Stallings 10th Edition
CS 320
Ch. 21 Microprogrammed Control

Microprogramming is a t**echnique to implement the control system of a CPU using a control store to hold the microoperations.**

Microprogramming was invented by **Maurice V. Wilkes in England in 1951.**

The first commercially successful microprogrammed machines were created by **IBM in IBM-360 series in 1964.**

Although microprogramming was invented in the early fifties it not widely used until the mid-sixties because ROM technology was **expensive or unavailable.**

The obvious disadvantages of a hardwired controller include: **inflexible design, hard to design correctly in the first place, and difficult to upgrade.**

A microprogram is a program of microinstructions which are a set of microoperations.

Firmware is another name for m**icrocode – mix of hardware and software.**

Microinstructions can be horizontal or vertical.
Horizontal control implies that each bit in the microinstruction directly effects some hardware event.
In vertical microprogramming the bits are encoded so that a decoding stage is required between the microinstruction and the hardware.

A control store (or control memory) is a l**ocation where all microinstructions are stored. This is analogous to program memory.**

Figure 21.2 page 733 shows a control memory with microinstruction sequences. **Each sequence represents on assembly language statement. When this is complete you typically jump to an instruction fetch to get the next instruction.**

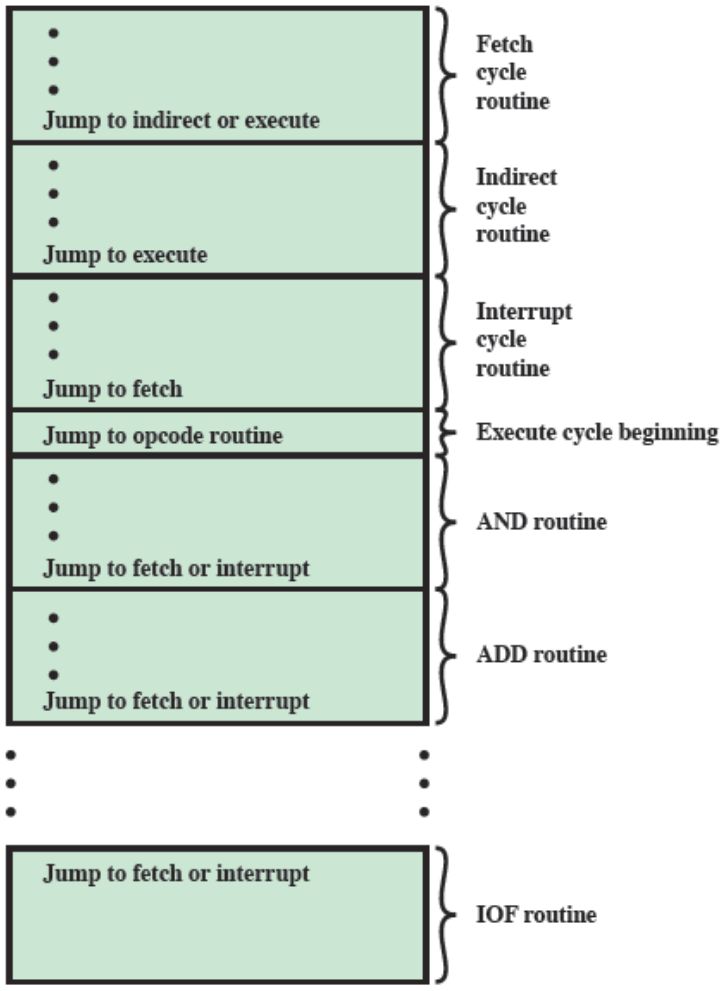


Figure 21.2 Organization of Control Memory

Stallings 10th Edition
This is a microprogrammed control unit.

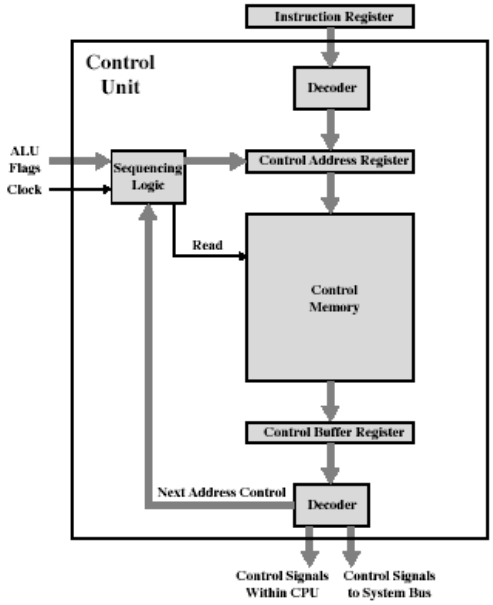


Figure 17.4 Functioning of Microprogrammed Control Unit

The instruction register gets instructions from t**he user program memory.**

The ALU flags are input to the sequencing logic for c**onditional branches**

To do a branch on this machine in microcode the s**equencer can generate next address based on a field in the present instruction.**

