Microprogrammed Control
Chapter 17

Team Members

History of Microprogrammed Control

- The term micro program was first coined by Wilkes in the early 1950’s.
- He proposed the approach to control unit design that was organized and systematic and avoided the complexities of hard wiring.
- In its 1964 February issue, Datamation Magazine reviewed microprogramming as “somewhat cloudy” because no micro programmed system was in use at this time. Amazing for such a revolution!

History/Overview (Cont.)

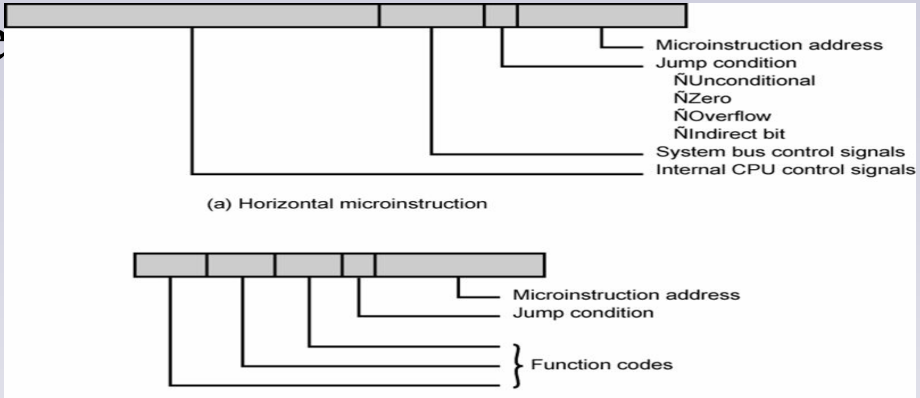
- In April 1964 IBM introduced its System/360 and all but the largest models where micro programmed.
- The 360 series predated the availability of semiconductor Rom.
- In recent years microprogramming has become less used but remaining a tool available for programmers.
- A great example is the Pentium 4 in which machine instruction is converted into a RISC(Reduced Instruction set computer) format.

Microinstructions

- In order to implement a control unit, the design must include sequencing through micro-operations, for executing micro-operations, for interpreting opcodes, and for making decisions based on ALU flags.
- Each microoperation is described in symbolic notation looking like a programming language. Referred to as microprogramming language.
- Each line describes a set of microinstructions occurring at one time.
- A sequence of instructions is known as a microprogram or firmware.

Microinstructions (Cont.)

- In any micro-operation, each control line coming from the control unit is either on or off. This can be represented as a binary digit.
- Then we construct a control word in which each bit represents one control line.
- Suppose we put our control words in memory, with each word having a unique address. Now add an address field to each control word, indicating the location of the next control word to be executed if a certain condition is true. Also add a few bits to specify the condition; the re
microinstruction.



(a) Horizontal microinstruction

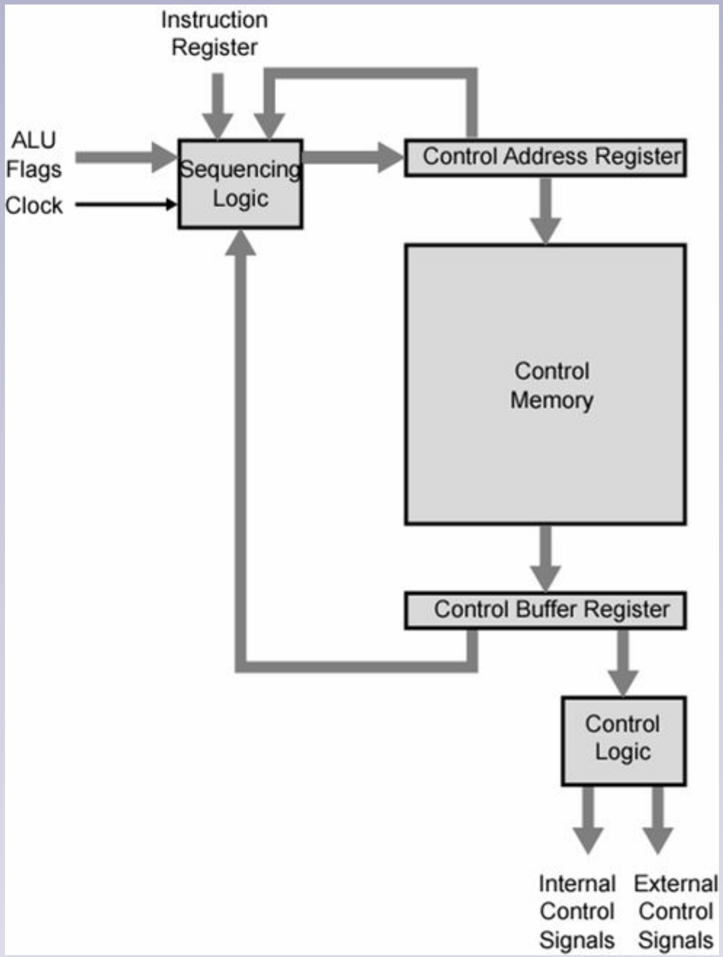
(b) Vertical microinstruction

Horizontal Microinstruction Execution

- To execute a microinstruction, turn on all the control lines indicated by a ‘0’bit. The result will cause one or more micro-operations to be performed.
- If the condition indicated by the condition bit is false, continue to the next microinstruction in sequence.
- If the condition is true, the next micro instruction to be executed is indicated in the address field.
- The control words are addressed in memory called control memory.

Microprogrammed Control Unit Function

- To execute an instruction, the sequencing logic unit issues a Read command to the control memory.
- The word whose address is specified in the control address register is read in the control buffer.
- The content of the control buffer generates control signals and next address information for the sequencing logic unit.
- The sequencing logic unit loads a new address into the control address register based on the next address info from the control buffer register and the ALU flags.
- Important (all this happens during one clock pulse).



