing logic, and the priority control logic. It controls the processor regulation of the UNIBUS, data transfers, UNIBUS ownership, and handles special conditions.
- The programmer's console includes the switch register, the data display, the address display, and the console control. It constitutes the user-machine interface that allows the programmer to start, stop, load, modify, continue, and monitor machine language programs.

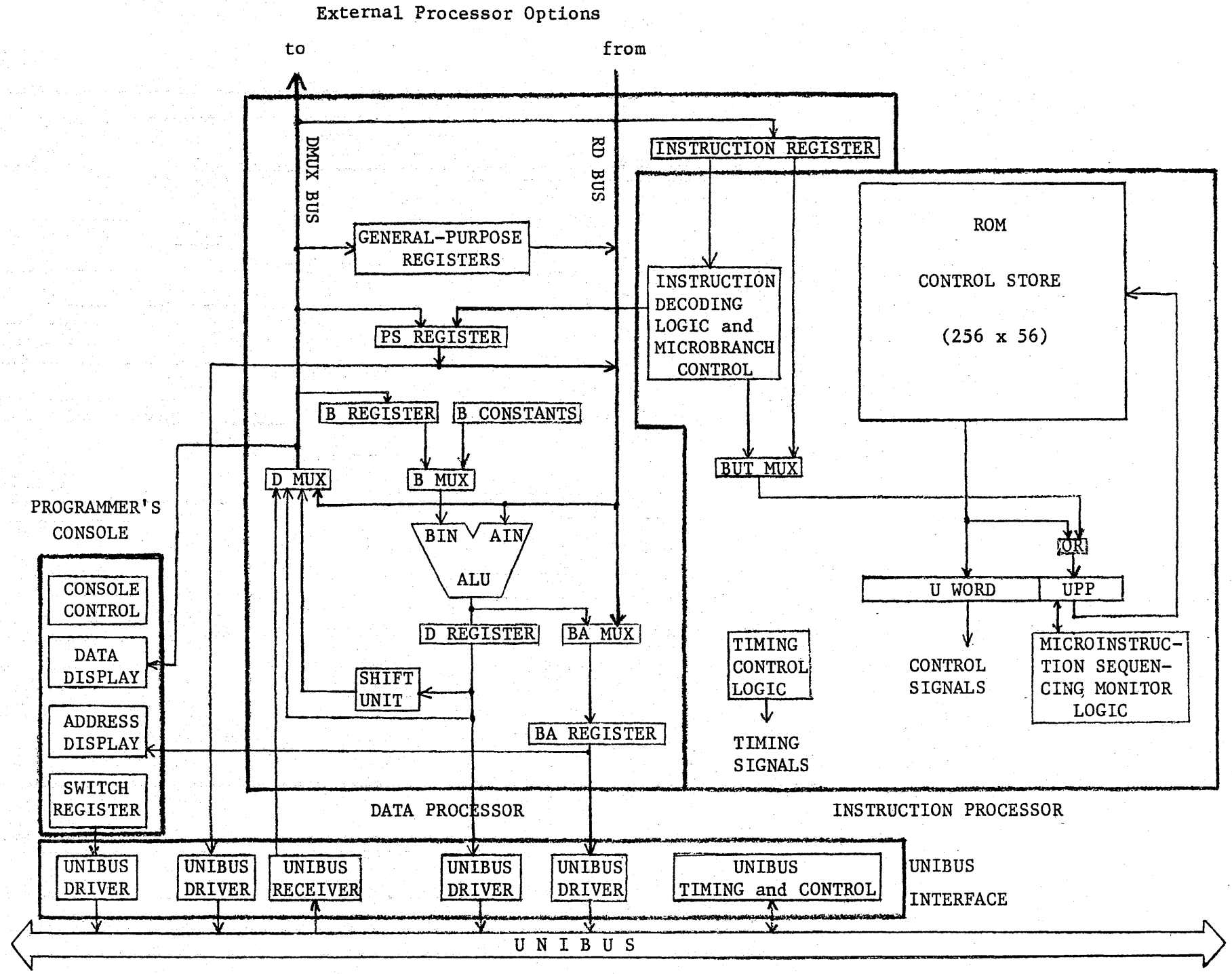


Figure 5: KD11-A Processor Register-Transfer Diagram

3.1.1 Instruction Processor

Subsequently, we discuss the functional hardware units of the KD11-A instruction processor.

ROM CONTROL STORE: Read-only memory (ROM) of 256 56-bit words which holds the microprograms for the interpretation of the basic PDP-11/40 machine instruction set.

INSTRUCTION DECODING LOGIC: Generates control signals which are sent to the data processor, the microbranch control logic, and special purpose combinational logic in the instruction processor.

MICROBRANCH CONTROL: Generates data signals which are sent to BUT MUX and are used to modify microinstruction addresses in UPP.

BUT MUX: (Branch Microtest Multiplexor) selects bits from the instruction register, microbranch control, instruction decoding logic, and machine status flags for the modification of microinstruction addresses in UPP.

U WORD: (Microword Register) 56-bit register to buffer the microinstructions fetched from the control store.

UPP: (Microprogram Pointer) eight low order bits of the U WORD which form an address register whose content points to the microinstruction to be executed next.

MICROINSTRUCTION SEQUENCING MONITOR LOGIC: monitors addresses of previously executed microinstructions to support maintenance, and gates special microinstruction addresses into UPP when exceptional conditions occur.

TIMING CONTROL LOGIC: generates timing signals for the control of synchronous operations in the processor and can be turned off and restarted to accommodate asynchronous processor operations.

The design of the instruction processor of the KD11-A processor is tuned towards the specific PDP-11/40 instruction set and, as a consequence, is not of a general purpose nature. The machine instruction decoding and the microinstruction sequencing are conditioned by the processor status, internal processor flags, selectable timing intervals, asynchronous timing conditions imposed by the UNIBUS interface control logic, machine instruction types and addressing modes, and only

partly by the specification of microoperations in microinstructions. The instruction decoding logic and microbranch control as shown in Fig. 5 consists of an interconnection of several combinational networks with specific tasks. The basic machine instruction decoding provides basic microbranch codes (BUBC) for several branch micro tests (BUT) and signals required by the microbranch control logic, by the condition code control logic and by the ALU control logic. The condition codes to be stored in the PS Register (cf. Fig. 5) are determined on the basis of the machine instruction decoding and the select processor status (SPS) field in the microinstruction under execution. The function of the ALU is not exclusively controlled by the microinstruction because the generation of ALU control signals is conditioned by internal signals generated during machine instruction decoding. Possible alterations of the ALU functions that are specified in the microinstruction are indicated by a specific microinstruction field, called discrete alteration of data (DAD).

Microinstruction addressing is carried out by ORing the address of the successor microinstruction as stored in the microprogram field (UPF) of the microinstruction with the output of BUT MUX into UPP. The inputs to BUT MUX are basic microbranch codes (BUBC) which are provided by the microbranch control logic, the instruction decoding logic, the instruction register and the processor flag control logic. The microbranch control logic generates BUBCs from microbranch control signals which it receives from the instruction decoding logic. The flag control logic monitors a variety of asynchronous conditions and as a result, may change the sequence of processor operations by generating appropriate BUBCs. The BUBC at the input of BUT MUX to be selected for the alteration of UPF is determined by the microbranch field (UBF) in the microinstruction. In addition to the microinstruction address alteration, the microinstruction sequencing monitor logic may modify the microinstruction sequencing by jamming an address into UPP. This mechanism allows for the modification of the microprogram control flow in order to handle exceptional processor conditions (errors, etc.).

3.1.2 Data Processor

The functional hardware units of the KD11-A data processor are described in the following list.

GENERAL-PURPOSE REGISTERS: 16 16-bit scratch pad registers which are loaded from the DMUX BUS, supply output to the RD BUS, and are addressed by 4-bit addresses.

PS REGISTER: (Processor Status Register) 16-bit register which holds the processor status word that specifies condition codes, processor priority, trap condition, and operational modes.

INSTRUCTION REGISTER: 16-bit register which holds machine instructions and whose output is applied to the instruction decoding logic, is used to control microbranching, and specifies general purpose register addresses.

B REGISTER: (ALU B-input Register) 16-bit register for the temporary storage of the ALU B-input.

B CONSTANTS: (ALU B-input Constants) combinational network that provides basic constants used in processor operations.

B MUX: (ALU B-input Multiplexor) selects the B-input of the ALU from either the B Constants, the B Register, or the lower/upper byte of the B Register.

D MUX: (Data Multiplexor) selects the value of the DMUX BUS from either the RD BUS, the D Register, the right-shifted D Register, or the UNIBUS data lines.

ALU: (Arithmetic Logic Unit) 16-bit functional unit which performs 16 arithmetic and 16 logical functions.