Since it is looking at the present microinstruction, the sequencer knows when the present instruction is done.

Stallings 10th Edition
The figure below is Wilkes' original microcode concept.

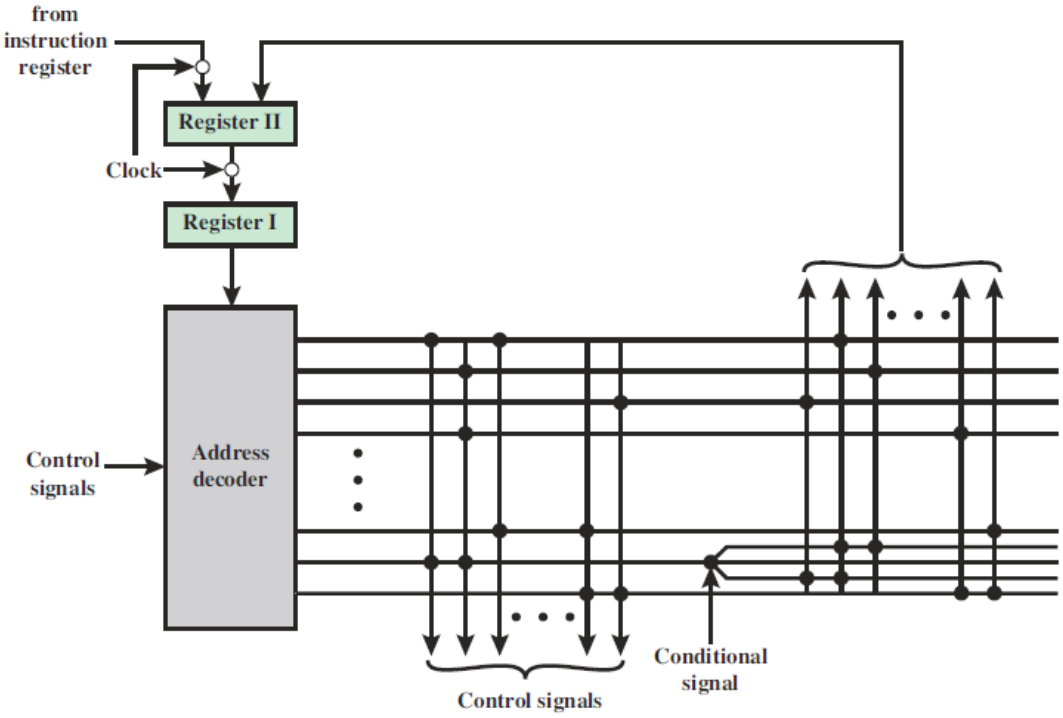
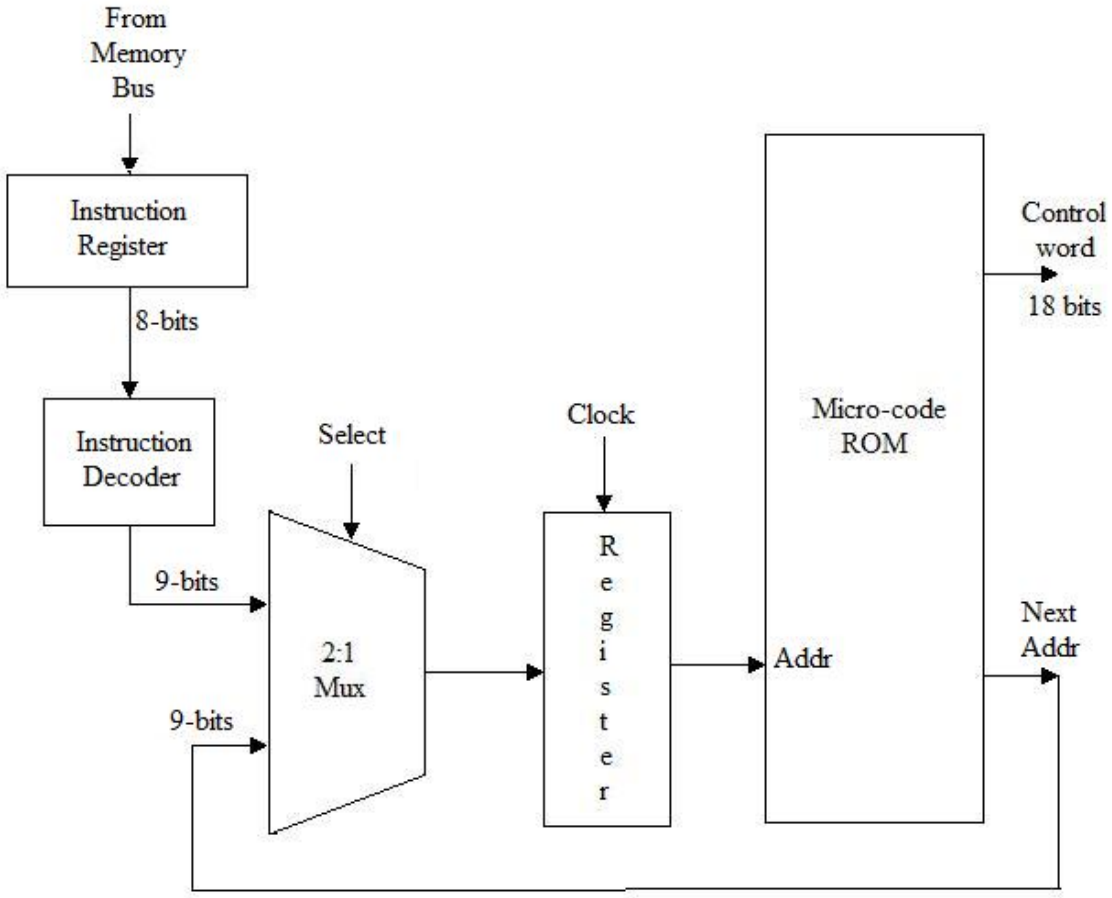