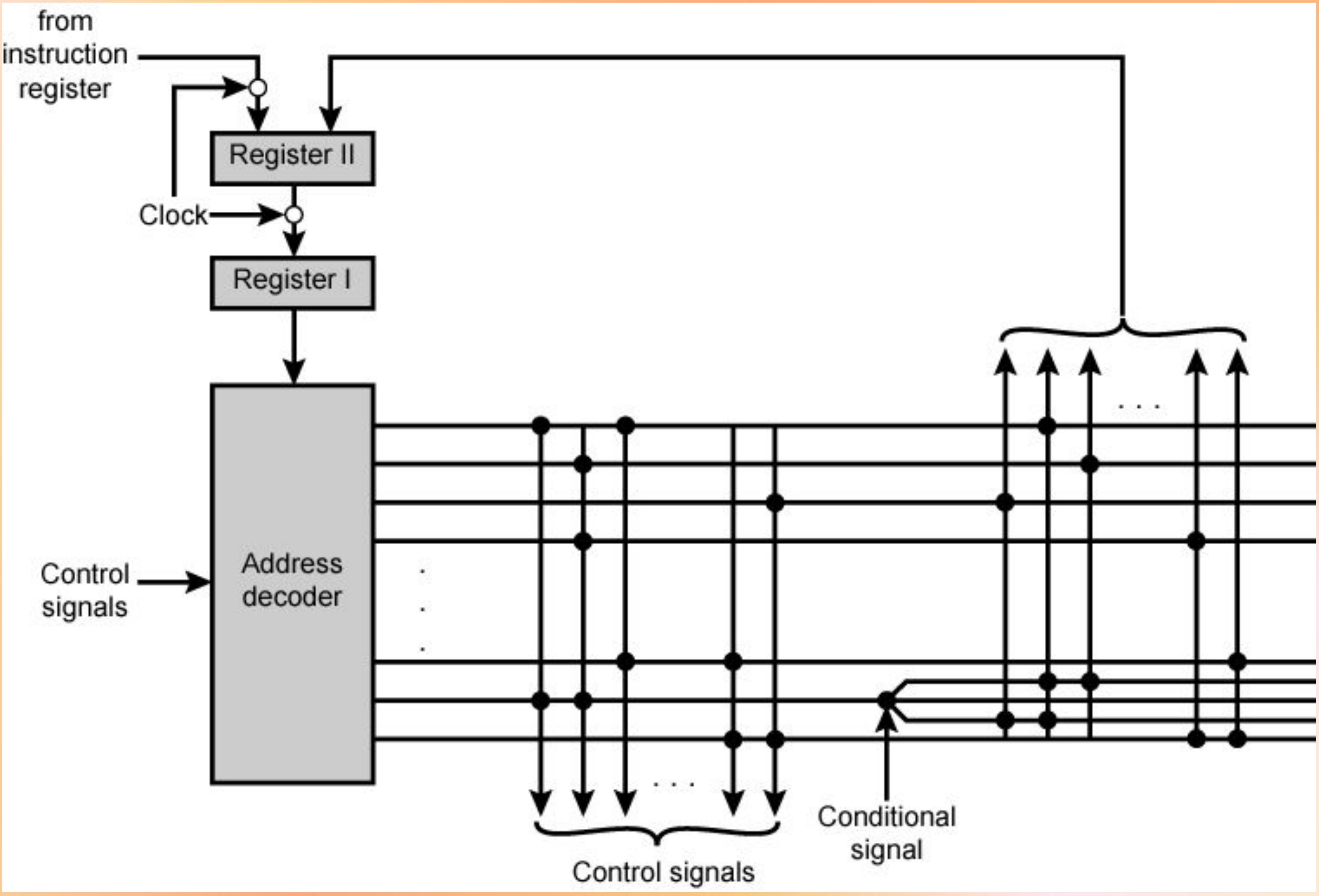
Key Terms/ Elaboration

- **Control memory:** stores the set of microinstructions.
- **Control Address register:** contains the address of the next instruction to be read.
- **Control Buffer register:** Stores a microinstruction when it is read.
- **Get the next instruction:** Adds 1 to the control address register.
- **Jump to new routine based on jump microinstruction:** Load the address field of the control buffer.
- **Jump to machine instruction routine:** Load the control address register based on the opcode in the IR.

Wilkes Control

- 1951
- Matrix partially filled with diodes
- During cycle, one row activated
  - Generates signals where diode present
  - First part of row generates control
  - Second generates address for next cycle

Wilkes's Microprogrammed Control Unit



Microinstructions for Wilkes Example

		Arithmetical Unit	Control Register Unit	Conditional Flip-Flop Set	Conditional Flip-Flop Use	Next Micro-instruction 0	Next Micro-instruction 1
	0		*F* to *G* and *E*			1	
	1		(*G* to '1') to *F*			2	
	2		Store to *G*			3	
	3		*G* to *E*			4	
	4		*E* to decoder			—	
A	5	*C* to *D*				16	
S	6	*C* to *D*				17	
H	7	Store to *B*				0	
V	8	Store to *A*				27	
T	9	*C* to Store				25	
U	10	*C* to Store				0	
R	11	*B* to *D*	*E* to *G*			19	
L	12	*C* to *D*	*E* to *G*			22	
G	13		*E* to *G*	$(1)C_5$		18	
I	14	Input to Store				0	
O	15	Store to Output				0	
	16	(*D* + Store) to *C*				0	
	17	(*D* − Store) to *C*				0	
	18				1	0	1
	19	*D* to *B* (*R*)*	(*G* − '1') to *E*			20	
	20	*C* to *D*		$(1)E_5$		21	
	21	*D* to *C* (*R*)			1	11	0
	22	*D* to *C* (*L*)†	(*G* − '1') to *E*			23	
	23	*B* to *D*		$(1)E_5$		24	
	24	*D* to *B* (*L*)			1	12	0
	25	'0' to *B*				26	
	26	*B* to *C*				0	
	27	'0' to *C*	'18' to *E*			28	
	28	*B* to *D*	*E* to *G*	$(1)B_1$		29	
	29	*D* to *B* (*R*)	(*G* − '1') to *E*			30	

-Notation A,B,C stands for the various registers in the arithmetical and control register units.