D REGISTER: (Data Register) 16-bit register for the temporary storage of the ALU output data.

BA MUX: (UNIBUS Address Multiplexor) selects a 16-bit UNIBUS address to be stored in the BA Register from either the RD BUS or the output of the ALU.

BA REGISTER: (UNIBUS Address Register) 16-bit register for the temporary storage of UNIBUS addresses; it is decoded in order to detect processor register addresses in the UNIBUS addressing scheme; its output is applied to the address display of the programmer's console.

SHIFT UNIT: performs a 1-bit right shift of the D Register output.

RD BUS: (Register Data Bus) 16-bit bus which provides the A-input to the ALU and an input to BA MUX; its input is received from a general purpose register, the PS Register, or external processor options, e.g., extended instruction set (EIS) or floating point instruction set (FIS).

DMUX BUS: 16-bit bus which provides the input to a general purpose register, the PS Register, the B Register, or external processor options (e.g., EIS, FIS).

The data processor of the KD11-A processor offers a high degree of flexibility for data routing. The major sources of ALU operands are the 16 general purpose registers. These registers are loaded from the DMUX BUS and hence, the ALU

output, the shifted ALU output, the RD BUS, and the UNIBUS data lines are direct inputs to thisscratch pad. The output from the general purpose registers is gated onto the RD BUS, such that any of the 16 scratch pad registers can be used as the ALU A-input, the input to the BA Register (via BA MUX), or the input to the B Register (via D MUX). However, the organization of the scratch pad allows only one of the 16 registers to be accessed at a time. Therefore, to use the contents of two general purpose registers as operands of a dyadic ALU operation requires the following steps.

Use of Two General Purpose Registers as ALU Operands

B ← R[i]	Use the B Register as temporary storage of the ALU B-input, R[i].
D ← B o R[j]	Perform the ALU operation, o, with the operands stored in the B Register and in R[j]. The ALU A-input R[j] is directly received from the RD BUS.

In addition to the general purpose registers, the PS Register and external processor options are also sources to the RD BUS. During the execution of a single microinstruction any combination of these sources can independently be gated onto the RD BUS. The result of such a multiple assignment to the RD BUS is the logical ORing of the source data. This effect may be exploited in particular situations, but generally requires special attention to coordinate RD BUS assignments in microinstructions.

The ALU is the heart of the data processor. The A-input is received from the RD BUS and hence, may be the content of one of the 16 general purpose registers or the PS Register, data from an external processor option (e.g., EIS or FIS) or a combination (logic OR) of these sources. The ALU B-input comes from the B MUX which receives inputs from the B Register and from the B Constants. The B MUX can perform the following manipulations that support byte operations:

- the B Register can be directly applied to the ALU B-input,
- the sign of the low order byte of the B Register (B<7>) can be extended into all bits of the high order byte (B<15:8>), while the low order byte (B<7:0>) is directly applied to the ALU B-input,
- the low order byte or the high order byte of the B Register can be duplicated at the ALU B-input,
- the low order byte and the high order byte of the B Register can be swapped.

The B Register, as a source of the ALU B-input receives its input (via D MUX)

from the UNIBUS data lines, the RD BUS, the D Register, or the shifted D Register. Thus, the result of an ALU operation may immediately be reused as an ALU operand.

Immediate Use of a Result of an ALU Operation as an ALU Operand

D ← B o R[i]; B ← D	The result of an ALU operation is temporarily stored in the B Register.
D ← B o R[j]; R[j] ← D	The subsequent ALU operation uses the result of the previous ALU operation.

As the result of an ALU operation can also be stored into the BA Register (via BA MUX), the ALU can also be used to perform address calculations.

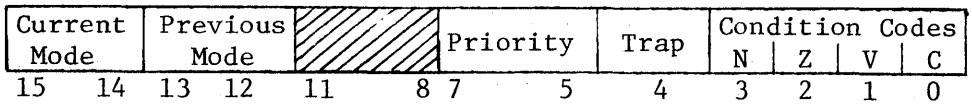
The shift unit at the output of the D Register performs a 1-bit right shift, with a carry bit inserted into the left most bit position. To this end, the D Register has been extended by a highest order bit, called D(C), to store the appropriate carry bit. The shift unit does not provide for left shifts. However, a 1-bit left shift of a general purpose register R[i] is easily accomplished by adding its content to itself.

One-bit Left Shift of a Register R[i]

B ← R[i]
D ← B + R[i]; R[i] ← D

$$\text{Leftshift}(R[i]) = 2 \cdot R[i] = R[i] + R[i]$$

The PS Register as depicted in Fig. 6 has a special position in the KD11-A processor. Its content can directly be gated onto the UNIBUS to facilitate changes of the processor status in the case of interrupts, traps, etc. As its content may also be gated onto the RD BUS, the processor status can be altered in the data processor. Besides being loaded from the DMUX BUS, the PS Register may be altered by the condition code control logic (which is part of the instruction decoding logic and microbranch control shown in Fig. 5).



Legend: Operational Modes: Kernel Mode (00)
Program has complete control of the machine.
User Mode (11)
Certain machine instructions are inhibited.

Processor Priority: The processor may operate on any one of eight (000,..., 111) priority levels.

Trap bit T: When T is set, a trap will occur on completion of the current instruction execution and a new processor status word will be loaded.

Condition Codes: Condition Codes contain information on the result of the last CPU operation.
N = 1, if the result was negative.
Z = 1, if the result was zero.
V = 1, if the operation resulted in an arithmetic overflow.
C = 1, if the operation resulted in a carry from the most significant bit.

Figure 6: KD11-A Processor Status Word

All inputs from the UNIBUS are gated onto the DMUX BUS from where they can be loaded into the B Register, the PS Register, one of the 16 general purpose registers, the instruction register, or the data display. Additionally, UNIBUS inputs can directly be transferred to an external processor option (e.g., EIS, FIS). The contents of the D Register, the PS Register, or the Switch Register may be gated onto the UNIBUS data lines. The address lines of the UNIBUS are loaded from the BA Register. The address display always displays the content of the BA Register. This organization of the data routing to and from the UNIBUS allows each main memory location or device register to be used as an accumulator.

Use of a Main Memory Location or Device Register as Accumulator

BA ← R[i]; DATA-IN; WAIT — Place a Unibus address into the BA Register and Initiate a READ from this address. Wait for the incoming data.

B ← UNIBUS DATA	Place the UNIBUS data into the B Register.
D ← B o R[j]; DATA-OUT; WAIT	Perform an ALU operation with the content of the main memory location or device register as operand, B, and initiate a WRITE to the address stored in the BA Register. Wait for the completion of the WRITE operation before the BA Register or the D Register are modified.

As the address in the BA Register is the same for the READ and WRITE operation, the addressed main memory location or device register is used like an accumulator.

3.1.3 UNIBUS Interface

The following functional hardware units are contained in the UNIBUS interface.

UNIBUS DRIVERS: place the contents of the D Register, PS register, Switch Register, and BA Register on the UNIBUS data or address lines, respectively.

UNIBUS RECEIVER: gates the UNIBUS data lines to DMUX.

UNIBUS TIMING AND CONTROL: handles synchronous data transfers with error checking and correction under the control of the instruction processor; provides asynchronous functions for the control of the UNIBUS communication protocol which is independent of the instruction processor.

The UNIBUS consists of 56 parallel wires which may be grouped as follows.

DATA TRANSFER SECTION

Address	A<17:0>	Carries UNIBUS addresses.
Data	D<15:0>	Carries data.
Control	C0,C1	Specifies data transfer.
Master Sync	MSYN	Timing control for data transfer initiation.
Slave Sync	SSYN	Timing control for data transfer completion.
Parity	PA,PB	Parity bits for the two bytes of data lines.
Interrupt	INTR	Interrupt line.

PRIORITY ARBITRATION SECTION

Bus Request	BR4,BR5,BR6,BR7	Slave requests use of the bus from the master.
Bus Grant	BG4,BG5,BG6,BG7	Master grants use of the bus to a slave.
Non-Processor Request	NPR	Slave requests use of the bus for a data transfer that is not assisted by the processor.
Non-Processor Grant	NPG	Master (processor) grants use of the bus for a non-processor data transfer.
Selection Acknowledge	SACK	Slave accepts bus grant.
Bus Busy	BBSY	Indicates that the data transfer section is in use.

INITIALIZATION SECTION

Initialize	INIT	Reset system by initializing all devices and clearing all error flags.
Power Monitoring	ACLO,DCLO	Initiates power failure interrupt.

