CAO: Lecture 25
Micro Instruction Sequencing

Topics Covered

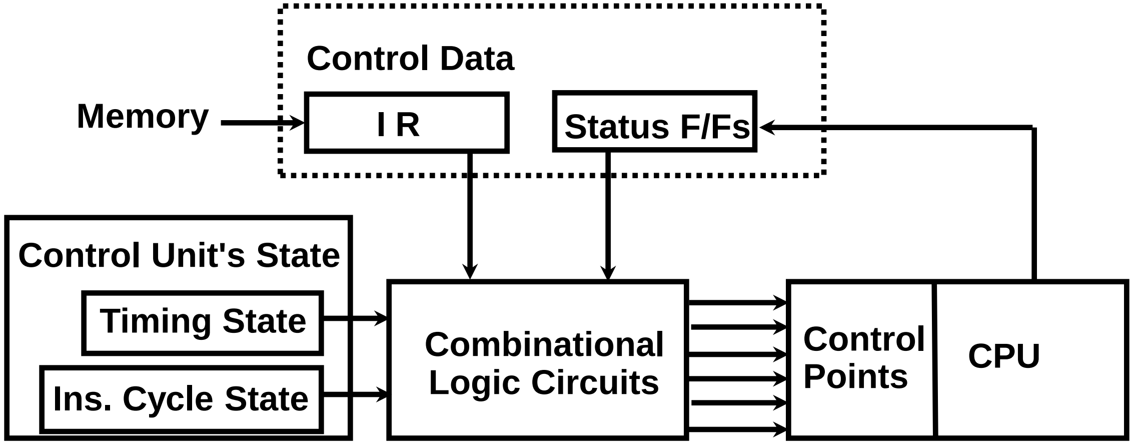
- Microprogrammed control
- Comparison of control unit implementations
- Microinstruction sequencing
- Conditional branch
- Mapping of instructions
- Microprogram example

MICROPROGRAMMED CONTROL

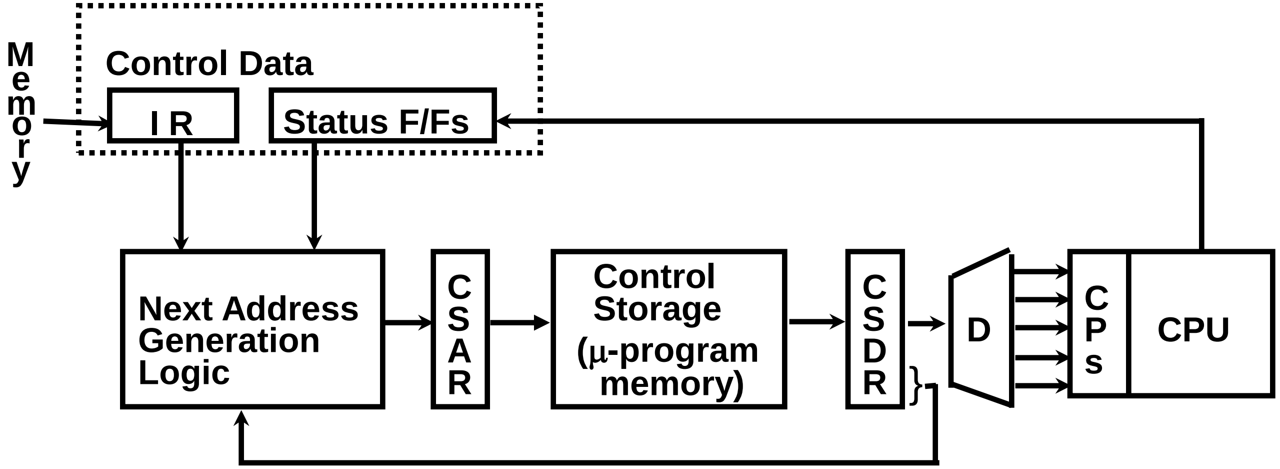
- Control Memory
- Sequencing Microinstructions
- Microprogram Example
- Design of Control Unit
- Microinstruction Format
- Nanostorage and Nanoprogram

COMPARISON OF CONTROL UNIT IMPLEMENTATIONS

Combinational Logic Circuits (Hard-wired)