-C to D indicates that the switching circuits connect the output of register C to the input register D.

-(D+A) to C indicates that the output register of A is connected to the one input of the adding unit and the output of the adder to register C.

Advantages and Disadvantages

Advantages of using a microprogrammed control unit over a hardwired control unit:

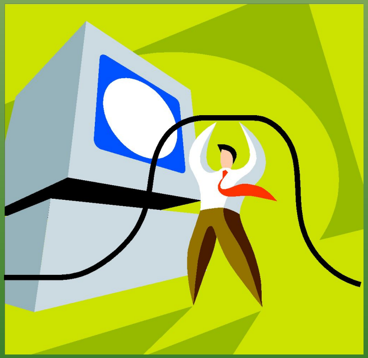
- Simplifies the design of the control unit.
- Makes it both cheaper and less error prone to implement.
- A hardwired control unit must contain complex logic for sequencing through the many micro-operations of the instruction cycle. In contrast, the decoders and sequencing logic unit of a microprogrammed control unit are very simple pieces of logic.

Disadvantage:

- A microprogrammed unit is slower than a hardwired unit.

Tasks Performed

- Microinstruction sequencing
- Microinstruction execution
- Must consider both together



Design Considerations

- Size of microinstructions
- Address generation time
  - Determined by instruction register
    - Once per cycle, after instruction is fetched
  - Next sequential address
    - Common in most designed
  - Branches
    - Both conditional and unconditional

Sequencing Techniques

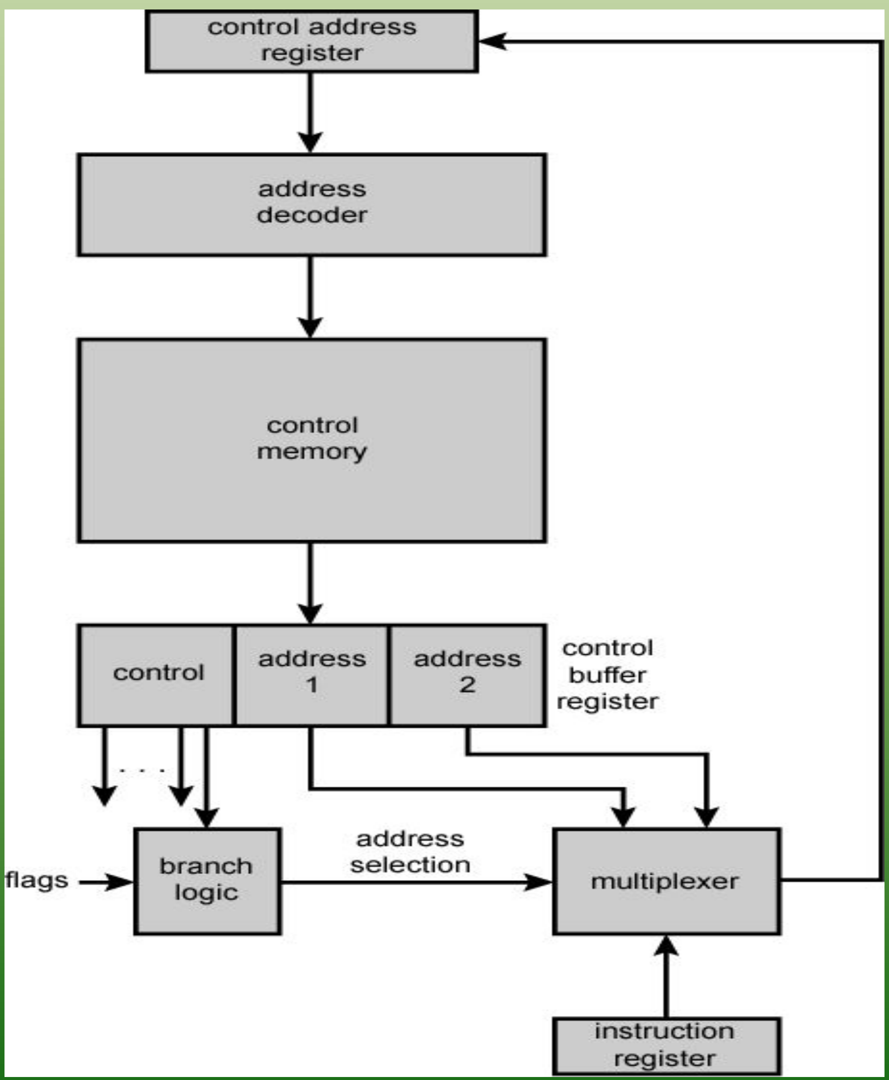
- Based on current microinstruction, condition flags, contents of IR, control memory address must be generated
- Based on format of address information