The synchronous operations of the UNIBUS timing and control logic control the data transfer section of the UNIBUS. The control lines C0, C1, and MSYN can directly be controlled by the microinstruction fields CBUS and BGBUS, respectively. The control of the UNIBUS priority arbitration section requires asynchronous operations of the UNIBUS timing and control logic. Therefore, the appropriate UNIBUS operations can only indirectly be affected by the microinstruction. Additionally, the UNIBUS timing and control logic independently senses the signals in the UNIBUS initialization section to handle power-on and power-off conditions. Although the PDP-11 data word and operational logic are 16 bits, the UNIBUS addresses are 18 bits. This gap is bridged by the KD11-A addressing logic which automatically decodes the content of the BA Register and generates the appropriate 18-bit addresses. All 16-bit UNIBUS addresses in the range [$160\ 000_8$: $177\ 777_8$] are automatically converted into addresses in the range [$760\ 000_8$: $777\ 777_8$] by setting the high order bits A<17:16> of the UNIBUS address lines to 1. For all other addresses in the UNIBUS address space, bits A<17:16> are set to 0. The 4K of UNIBUS addresses in the range [$760\ 000_8$: $777\ 777_8$] are used as UNIBUS device addresses. These addresses include the KD11-A processor registers. When the addressing logic detects the

address of a processor register, the associated UNIBUS data transfer is inhibited and the specified register access is autonomously performed by the KD11-A processor.

The KD11-A addressing logic may be extended by the memory management option [12] which is located between the KD11-A processor and the UNIBUS address lines. The basic framework for this extension is the fact that the KD11-A word length allows only for address references up to 32K 16-bit words, while the KD11-A addressing logic and UNIBUS address lines can handle address references up to 128K 16-bit words. With the memory management, the normal 16-bit addresses are interpreted as virtual addresses (VA) that contain information to be used in constructing an 18-bit physical address (PA). To this end, VA is combined with relocation and description information as stored in one of the active page registers (APR). The memory management hardware includes two sets of eight 32-bit active page registers which describe and locate the currently active (max.16) memory pages. The size of memory pages may be between 32 and 4,096 16-bit words. This organization permits several user or system programs, each starting at virtual address 0, to reside simultaneously in protected areas of the physical memory space, and it provides the ability to communicate between two areas.

3.1.4 Programmer's Console

Subsequently, we briefly discuss the functional hardware units of the programmer's console.

CONSOLE CONTROL: provides manual functions such as START, HALT, LOAD, ADDRESS, EXAMINE, DEPOSIT, and CONTINUE under the control of the instruction processor.

DATA DISPLAY: consists of 16 indicator lights which display the output of DMUX.

ADDRESS DISPLAY: consists of 18 indicator lights which display the UNIBUS address that is generated from the current content of the BA Register.

SWITCH REGISTER: consists of 18 manually operated switches and 18 indicator lights; its content is gated onto the UNIBUS data lines.

The functions provided by the control switches and the Switch Register of the programmer's console are implemented by microcode routines in the ROM control store. The PDP-11 emulator senses switch activations which effect branches to the appropriate microcode routines. Address input via the 18-bit Switch Register must be stored in the 16-bit BA Register, before it can be gated onto the UNIBUS address lines. This mechanism includes address transformations in the KD11-A addressing logic or the memory management hardware as well as the display of the appropriately transformed address input on the address display. The 18-bit display on the address display represents the content of the BA Register after the conversion by the KD11-A addressing logic. If the system is equipped with the memory management option, the address display shows the constructed 18-bit physical address, PA.

3.2 Microinstruction Format

The KD11-A processor has a horizontal microinstruction format. The horizontal microinstruction format allows for the specification of several microoperations in a single microinstruction that, generally, are carried out concurrently. A single horizontal microinstruction may control all hardware resources in a processor. Furthermore, the control store addresses of successor microinstructions are usually specified as part of horizontal microinstructions. An additional attribute of horizontal microinstructions is that the represented microoperations are not highly encoded and therefore, the microinstructions are relatively wide (60-120 or more bits). As horizontal microinstructions usually allow for simultaneous control of all hardware resources and the specification of microinstruction successor addresses, there is no distinction between operate microinstructions (arithmetic and logic operations, transfer operations, etc.) and branch microinstructions.

The microinstruction word length of the KD11-A processor is 56 bits. The microinstruction is divided into 26 fields. In the design of the WCS 11/40, the logical bit position assignments of the 26 fields in the KD11-A microinstruction (U instruction) have been changed into an extended microinstruction (XU instruction). Particularly, the field specifying the address of the successor microinstruction (UPF) has been moved from the low order bit positions U<7:0> to the bit positions XU<55:48>, in order to allow for the necessary extension of the control store address space. Throughout this text, we adopt the modified PDP-11/40E field assignment in the XU instructions. The microinstruction format of the

basic KD11-A processor microinstruction XU<55:0> and the associated bit position assignments are depicted in Fig. 7.

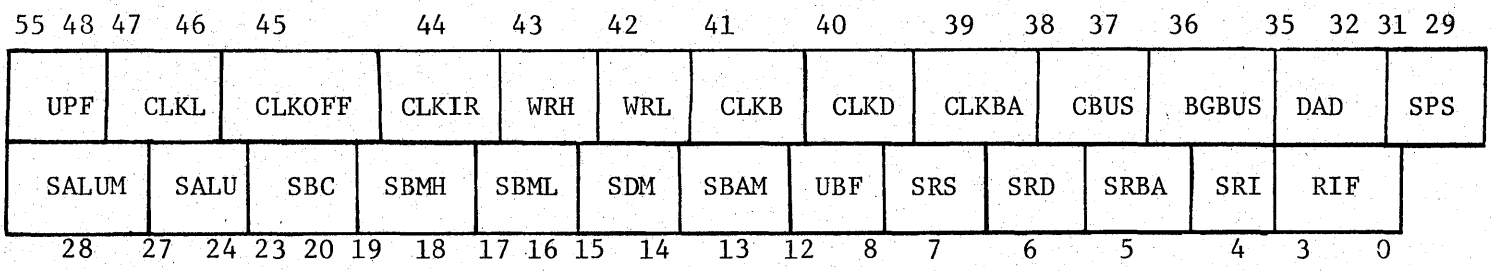


Figure 7: KD11-A Microinstruction Format

The 26 fields in the KD11-A micorinstruction may be divided into 9 groups which are discussed in the following subsections.

3.2.1 Clock Control

CLKL (XU<47:46>): Processor Clock Length Control
Allows the selection from three basic KD11-A processor clock cycles.

CLKOFF (XU<45>): Processor Clock Off
When set, turns processor clock off.

3.2.2 Register Load Control

CLKIR (XU<44>): Clock Instruction Register
Allows clocking the D MUX output into the instruction register.

WRH (XU<43>): Write High Order Byte of DMUX BUS
Allows writing the high order byte of the D MUX output into a selected general purpose register (cf. subsection 3.2.8).

WRL (XU<42>): Write Low Order Byte of DMUX BUS
Allows writing the low order byte of the D MUX output into a selected general purpose register (cf. subsection 3.2.8).

CLKB (XU<41>): Clock B Register
Allows clocking the D MUX output into the B Register.

CLKD (XU<40>): Clock D Register
Allows clocking the ALU output into the D Register.

CLKBA (XU<39>): Clock BA Register

Allows clocking the BA MUX output into the BA Register.

3.2.3 UNIBUS Control

CBUS (XU<38:37>): Control of UNIBUS

Allows the specification of UNIBUS data transfers.

BGBUS (XU<36>): Begin a UNIBUS Transfer

Allows the initiation of a UNIBUS data transfer as specified in CBUS.

3.2.4 Instruction Processor Logic Control

DAD (XU<35:32>): Discrete Alteration of Data

Directs the ALU control logic (which is associated with the instruction decoding logic) as to alterations of operations to be performed by functional hardware units in the data processor.

3.2.5 Processor Status Control

SPS (XU<31:29>): Select Processor Status

Determines loading of the PS Register from the DMUX BUS, clocking of condition codes into the PS Register, and gating of the PS Register onto the RD BUS.

3.2.6 ALU Control

SALUM (XU<28>): Select ALU Mode

Selects ALU mode of operation (arithmetic or logical)

SALU (XU<27:24>): Select ALU Function

Allows the selection of 16 arithmetic or 16 logical ALU functions.

3.2.7 Multiplexor Control

SBC (XU<23:20>): Select Input to B MUX from B Constants

Allows the selection of a constant to be gated to the ALU B-input.

SBMH (XU<19:18>): Select Input to B MUX's High Order Byte

Allows the selection of bytes from the B Register and the B Constants to be gated to the high order byte of the ALU B-input.

SBML (XU<17:16>): Select Input to B MUX's Low Order Byte
Allows the selection of bytes from the B Register and the B Constants to be gated to the low order byte of the ALU B-input.

SDM (XU<15:14>): Select Input to D MUX
Selects the RD BUS, the UNIBUS data lines, the D Register, or the right-shifted D Register as the input to D MUX.