Figure 20.5 Wilkes's Microprogrammed Control Unit

Wilkes used a diode matrix. This is a **ROM using diodes as memory elements.**
Register II has two input sources: **One is for the opcode from the instruction register and the other is feedback from the present microinstruction.**
Note that the **Address of next microinstruction is fed back to the control store to help determine the address of the next microinstruction.**

Stallings 10th Edition

The figure below is similar to that of the Wilkes' machine but in block diagram form.



• The control word is 18 bits long.
• The Control ROM has a 9-bit address so it is 512 x 27
• The Clock is the system clock and determines the speed at which microinstructions are executed.
• The instruction decoder translates an 8-bit opcode into the starting address in ROM for the microinstruction sequence.
• The Select line on the mux is set by one of the 18 lines in the control word.

The advantages of microprogramming are:
Flexibility and simplicity.
This is a general solution to the controller problem.

The disadvantages of microprogramming are:
Speed.
The general solution is slower than hardwired and will take up more space.

The address of the next microinstruction comes from one of three sources: **IR Op code, next sequential microinstruction, or a branch address.**

Microprogramming is usually slower than a hard wired controller because we h**ave to pass through ROM decoder and fetch sequences of microinstructions for**

operations instead of having hardware directly control the sequencing of operations.

Optimization of branching microinstructions is very important. **One out of every 3 or 4 microinstructions is a branch.**

One of the approaches for generating the next microinstruction address is to use a two address field as shown below.

Each microinstruction contains a control word and two addresses. The branching logic then decides which of these two addresses becomes the next address.