  -Two address fields

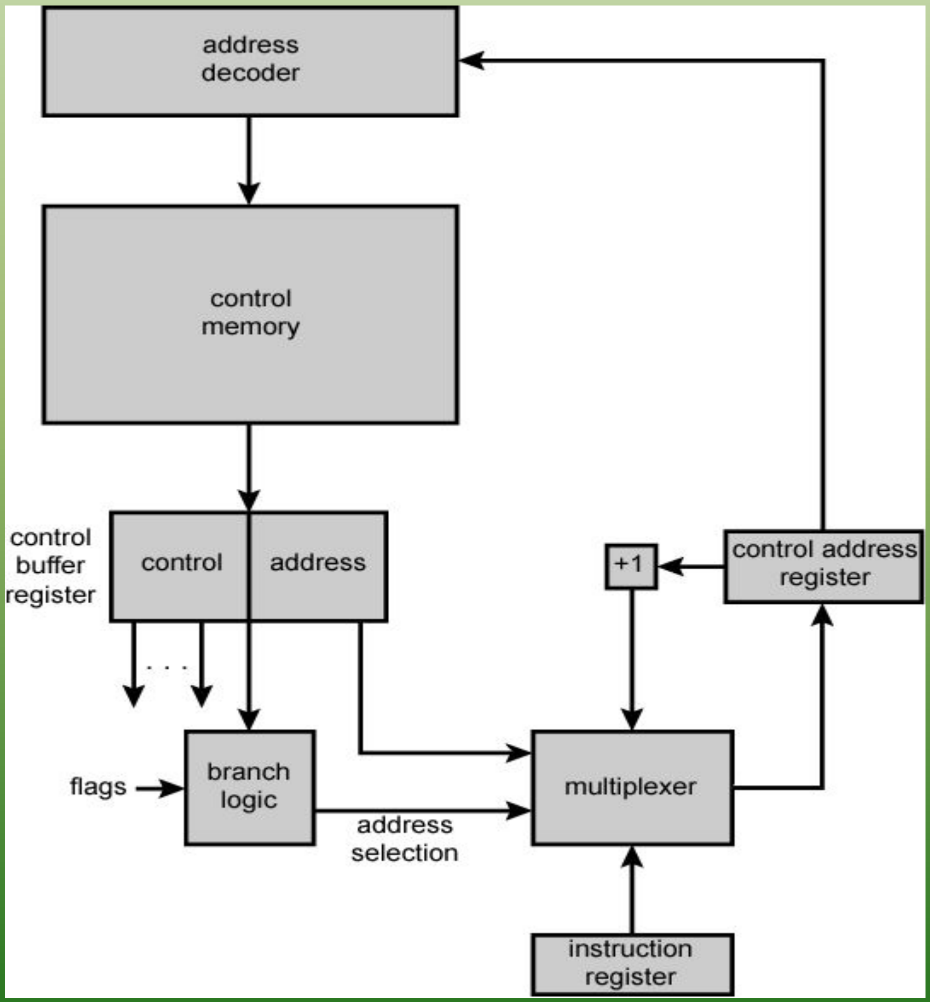
  -Single address field

  -Variable format

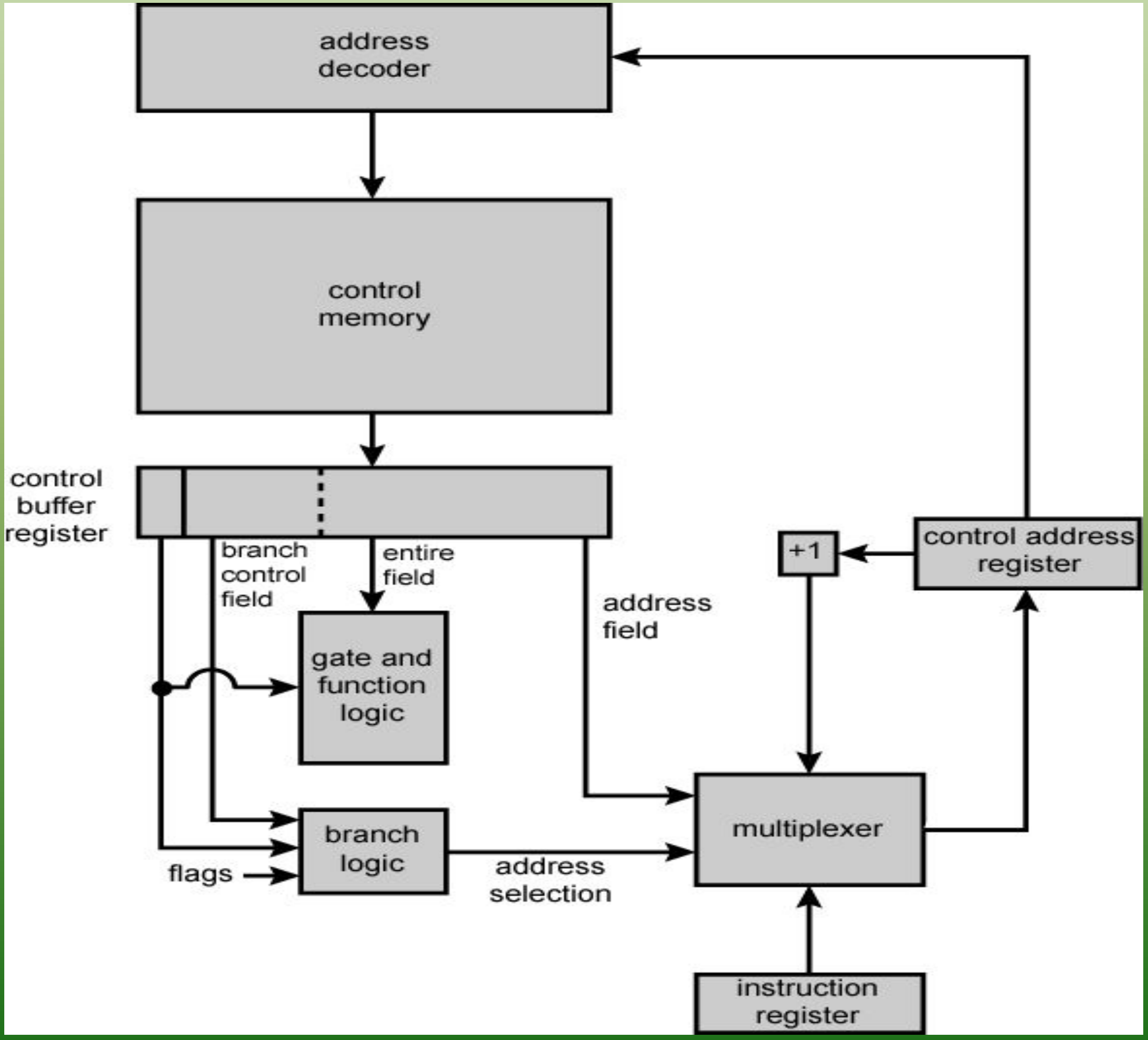
Branch Control Logic: 2 Address Fields



Branch Control Logic: Single Address Field



Branch Control Logic: Variable Format

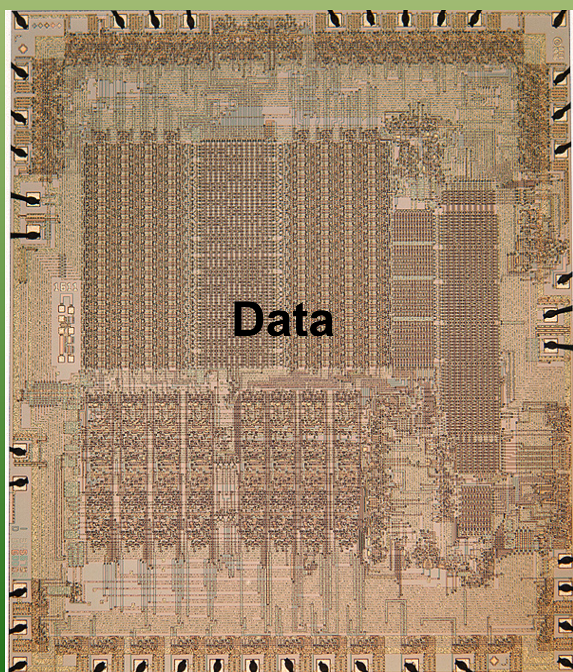


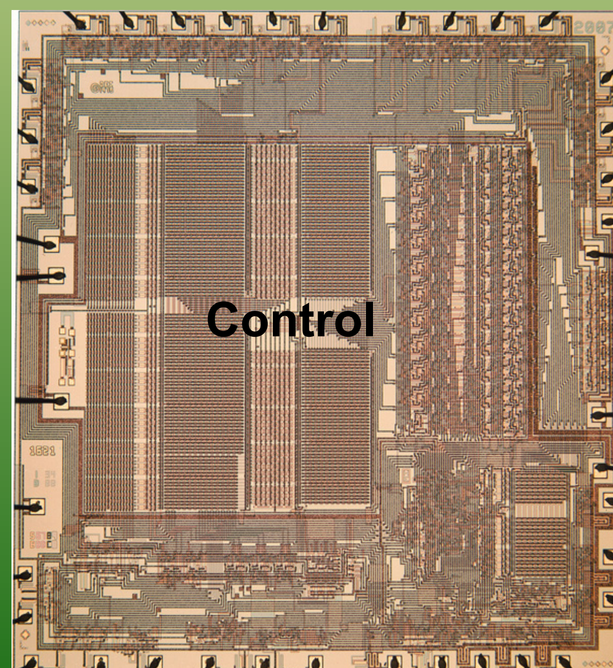