SBAM (XU<13>): Select Input to BA MUX
Selects the output of the ALU or the RD BUS as the input to BA MUX.

3.2.8 General Purpose Register Addressing Control

SRS (XU<7>): Select General Purpose Register Address from IR Source Field
Allows IR<8:6> to be used as a source of a general purpose register address (cf. Fig. 3).

SRD (XU<6>): Select General Purpose Register Address from IR Destination Field
Allows IR<2:0> to be used as a source of a general purpose register address (cf. Fig. 3).

SRBA (XU<5>): Select General Purpose Register Address from the BA Register
Allows BA<3:0> to be used as a source of a general purpose register address.

SRI (XU<4>): Select General Purpose Register Address from RIF
Allows XU<3:0> = RIF to be used as a source of a general purpose register address.

RIF (XU<3:0>): Register Immediate Field
Used as source of a general purpose register address when enabled by SRI.

3.2.9 Microinstruction Sequencing Control

UBF (XU<12:8>): Micro Branch Field
Specifies the branch micro test (BUT) to be performed, in order to generate the address of the successor microinstruction by ORing the determined basic microbranch code (BUBC) into UPP<5:0>.

UPF (XU<55:48>): Microprogram Pointer Field
Used to specify the address of the next microinstruction to be executed. The specified address may be modified as a result of a branch micro test (BUT) specified in UBF.

The discussion of fields in the extended microinstruction (XU instruction) is continued in subsection 5.2.

4. WCS 11/40 Architecture

The WCS 11/40 is a writable control store option for the PDP-11/40 that is manufactured and marketed by the 3-Rivers Computer Corporation. The objectives for the design of the WCS 11/40 are reflected by the following features.

- The provision of 1K 80-bit words of random access (RAM) control store is sufficient to hold user microprogrammed emulators of sophisticated architectures or user microcode to extend or alter the basic PDP-11/40 instruction set.
- The RAM is usable as a scratch pad memory of 5K 16-bit words.
- A 16-word 16-bit stack in the WCS 11/40 is usable for temporary data storage, in addition to the 16 general purpose registers in the KD11-A processor.
- Additional functional hardware units in the WCS 11/40, such as a shift/mask unit and a carry control unit extend the data manipulation capabilities of the KD11-A processor.
- All normal features and options (e.g., memory management, etc.) of the PDP-11/40 are unaffected by the WCS 11/40.

The conceptual structure of the WCS 11/40 is primarily determined by the last of the above features. It is devised as an accessory to the KD11-A processor that allows user microprograms access to all functional hardware units and data paths in the KD11-A processor as well as in the WCS 11/40. Hence, the definition of new machine language instructions may take advantage of all features of the extended processor. User defined machine language instructions can easily be incorporated into the basic set of PDP-11 machine instructions by using undefined PDP-11 operation codes (op-codes). For example, the machine instructions provided by the EIS and FIS processor hardware, which is displaced in the processor cabinet by the WCS 11/40 hardware,have been reinstalled in the form of microcode. Furthermore, the basic PDP-11 instruction decoding may be inhibited, such that the complete set of PDP-11 op-codes is available for the definition of instruction set emulators of arbitrary architectures.

The addressing of the hardware stack and of the RAM is independent of the KD11-A processor hardware. Hence, it is possible to use these added storage

facilities without modifying the uniform UNIBUS addressing scheme of the basic PDP-11. The RAM addresses are byte addresses to be compatible with the PDP-11 addressing scheme, even though the RAM (scratch pad) access is to 16-bit words.

5. WCS 11/40 Micro Level Architecture

5.1 WCS 11/40 Structure

A simplified diagram of the WCS 11/40 hardware is shown in Fig. 8. The hardware units in the WCS may be divided into an instruction processor section and a data processor section. However, this division is logically not very decisive, as the functional hardware units in the data processor section are also used for the calculation of microinstruction addresses.

- The instruction processor section includes the random access (RAM) writable control store and part of the microinstruction address selection logic. As the bipolar RAM is volatile, the basic bootstrap microcode for the WCS 11/40 is stored in a programmable read-only (PROM) control store. The instruction processor section generates control signals for the data processor section.
- The data processor section includes the 16 word 16-bit stack, functional hardware units for microinstruction addressing (which may also be used as an accessory to the functional hardware units of the KD11-A processor) and data paths with associated multiplexing, shifting, and masking logic. Its direct interconnection to the major data paths (DMUX BUS, RD BUS) of the KD11-A processor facilitates data exchange with the basic PDP-11 processor.

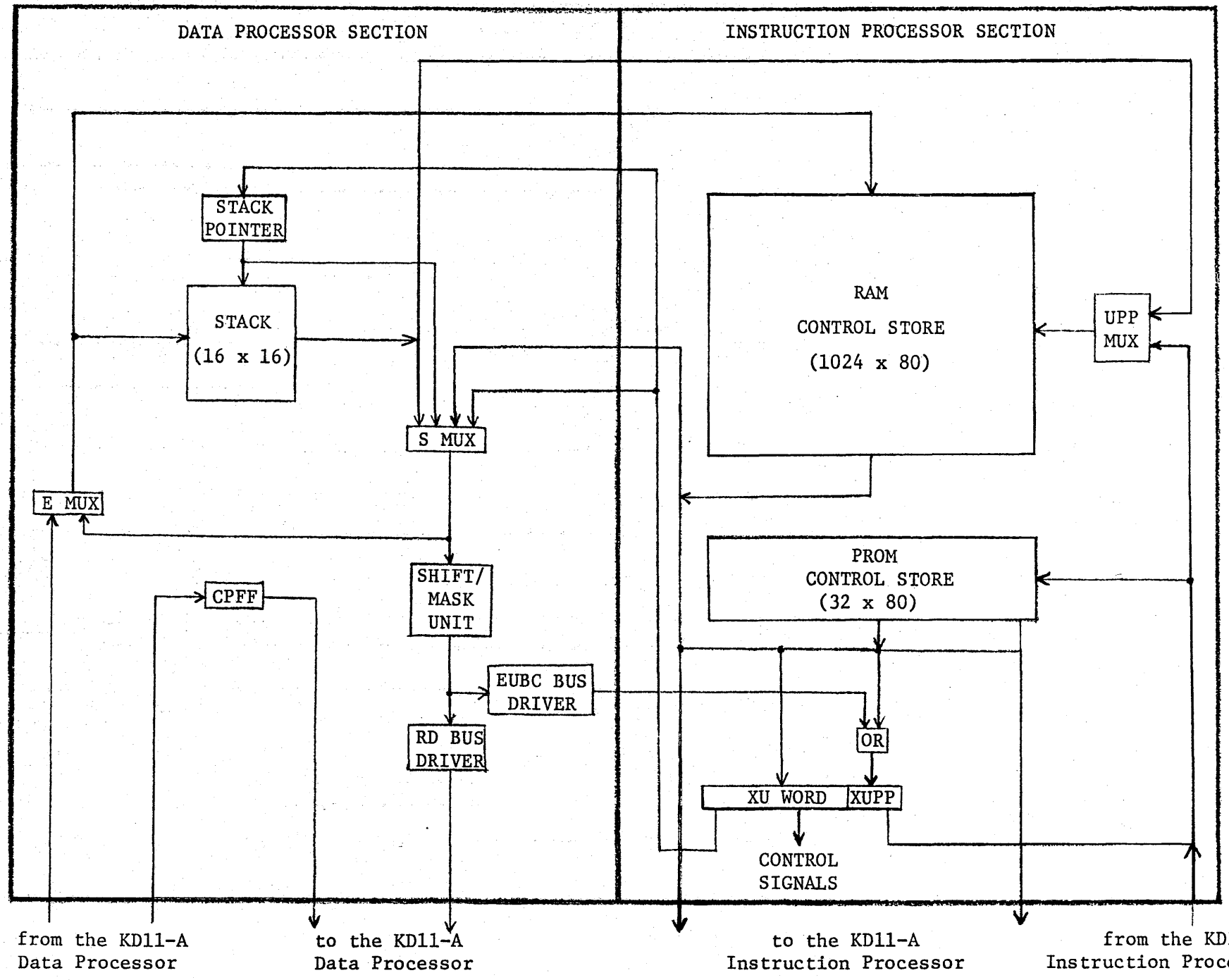


Figure 8: WCS 11/40 Register Transfer Diagram

5.1.1 Functional Hardware Units

In this subsection, the functional hardware units of the WCS 11/40 are briefly described.

Instruction Processor Section

RAM CONTROL STORE: 5K of 16-bit words of bipolar memory which receives 16-bit data inputs and supplies 80-bit words at the output. It can be used to store 1K of 80-bit microinstructions or as a 5K 16-bit data scratch pad.

PROM CONTROL STORE: 32 80-bit words of non-volatile memory that holds 80-bit microinstructions.