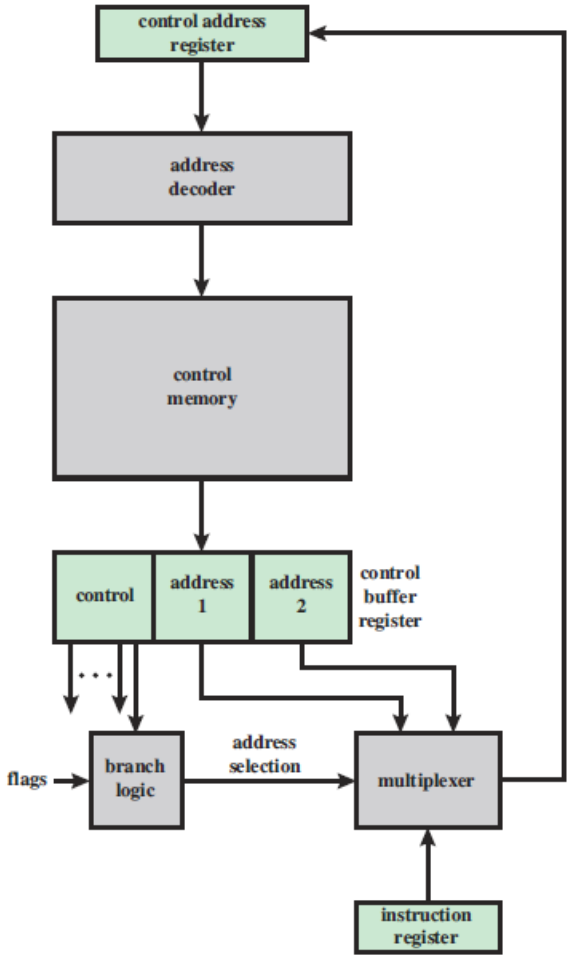


Figure 20.6 Branch Control Logic: Two Address Fields

Stallings 10th Edition

Another approach is to use a single address field in the microinstruction. **Logic allows the next address to be either the next sequential address, the address in the address field, or an address generated from the op code.**

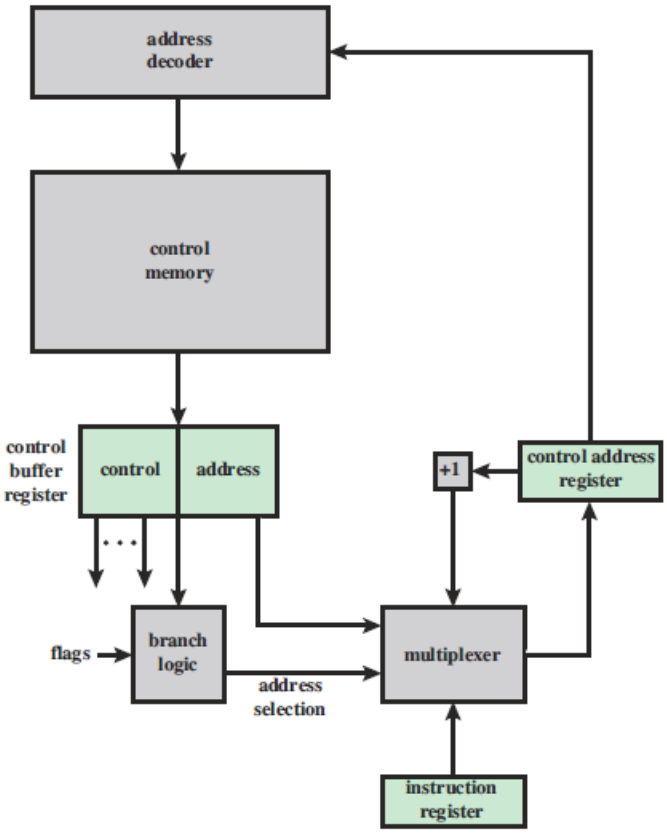


Figure 20.7 Branch Control Logic: Single Address Field

Stallings 10th Edition

20. A third approach is to use a variable instruction format. **One bit of the control word determines whether the remainder of the microinstruction has an address or is an extension of the control word.**