Address Generation

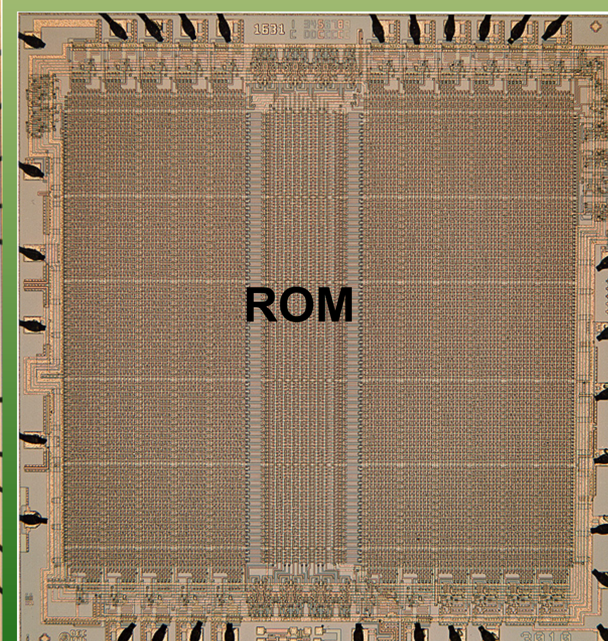
Explicit	Implicit
Two-field	Mapping
Unconditional Branch	Addition
Conditional branch	Residual control

LSI-11 Microinstruction Sequencing

- LSI means Large Scale Integration (thousands of transistors)
- QBUS – multiplexed the address & data on the same bus pathways