XU WORD: (Extended Microword Register) 24-bit register to buffer the 24 high order bits of the 80-bit microinstructions received from the RAM control store or the PROM control store.

XUPP: (Extended Microprogram Pointer) three high order bits of the XU WORD which extend UPP<7:0> of the KD11-A processor into an 11-bit microprogram pointer.

UPP MUX: (Microprogram Pointer Multiplexor) selects 10-bit addresses from the extended microprogram pointer XUPP<2:0>,UPP<7:0> or the WCS 11/40 stack (cf. subsection 6.1.2).

Data Processor Section

STACK: 16-word 16-bit memory that can be used as a data or address push-pop stack. It receives input from E MUX, can supply output to S MUX and UPP MUX, and is addressed from the stack pointer.

STACK POINTER: 4-bit register that holds the stack address. It can be set from the XU WORD (SC<3:0> = XU<75:72>) and can be incremented or decremented for pop or push stack operations, respectively.

E MUX: (E Multiplexor) selects data from the DMUX BUS or the S MUX output to be supplied to the stack or the RAM.

S MUX: (S Multiplexor) selects the 16-bit stack output, the stack pointer, the 16 high order bits of the XU WORD (EMIT field), or one of the five 16-bit fields of the 80-bit RAM output and supplies the selected input to the shift/mask unit or the E MUX.

SHIFT/MASK UNIT: performs right shifts or right/left masking and allows the extraction of any contiguous n-bit field ($n \leq 16$) to be located anywhere in the output to the RD BUS or the EUBC BUS.

CPFF: (Carry Propagate Flip Flop) provides intermediate storage for KD11-A ALU carries.

RD BUS DRIVER: places the 16-bit output of the shift/mask unit on the RD BUS.

EUBC BUS DRIVER: places 11 bits of the shift/mask unit output on the EUBC BUS.

EUBC BUS: (Extended Microbranch Condition Bus) supplies 11-bit microinstruction addresses to XUPP<2:0>,UPP<7:0>.

5.1.2 Operational Characteristics

The RAM control store words are divided into five 16-bit fields. This arrangement is necessary, as data to be written into the RAM are supplied from the KD11-A processor. However, this control store organization also allows for the utilization of the RAM as a scratch pad of 16-bit words. The output obtained by reading the RAM is always an 80-bit word from which the appropriate 16-bit field can be selected through S MUX. The bit assignments of the 16-bit fields in the 80-bit RAM word are shown below.

Field 0: XU<15:0>
Field 1: XU<31:16>
Field 2: XU<47:32>
Field 3: XU<63:48>
Field 4: XU<79:64>

The RAM can be addressed with the extended microprogram pointer XUPP<2:0>, UPP<7:0> or from the top of the WCS 11/40 stack. The RAM is addressed from the top of the stack when it functions as scratch pad.

The addressing of the 16-bit fields in the RAM is logically independent of the KD11-A processor. Therefore, RAM READ/WRITE operations must be controlled by microprograms in the WCS 11/40. As the WCS 11/40 microinstruction allows for access to all functional hardware units in the PDP-11/40E, the ALU and the general purpose registers in the KD11-A processor may be used to microprogram RAM READ/WRITE operations. Data to be written into the RAM is supplied from the KD11-A processor via the DMUX BUS. The RD BUS is used to transfer data read from the RAM into the KD11-A processor. For the execution of RAM READ/WRITE operations, data is usually stored in the general purpose register R[0] and the RAM address of a 16-bit field is stored in R[1].

RAM WRITE

TOS ← R[1]	The address of a 16-bit field of a RAM word is transferred to the top of the stack in the WCS 11/40.
D ← NOT R[0]	Before data can be written into the RAM it must be negated, using the ALU in the KD11-A processor.
RAM[TOS] ← D	The addressed 16-bit field in the RAM is written with the content of the D Register.

RAM READ

TOS ← R[1]	Transfer of the RAM address to TOS.
D ← RAM[TOS]; R[0] ← D	The RAM word containing the 16-bit field to be read is selected by S MUX. It is then gated into the D Register by performing the identity operation in the ALU. The data word read is transferred into R[0] via D MUX.

TOS denotes the current top of the stack as pointed to by the stack pointer, SP.

The stack in the WCS 11/40 can be used as a push-pop stack or as a set of general purpose registers (in addition to the general purpose registers in the KD11-A processor). When it is used as a push-pop stack, pushing (writing) decrements the stack pointer, SP, and places the input value on the stack; popping (reading) increments the stack pointer, SP, and takes a value from the top of the stack. There is no built-in protection against stack overflow or underflow. Furthermore, the content of the 4-bit stack pointer register is undefined after power is first applied. SP can be initialized or set from the SC<3:0> = XU<75:72> field in the XU instruction. When the stack is used as a scratch pad, the current value of the stack pointer, SP, determines the stack location to be accessed. As the stack addressing is independent of the KD11-A addressing scheme, the operation mode (push/pop or scratch pad) of the stack and the stack addressing via the stack pointer need be controlled by the WCS 11/40 microinstruction. Input data to the stack comes from E MUX; output data may go to UPP MUX (RAM addressing), the RD BUS (data stack or scratch pad), or the EUBC BUS (microinstruction sequencing).

Using the Stack from the KD11-A Processor

Push:	S ← DMUX BUS	Push a 16-bit word from the DMUX BUS into the stack.
Pop:	RD BUS ← S	Transfer the current top of the stack onto the RD BUS and pop the stack.

For push/pop operations, the stack location to be accessed is denoted S.

A 16-bit microliteral field (EMIT) in the XU instruction permits the following implementation of a mechanism for the management of nested microprogram subroutines using the WCS 11/40 hardware stack.

S ← EMIT; GOTO<subroutine>	Specify the return address in the EMIT field and push it onto the stack. Jump to the microsubroutine.
EUBC BUS ← S	Pop the stack and place the output on the EUBC BUS which feeds into XUPP<2:0>,UPP<7:0>; return to the calling microprogram.

5.2 Extended Microinstruction Format

The PDP-11/40E microinstruction is an extension of the basic 56-bit U instruction (cf. subsection 3.2) into the 80-bit XU instruction. The additional bits, XU<79:56>, of the XU instruction are used to control the functional hardware units of the WCS 11/40. The microinstruction extension, XU<79:56>, is divided into ten fields whose functionality generalizes the PDP-11/40E micro level architecture as compared to the PDP-11/40 micro level architecture. The format of XU<79:56> is not purely horizontal, but some fields are interpreted by the WCS 11/40 controller which generates the appropriate control signals. This organization allows the user to microprogram mechanisms for microinstruction field extraction and microbranching that are independent of the particularities of the PDP-11 machine instruction interpretation. The bit position assignments of the ten fields in XU<79:56> are depicted in Fig. 9. The fields in XU<79:56> may be divided into the 6 groups which are discussed in the following subsections.

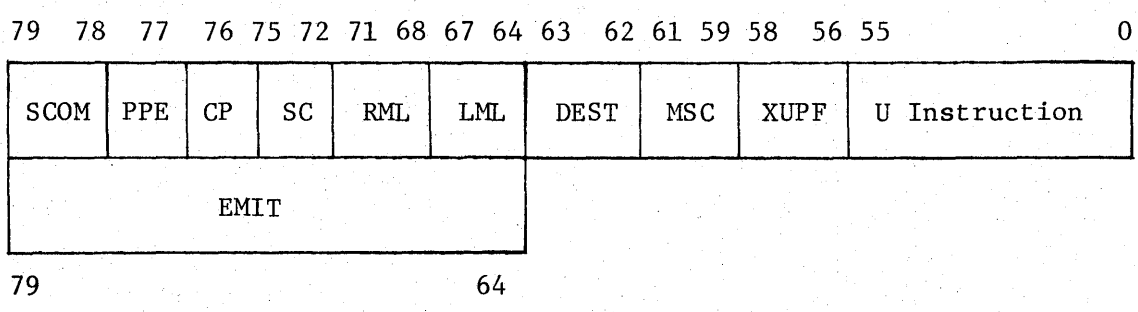


Figure 9: WCS 11/40 Microinstruction Format

5.2.1 Microinstruction Sequencing Control

XUPF (XU<58:56>): Extended Microprogram Pointer Field

Concatenated with UPF (XU<55:48>), this field forms an 11-bit microinstruciton address in the extended control store address space (addition of the RAM and PROM control stores to the ROM control store).

5.2.2 WCS 11/40 Data Paths Control

MSC (XU<61:59>): Mask/Shift Control

DEST (XU<63:62>): Destination

The DEST and MSC fields are combined into a 5-bit field that specifies how the bits XU<79:64> are to be interpreted (as function fields or EMIT field) and how the WCS 11/40 data paths are set up for the execution of the current microinstruction.