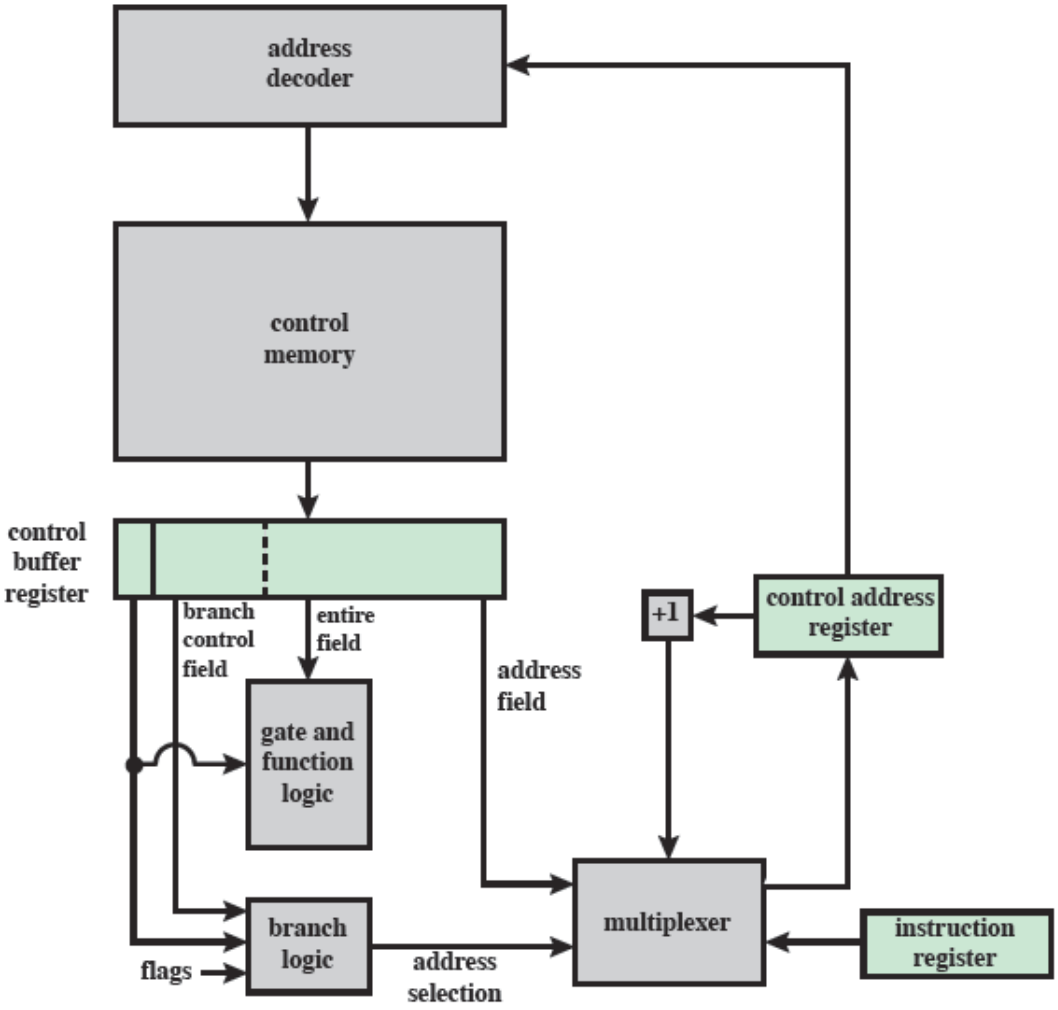


Figure 21.8 Branch Control Logic: Variable Format

Microinstruction can do conditional branches. The **ALU flags, part of op code or address mode, part of selected registers, or status bits within the control unit not seen by the programmer are used for this purpose.**

The author refers to a technique called "mapping" in which an address gets translated to another address to access the control store. This is necessary because the o**p code is largely independent of machine structure so the op code gets mapped to an address of the control store where that instruction's microinstruction sequence is stored.**

A horizontally microprogrammed machine is said to be unpacked while a vertically microprogrammed machine is said to be packed. **Vertical microprogramming implies a degree of encoding of the control bits. Hence, each bit in the**

Stallings 10th Edition

microinstruction contains more information while horizontal microprogrammed bits have each bit committed to a single function.

The rule of thumb width for horizontal and vertical microprogram instructions: **16 to 40 bits is vertical and 40 to 100 bits or more is horizontal.**

One thing that happens in vertical microprogramming is that in practice, more bits are used for encoding than are actually necessary. **When bits are encoded as say a 3:8 line decoder then the decoder makes only one of the 8 options available at any one time. This isn't always practical.**

The following table shows characteristics for vertical and horizontal microprograms:

Characteristic	Horizontal	Vertical
encoding	**unencoded**	**highly encoded**
number of bits	**lots of bits**	**relatively few bits**
control of hardware	**very detailed**	**gross view**
programmability	**difficult**	**easy**
concurrency	**fully exploited**	**not fully exploited**
amount of control logic	**little control logic**	**complex control logic**
speed	**fast execution**	**slow**
performance	**optimized**	**not optimized**
programming	**not optimized**	**optimized**

The terms hard and soft are often used to describe horizontal and vertical microprogrammed machines. **Horizontal is hard and vertical is soft. These terms describe the degree of closeness to of the microcode to the hardware.**