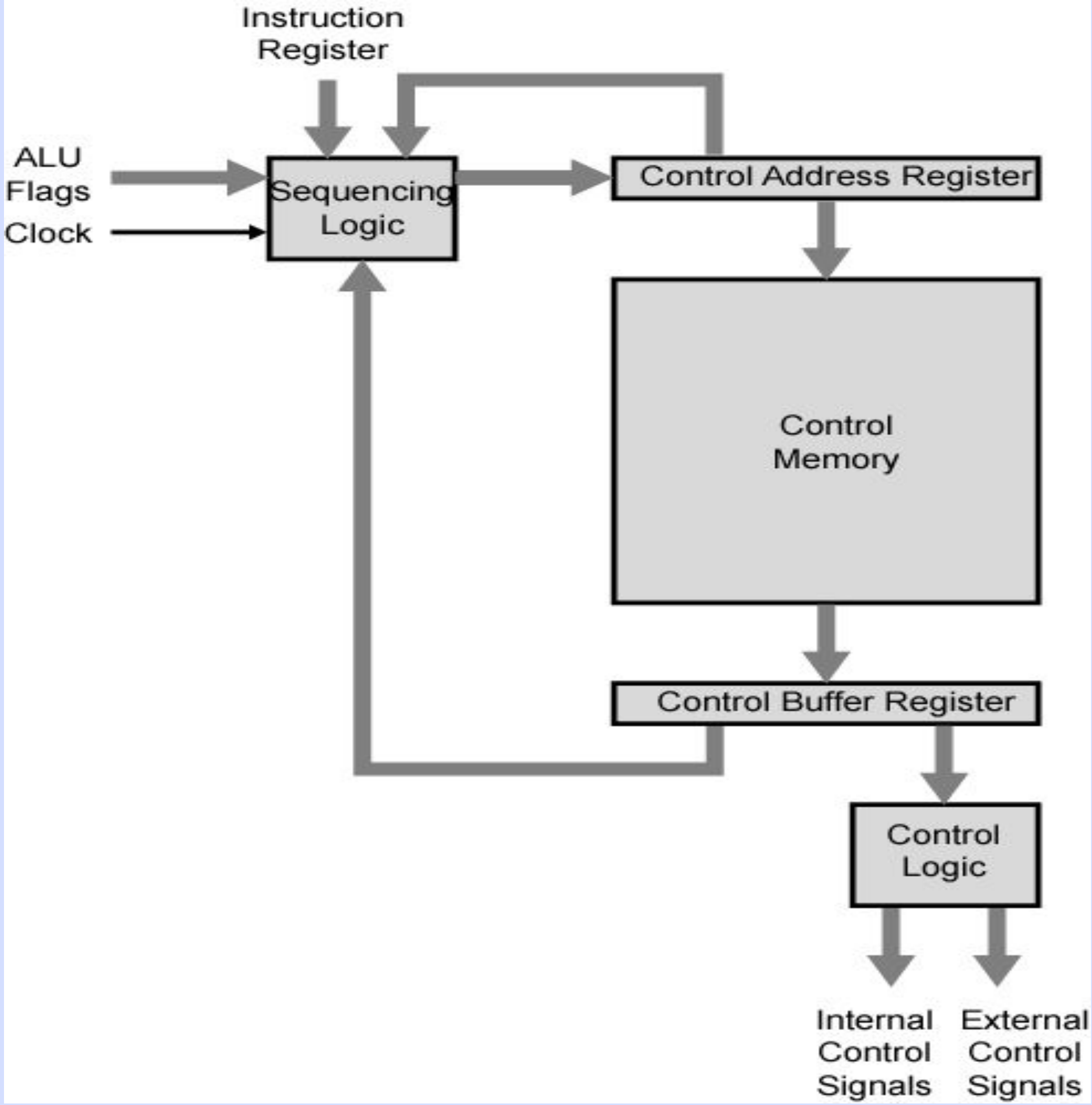
5 Aspects of Microinstruction Addressing

- *Next Sequential Address-* Incrementing the CAR
- *Opcode Mapping-* The microinstruction address is determined by the opcode
- *Subroutine Facility-* Single, Double Address Field, or Variable Format
- *Interrupt Testing-* If an interrupt has occurred, a the interrupt procedure determines the next microinstruction
- *Branch-* conditional and unconditional branch instructions are used

The Microinstruction Cycle

- The basic event on a microprogrammed processor
- Each cycle is made up of two parts:
  - Fetch
  - Execute

Organization



Microprogramming Classifications

- Horizontal/vertical

  -Refer to the relative width of microinstructions

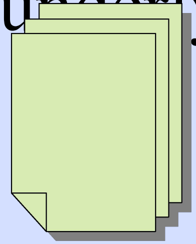
- Packed/unpacked

  -Relate to the degree of identification between a given control task and specific microinstruction bits.

- Hard/soft

  -Suggest the degree of closeness to the underlying control signals and hardware layout