5.2.3 Shift/Mask Control

LML (XU<67:64>): Left Mask Limit

Specifies the number of bits of the S MUX output that are to be masked off from the left.

RML (XU<71:68>): Right Mask Limit

Specifies the number of bits of the S MUX output that are to be masked off from the right.

SC (XU<75:72>): Shift Count

Specifies the number of bit positions (between 0 and 15) for a right rotate of the S MUX output.

Depending on the specification of DEST/MSC, this field might also be used to specify a 4-bit value to be transferred into the stack pointer, SP.

5.2.4 Carry Control

CP (XU<76>): Carry Propagate Control

Specifies the application of the content of CPFF to the carry input of the ALU and the storage of a new carry bit (as determined by SCOM) into CPFF.

SCOM (XU<79:78>): Select Carry Out Multiplexer

Specifies the selection of a carry bit from the ALU carry outputs (ALU15, word carry, byte carry) or the condition code bit,

PS(C), of the processor status word that is to be stored into the high order bit extension of the D Register (D(C)) and into CPFF.

5.2.5 Stack Control

PPE (XU<77>): Push/Pop Enable

Specifies if stack read/write operations are combined with pop/push operations, respectively.

5.2.6 Arithmetic and Addressing Constants

EMIT (XU<79:64>): Microliteral Field

Used to specify 16-bit arithmetic or addressing constants. The use of XU<79:64> for microliterals is determined by the specification in DEST/MSC. If XU<79:64> is used as a microliteral field, the fields LML, RML, SC, CP, PPE, SCOM of the XU WORD serve as a data register whose content is transferred to the input of S MUX.

6. PDP-11/40E Micro Level Organization

6.1 KD11-A - WCS 11/40 Interface

The WCS 11/40 hardware is included in the PDP-11/40 processor cabinet. To this end, the processor slots for the optional EIS and FIS hardware are preempted. The interconnection of the WCS 11/40 and the KD11-A processor is facilitated by the fact that the input to the microword register (U WORD) as well as the input and the output of the microprogram pointer (UPP) of the KD11-A processor can directly be connected to the WCS 11/40 boards. Furthermore, the DMUX BUS and the RD BUS of the KD11-A processor, which supply data to and receive data from external processor options(EIS,FIS) respectively, are used to interconnect the KD11-A and WCS 11/40 data paths. This integration of the WCS 11/40 hardware into the KD11-A processor allows the basic processor cycle time to be retained with the addition of the writable control store extension.

6.1.1 Data Processor Interface

The interconnection of the KD11-A data processor with the WCS 11/40 data processor section is depicted in Fig. 10. The DMUX BUS and the RD BUS are the

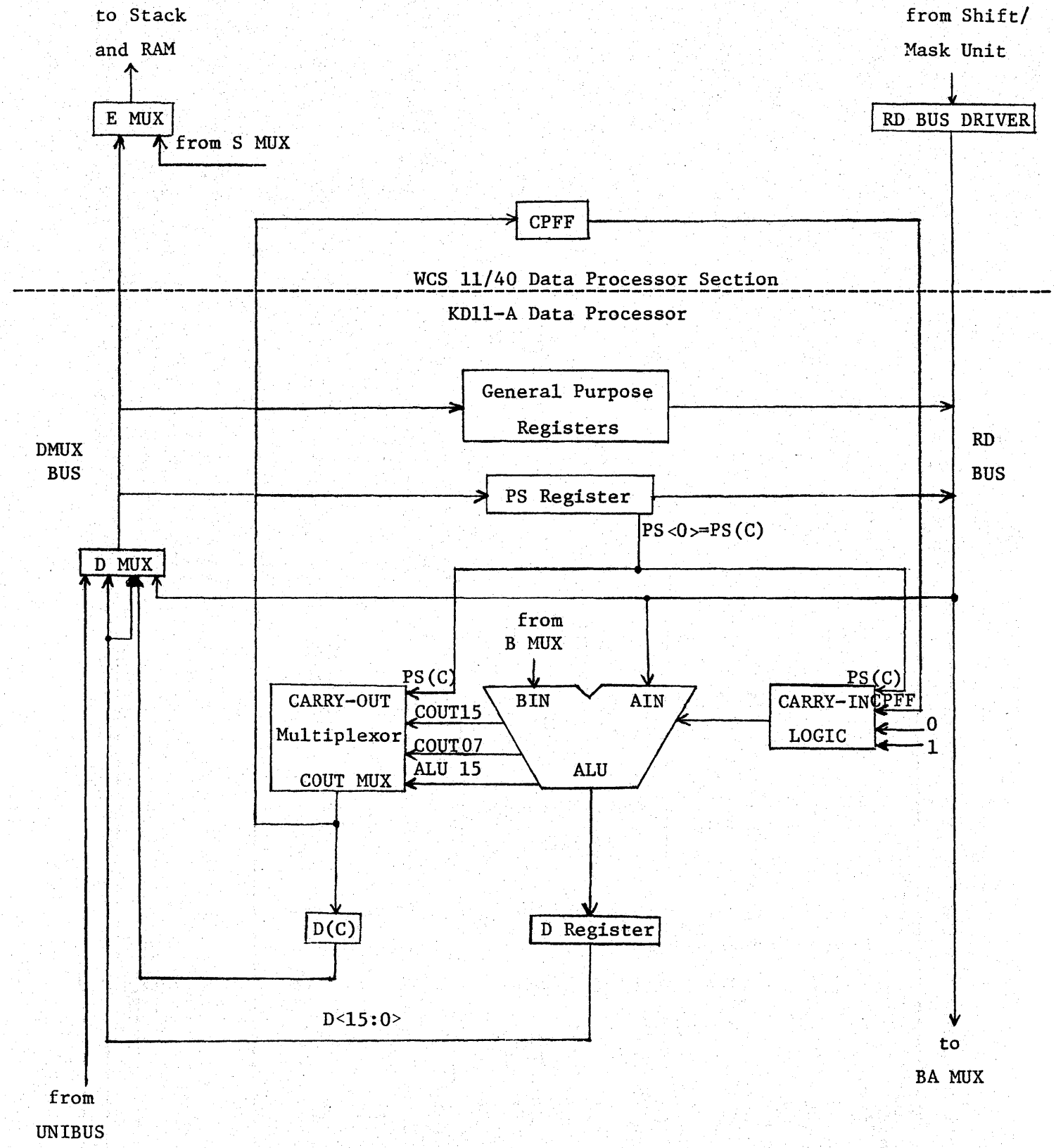


Figure 10: Data Processor Interface

major data paths. Although there are several destinations for data on the RD BUS, timing constraints make the D Register the only destination of data that comes from the WCS 11/40. To store WCS 11/40 data in the D Register requires that the data on the RD BUS be gated through the ALU by performing the ALU identity operation. The transfer of data from the output of the shift/mask unit (stack word, stack pointer, 16-bit RAM field, EMIT) into a general purpose register is described by the following microoperations.

D ← <output of shift/mask unit>; R[i] ← D .

A microoperation of the form,

R[i] ← <output of shift/mask unit> ,

does not work. Data to be transferred into the WCS 11/40 (stack word or 16-bit RAM field) comes from the DMUX BUS. It may be supplied by the D Register, the UNIBUS, a general purpose register, or the PS Register. However, KD11-A data must be complemented in the ALU, before being written into the RAM control store. Therefore, the D Register (ALU output) is usually the source of data to be written into the RAM control store.

The carry propagate flip flop (CPFF) in the WCS 11/40 is connected to the carry input and carry outputs of the ALU. The value stored in CPFF is applied to the carry-in logic (CIN logic) when the carry propagate control (CP) is enabled in the microinstruction under execution. At the end of the microinstruction execution, the output of the carry-out multiplexor (COUT MUX) is stored in CPFF. This feature is particularly useful for micro-programming multiple precision arithmetic operations.

In the basic PDP-11/40, ALU carries are not under direct control of micro-instructions, but are controlled by the ALU control logic which, in turn is directed by the machine instruction decoding logic. The ALU control logic determines the application of the carries 0 or 1 or of the PS Register carry, P(C), to the CIN logic. The WCS 11/40 allows for direct, microprogrammed control of ALU carry inputs via the carry propagate control field, CP, in the extended microinstruction. Furthermore, ALU carry outputs can directly be controlled by the select carry-out multiplexor field (SCOM) in the extended WCS 11/40 microinstruction. COUT MUX selects one of the following carries: COUT 15, COUT 07, ALU 15, and PS(C). COUT 15 and COUT 07 are the carries of ALU word and byte operations, respectively. ALU 15 is the bit 15 output of the ALU. The selected ALU output carry is stored in the high order bit extension of the D Register that is called D(C). When the carry propagate control is enabled,

the ALU output carry is also stored in CPFF.