The terms packed and unpacked are often used to describe horizontal and vertical microprogrammed machines. **Horizontal is unpacked and vertical is packed. These terms describe the degree of encoding of the microcode.**

Stallings 10th Edition

The Figure below describes "Direct encoding" and "Indirect encoding". **In direct encoding there is only one level of decoders between the microinstruction and the hardware. In indirect encoding there are multiple layers of encoding.**

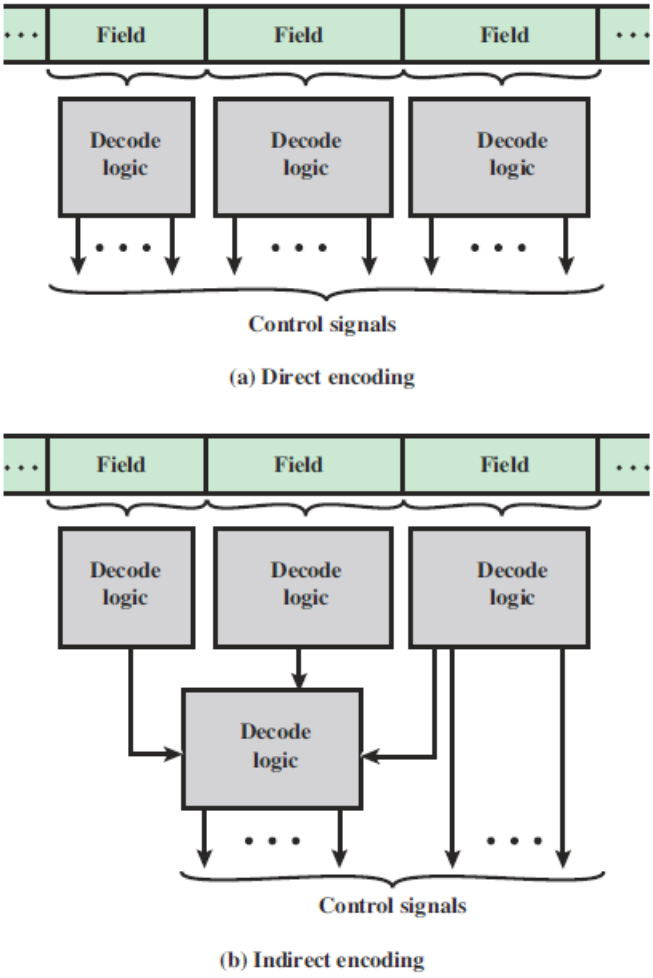


Figure 20.11 Microinstruction Encoding

There are two approaches that can be taken to organize the encoded microinstructions in a vertical machine: functional encoding and resource encoding.

Functional encoding groups signal fields by their function such as moving data between registers, doing logical operations, etc.

Resource encoding groups fields by what resources they control such as the ALU, register reads, etc.

The LSI-11 is/was **a microprocessor implementation of the PDP-11 that is microprogrammed. The LSI-11 (PDP-11/03), introduced in February 1975 is the first PDP-11 model produced using large-scale integration; the entire CPU is contained on four LSI chips made by Western Digital.**

Stallings 10th Edition

The figure below shows the LSI-11 and its control unit.

- The control store is **2048 words by 22 bits**
- The ALU is in **the Data Chip.**
- The purpose of the Control Address Register is to h**old the address of the next microinstruction.**
- The purpose of the Control data register **holds the microprogram control word.**
- Note that there is a feedback path from the Control data register to the microprogram sequence control. **The next microprogram address may be dependent on the present microinstruction. This loop allows conditional jumps.**
- The return register **allows the microprogrammer to save a single return address so that microinstruction sequences can have subprograms 1-deep.**
- The LSI-11 is s**trongly vertical.**