- Direct/indirect encoding

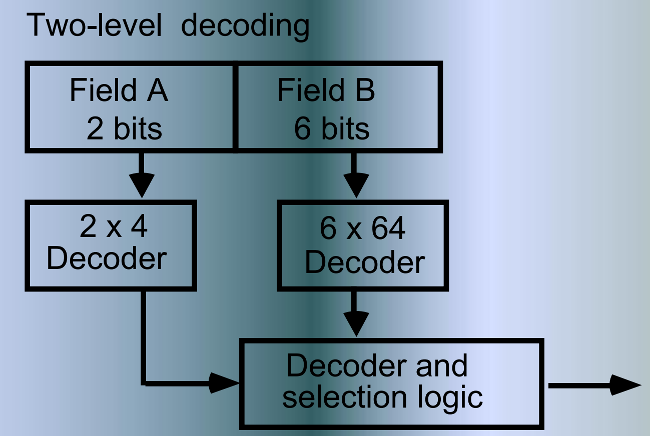


Horizontal Microprogramming

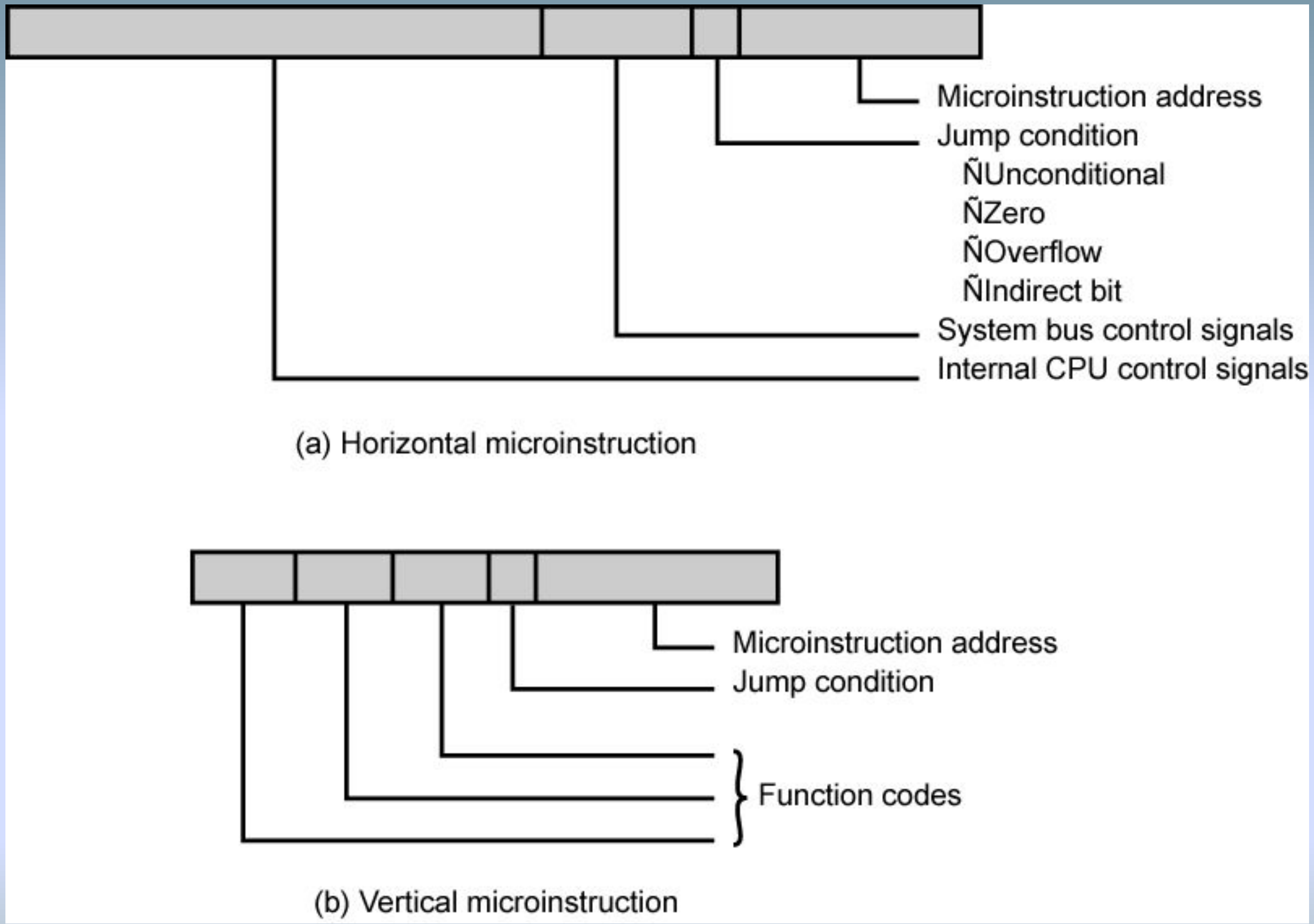
- Each microinstruction specifies many different micro-operations to be performed in parallel
- Wide control memory word(more bits)
- Optimizing performance as result of fast execution
- Efficient hardware utilization
- Little encoding of control information
- More difficult to program

Vertical Microprogramming

- Each microinstruction specifies a single micro-operation to be performed
- Control word width is narrow
- Limited ability to express parallelism
- Control signals encoded into function codes – needs to be decoded using decoder circuits(more hardware)
- Slower execution
- Optimize programming



Typical Microinstruction Formats



Packed/Unpacked Microprogramming

- Packing connotes encoding
- As the bits become more packed, a given number of bits contain more information

Hard/Soft Microprogramming

- Suggest the degree of closeness to the underlying control signals and hardware layout
- Hard microprograms are generally more fixed and committed to read-only memory
- Soft microprograms are more flexible and are suggestive of user microprogramming

Advantages Over Wilkes

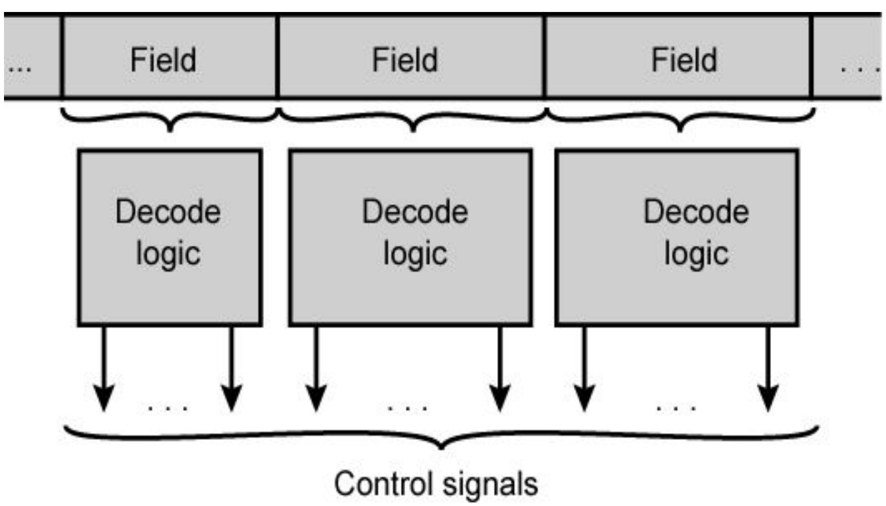
- Wilkes had each bit directly produce a control signal or one bit of the next address
- More complex address sequencing schemes, using fewer microinstruction bits, are possible
- Control word bits can be saved by encoding and subsequently decoding control information
- Require more complex sequencing logic module