The shift unit of the KD11-A processor (cf. Fig. 5) is implemented by D MUX which can select as input the UNIBUS data lines, the content of the RD BUS, the content of the D Register, or the content of the right shifted D Register. In the latter case, the vacated high order bit, DMUX <15>, is assigned the carry stored in D(C).

6.1.2 Instruction Processor Interface

The instruction processor interface which connects the KD11-A instruction processor with the WCS 11/40 instruction processor section is shown in Fig. 11. The extended 80-bit microinstruction of the WCS 11/40 (XU instruction) includes the basic 56-bit microinstruction of the KD11-A processor (U instruction). The additional 24 bits of the XU instruction specify control signals for the functional hardware units of the WCS 11/40. The DEST and MSC fields (cf. subsection 5.2.2) are interpreted by the WCS 11/40 controller and determine the usage of the bits XU<79:64>. The microinstruction buffer is divided into two parts, the 56-bit U WORD in the KD11-A processor and the 24-bit XU WORD in the WCS 11/40. Basic microinstructions as supplied by the ROM control store are gated into the U WORD, via the U BUS. Extended microinstructions as fetched from the RAM or PROM control store are supplied to the 80-bit XU BUS. The bits, XU BUX<79:56>, are gated into the XU WORD, whereas the bits, XU BUS <55:0>, are placed onto the 56-bit U BUS.

To accommodate the extended control store in the PDP-11/40E requires the extension of the KD11-A control store address space of 256 words to 2K of addressable control store locations. The appropriate 11-bit control store addresses are specified by concatenating the microprogram pointer fields, XUPF<2:0> and UPF<7:0>, in the XU instruction. That is, the extended microprogram address, XUPA, is defined,

XUPA<10:0> = XUBUS<58:48>

(cf. subsection 5.2.1). Physically, XUPA is implemented as follows,

XUPA<10:0> = XUPP<2:0>,UPP<7:0>
= XU BUS<58:56>,U BUS<7:0>.

The underlying arrangement of the logical bit position assignments in the low-order 56 bits of the XU instruction is defined by the following equivalences,

XU BUS<47:0> = U BUS<55:8> and XU BUS<55:48> = U BUS<7:0>.

Microbranching is effected by ORing bits EUBC BUS<10:8> and EUBC BUS<7:0> of the

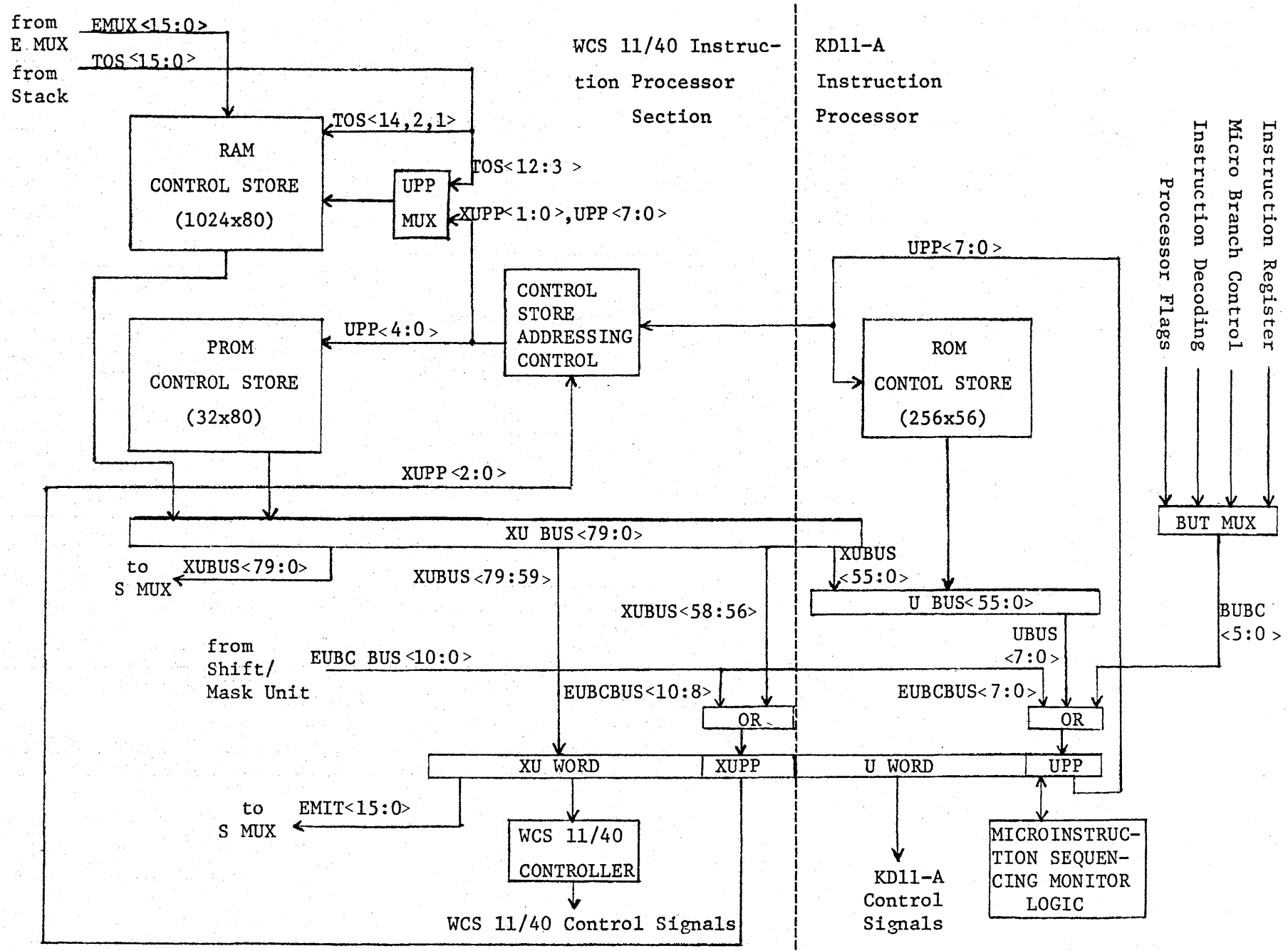


Figure 11: Instruction Processor Interface

11-bit EUBC BUS into XUPP<2:0> and UPP<7:0>, respectively. Furthermore, the basic microbranch codes, BUBC, as selected by BUT MUX are ORed into UPP<5:0>.

The address assignment in the extended control store address space is depicted in Fig. 12.