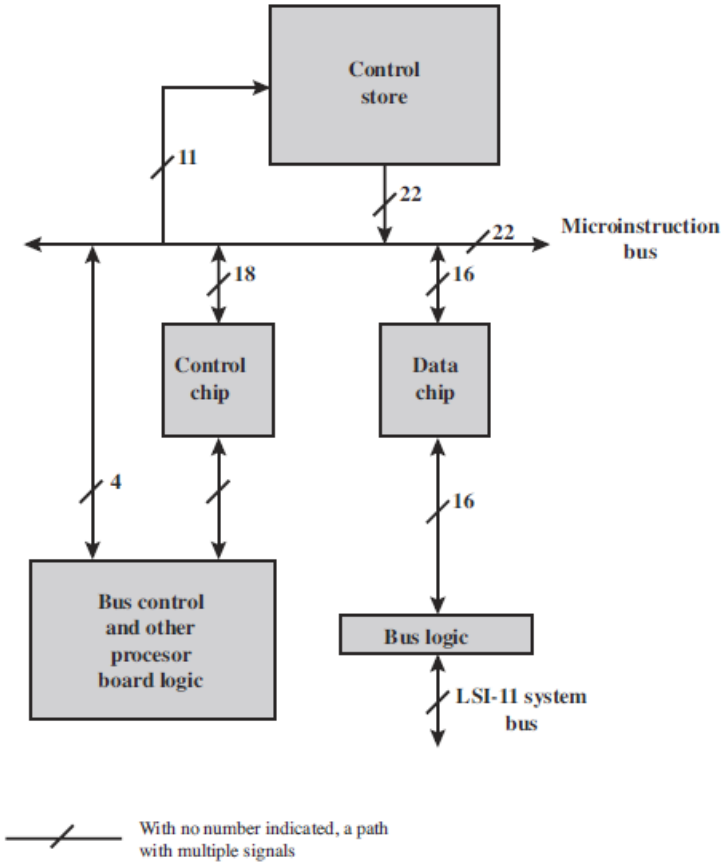


Figure 20.13 Simplified Block Diagram of the LSI-11 Processor

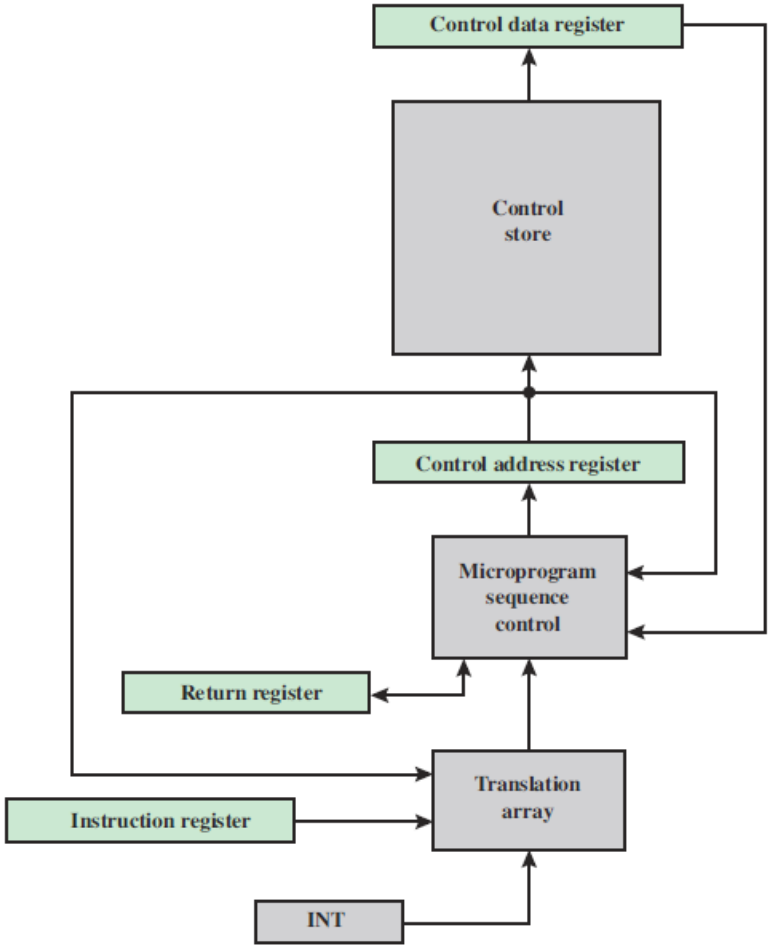
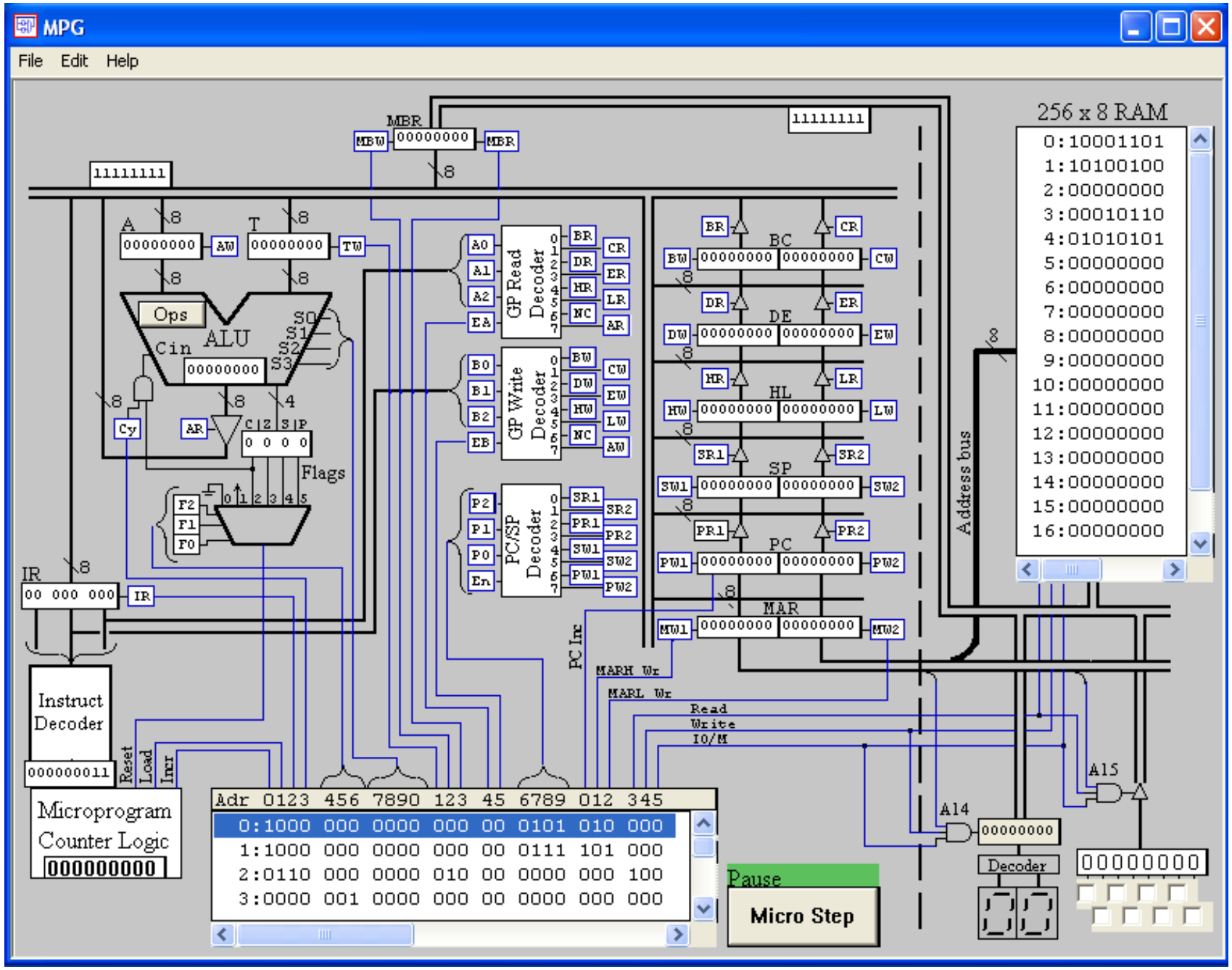


Figure 20.14 Organization of the LSI-11 Control Unit

Some machines have a writable control store. This allows you t**o create your own specialized instruction set.**

The downside of creating your own instruction set is that you must then create your own assembler, compiler, etc.

Microprogramming would be better than hard wired control at providing good high level language support?



An 8085 like machine that is microprogrammed.

Bits	Operation
0	MPC ← MPC + 1
1	MPC ← Instruction Decoder output
2	IR ← Bus
3	ALU Cin < Cy
4-6	MPC Reset = {0, 1, Cy, Z, S, P}
7-10	ALU Function Select
11	T ← Bus
12	Internal Bus ← External Bus
13	External Bus ← Internal Bus
14	Enable Read Decoder
15	Enable Write Decoder
16-18	Select Stack/PC high and low read and write
19	Enable Stack/PC read/Write
20	PC ← PC + 1
21	MARH ← Bus
22	MARL ← Bus
23	Memory Read enable
24	Memory Write enable
25	IO/M select

Microinstructions