Encoding Techniques

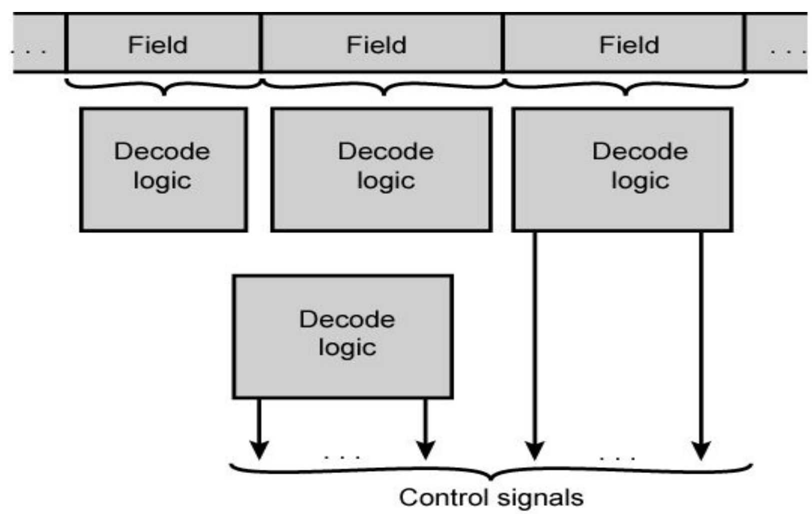
- Microinstruction are organized as set of fields
- When microinstruction is executed, every field is decoded and generated control signals.
- Activates one or more control signals
- Organize format into independent fields
  - Field depicts set of actions (pattern of control signals)
  - Actions from different fields can occur simultaneously
- Define each field such that the alternative actions that can be specified by the field are mutually exclusive.
- Organizing the encoded microinstruction into field; function and resource.
- Only one of the actions specified for a given field could occur at a time.
- Another aspect of encoding is direct and indirect.

Microinstruction Encoding

This are two different way of encoding the direct and indirect. This are two basic technique of encoding illustration . The microinstruction is organized as a set of fields. Each field contains a code, which upon deciding, activates one or more control signals.



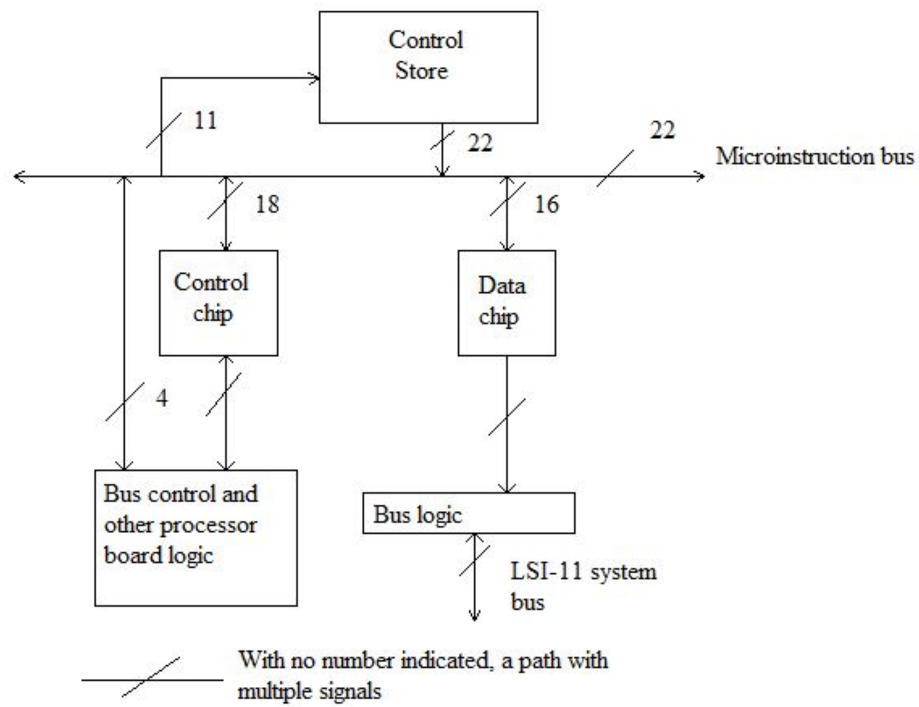
(a) Direct encoding



(b) Indirect encoding

LSI-11 Control Unit Organization

The LSI-11 is the first member of thePDP-11 family that was offered as a single-board processor. The board contains three LSI chips, an internal bus known as the microinstruction bus.



LSI-11 Microinstruction format

The LSI-11 uses an extremely vertical microinstruction format, witch is only 22 bits wide. The microinstruction set strongly resembles the PDP-11 machine instruction set that is implemented. This designs was intended to optimize the performance of a control unit within the constraint of a vertical, easily programmed design.

Review Questions

1. What is a horizontal microinstruction?
2. Define the roles of the control buffer register and control address register?
3. What is the principle disadvantage of a microprogrammed unit?
4. In Wilkes Diagram, what does the first part of the row generate?
5. What are the two tasks that must be considered when desi[illegible]g a control unit?

6. What are the three types of Branch control Logic?
7. What tasks does the sequencing logic unit perform?
8. Which classification of microprogramming optimizes the performance of your system? Give two reasons why.
9. What parts is the microinstruction cycle made up of?
10. In the original proposal by Wilkes, each bit of a microinstruction produced what?

Answers

1. Suppose we put our control words in memory, with each word having a unique address. Now add an address field to each control word, indicating the location of the next control word to be executed if a certain condition is true. Also add a few bits to specify the condition and the result is horizontal microinstruction.
2. The CAR holds the address of the next instruction to be read.

   The CBR stores the data last read from memory.
3. It will be slower than a hardwired unit of comparable technology.
4. It generates the control signals that control the operation of the processor.
5. Microinstruction sequencing, and microinstruction execution.
6. Single address field, double address field, and variable format.
7. It is in charge of loading the C.A.R. and issuing read commands.
8. Horizontal, because of the high degree of parallelism, less hardware, less encoding…
9. It’s made up of two parts, fetch and execute.
10. Each bit directly produced a control signal or directly produced one bit of next address