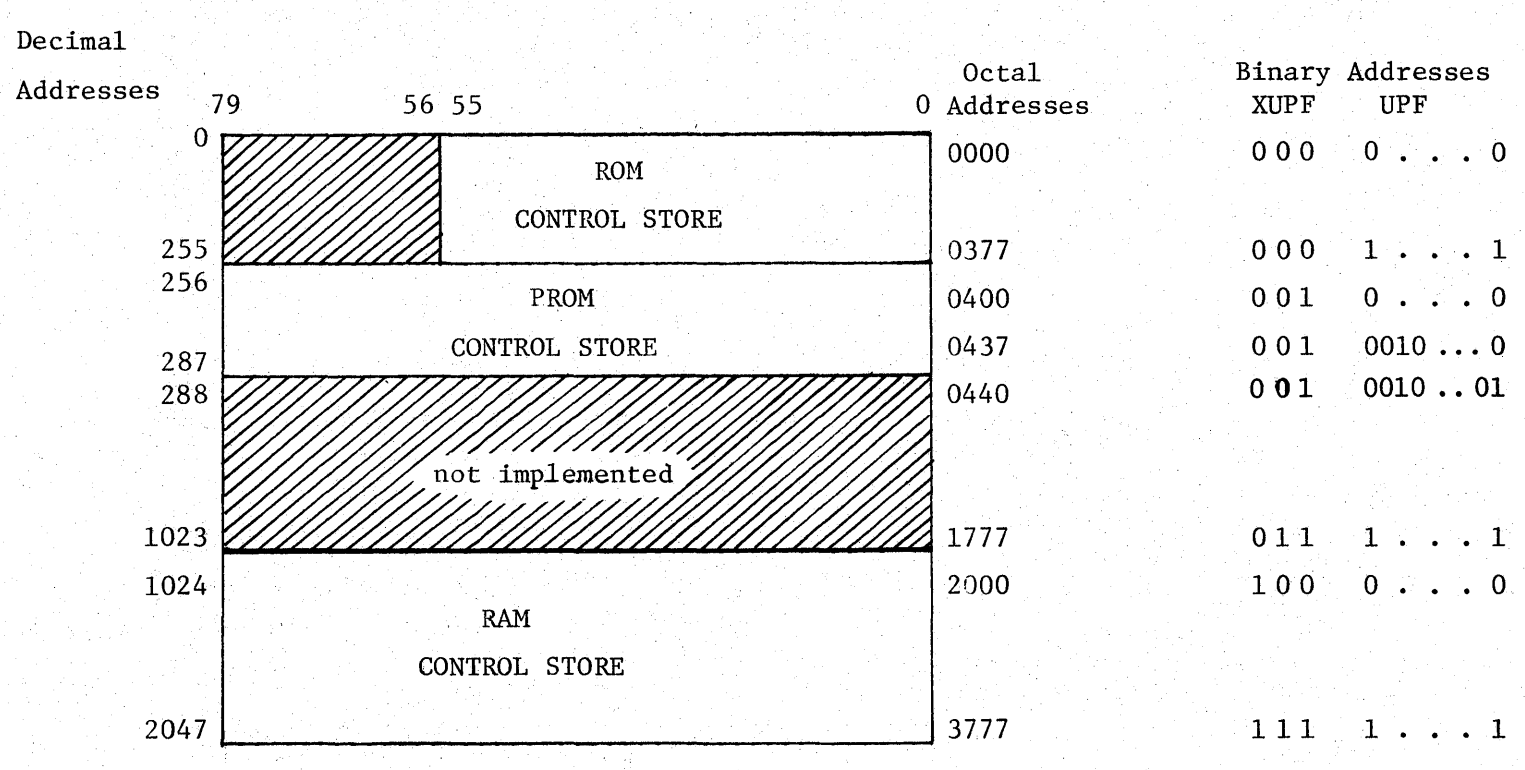


Figure 12: Extended Control Store Address Space

The first 256 words in the extended control store represent the 56-bit word ROM in the KD11-A processor. Hence, the 8-bit microinstruction addresses as stored in UPP can be used to address the ROM, without modifying the KD11-A control store addressing scheme. The PROM control store which holds the basic bootstrap microcode for the WCS 11/40 is implemented in locations 256 to 287 of the extended control store. The next 736 locations are not implemented. The RAM control store is represented by locations 1024 to 2047.

The control store addressing control logic supervises the addressing of the RAM and PROM control stores. It receives as input the 11-bit extended microprogram address, XUPA = XUPP<2:0>,UPP<7:0>. The PROM addressing logic is enabled, when XUPP<2:0> = 000 and an unused machine instruction op-code is detected by the instruction decoding logic. The PROM is also enabled, when XUPP<2:0> = 001,

i.e., whenever XUPA falls in the range 400 to 477. When enabled, the PROM is addressed by UPP<4:0>. The RAM addressing logic is enabled when XUPP<2:0> = 1xx. Microinstructions in the RAM control store are addressed by the 10-bit address, XUPP<1:0>,UPP<7:0>. The values, XUPP<2:0> = 010 and XUPP<2:0> = 011, are illegal.

The multiplexor at the address input of the RAM (UPP MUX) allows for the selection of the 10-bit microinstruction addresses, XUPP<1:0>,UPP<7:0>, or the current top of the WCS 11/40 stack, TOS, to address a word in the RAM. TOS supplies addresses for RAM READ/WRITE operations. To this end, the 16 bits of TOS are interpreted as follows:

TOS<12:3> specifies which of the 1024 RAM locations is to be accessed;
TOS<14,2:1> specifies which of the five 16-bit fields in the selected location is to be read or written;
TOS<15,13,0> is ignored by the addressing hardware.

The address map for the 16-bit fields in the RAM is shown in Fig. 13.

	TOS<14,2:1>					
	100	011	010	001	000	TOS<12:3>
1024	60000	20006	20004	20002	20000	0000
1025	60010	20016	20014	20012	20010	0001
2046	77760	37766	37764	37762	37760	1776
2047	77770	37776	37774	37772	37770	1777
	Field 4	Field 3	Field 2	Field 1	Field 0	

Figure 13: RAM Address Map.

The field addresses, FA, are constructed as follows,

FA = TOS<14>,1,TOS<12>,TOS<11:9>,TOS<8:6>,TOS<5:3>,TOS<2:1>,0

Hence, TOS<14,2:1>= 101, TOS<14,2:1>= 110, and TOS<14,2:1>= 111 are illegal field codes.

For RAM READ and WRITE operations, UPP MUX selects the 10-bit address TOS<12:3> which determines the RAM location to be accessed. In the case of a WRITE operation, the field code, TOS<14,2:1>, is directly applied to the RAM addressing logic. The 16-bit data input comes from E MUX. In the case of a READ operation, the 80-bit RAM word as determined by TOS<12:3> is gated onto XU BUS<79:0>. The appropriate 16-bit field is selected by applying the field code, TOS<14,2:1>, to S MUX.

The bit position assignments in the five 16-bit fields of the 80-bit XU instruction are illustrated in Fig. 14.

Field 0: TOS<14,2:1> = 000

15–14	13	12–8	7	6	5	4	3–0
SDM	SBAM	UBF	SRS	SRD	SRBA	SRI	RIF

Field 1: TOS<14,2:1> = 001

15–13	12	11–8	7–4	3–2	1–0
SPS	SALUM	SALU	SBC	SBMH	SBML

Field 2: TOS<14,2:1> = 010

15–14	13	12	11	10	9	8	7	6	5	4	3–0
CLKL	CLKOFF	CLKIR	WRH	WRL	CLKB	CLKD	CLKBA	CBUS		BGBUS	DAD

Field 3: TOS<14,2:1> = 011

15–14	13–11	10–8	7–0
DEST	MSC	XUPF	UPF

Field 4: TOS<14,2:1> = 100

15–14	13	12	11–8	7–4	3–0
SCOM	PPE	CP	SC	RML	LML
EMIT					

Figure 14: 16-bit Fields in the XU Instruction

6.1.3 Control Interface

As the WCS 11/40 is implemented as an external processor option of the PDP-11/40, special provision must be taken to transfer control to micro-programs in the writable control store. The control transfer is implemented such that the microinstruction at location 0 of the PROM control store (location 0400 in the extended control store address space) is executed, when an unused PDP-11 machine instruction op-code is decoded while control is executed from the ROM in the KD11-A processor. The unused 16-bit PDP-11 op-codes are listed in Fig. 15.

Op-Code(octal)	Number
0 0 0 0 0 7	1
0 0 0 2 1 0 ⋮ 0 0 0 2 2 7	16
0 0 7 0 0 0 ⋮ 0 0 7 7 7 7	512
0 7 5 0 4 0 ⋮ 0 7 6 7 7 7	992
1 0 6 4 0 0 ⋮ 1 0 6 4 7 7	64
1 0 6 7 0 0 ⋮ 1 0 7 7 7 7	576
	2,161 Total

Figure 15: Unused PDP-11 Op-Codes

Potentially, there are 2,161 unused 16-bit op-codes. However, this number of available op-codes decreases considerably, when, for example, single or double operand machine instructions of the formats depicted in Fig. 3 are to be implemented. Nevertheless, the organization of the KD11-A - WCS 11/40 control interface is flexible enough to allow for the implementation of complete machine instruction set emulators.

As the bipolar RAM control store is volatile (i.e., it looses its content whenever power is removed), the non-volatile PROM control store is located at the entry to the WCS 11/40 control store. The 32-word PROM contains bootstrap microcode for loading and reading the RAM as well as for decoding unused op-codes to provide entries into the RAM control store. The bootstrap microcode in the PROM is based on the following organization.

- The machine instruction with an unused op-code that causes a control transfer to control store location 0400 is automatically loaded in R[13], the B Register and the instruction register.
- The unused op-code, 000007 (cf. Fig. 15), is reserved for RAM READ/WRITE operations.
- Three general purpose registers are used for RAM READ/WRITE operations: R[0] contains the data, R[1] contains the RAM address in the format discussed in subsection 6.1.2, and R[2] contains a specification of the RAM access. (RAM WRITE: R[2] = 0; RAM READ: R[2] ≠ 0).
- The RAM control store may be entered at locations 2000 or 2001. Entry point 2001 is used when the KD11-A processor is in user mode (PS<15:14> = 11). When the KD11-A processor is in kernel mode (PS<15:14> = 00) and the detected unused op-code is not 000007, control is transferred to entry point 2000.

The bootstrap microcode in the PROM control store performs the functions of the following program.

TOS ← PS

To test the processor status, the processor status word, PS, is loaded into TOS.

IF TOS<15> = 1 THEN TOS ← R[13]; GOTO 2001

The KD11-A processor is in user mode if PS<15> = TOS<15> = 1. In this case, the machine instruction that caused the control transfer is loaded into TOS and control is transferred to the entry point 2001 of the RAM control store. Remember that the machine instruction that caused the control transfer is stored in R[13].

ELSE D ← 7 XOR B

The detected unused op-code is compared with 000007. If the unused op-code is 000007, the value 0 is assigned to the D Register. Remember, the machine instruction that caused the control transfer is also located in the B Register.