Microprogram



TERMINOLOGY

Microprogram

- Program stored in memory that generates all the control signals required to execute the instruction set correctly
- Consists of microinstructions

Microinstruction

- Contains a control word and a sequencing word
  - Control Word - All the control information required for one clock cycle
  - Sequencing Word - Information needed to decide the next microinstruction address
- Vocabulary to write a microprogram

Control Memory(Control Storage: CS)

- Storage in the microprogrammed control unit to store the microprogram

Writeable Control Memory(Writeable Control Storage:WCS)

- CS whose contents can be modified
  - -> Allows the microprogram can be changed
  - -> Instruction set can be changed or modified

Dynamic Microprogramming

- Computer system whose control unit is implemented with a microprogram in WCS
- Microprogram can be changed by a systems programmer or a user

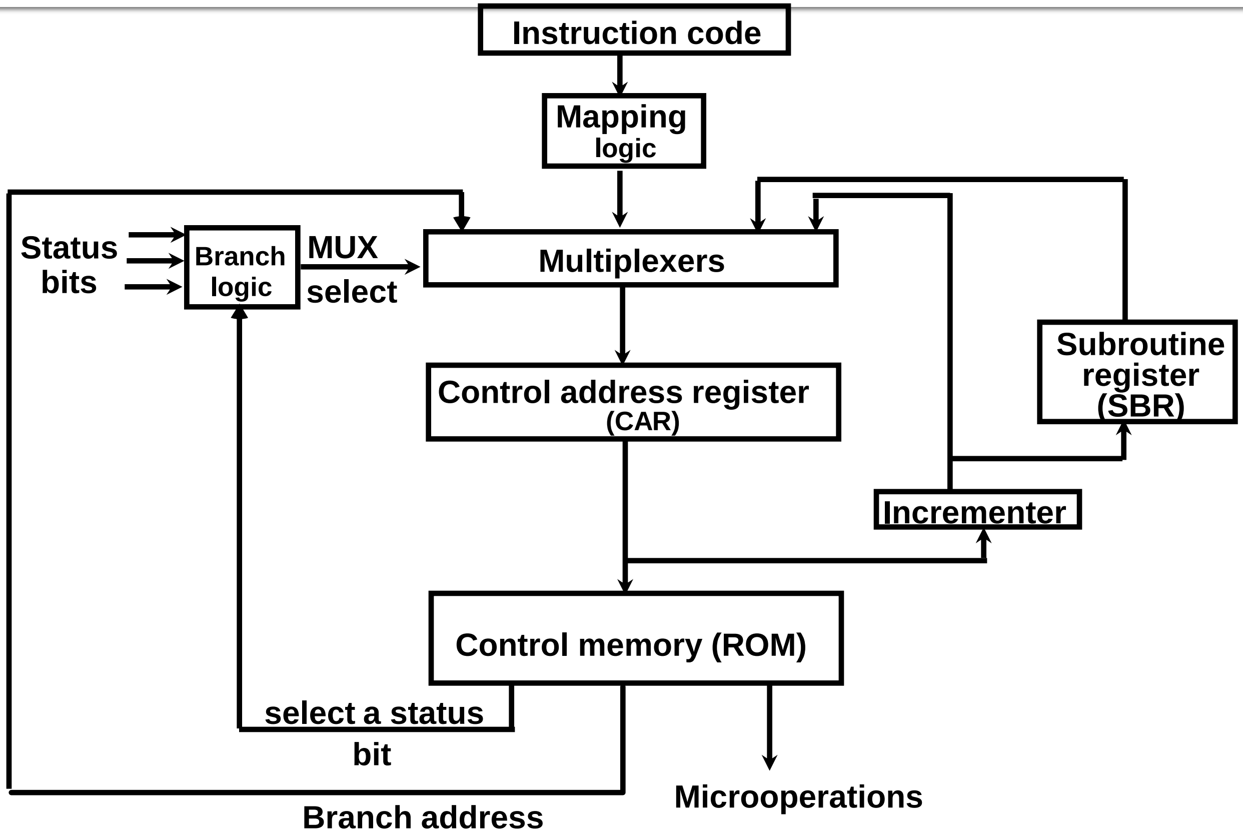
TERMINOLOGY

Sequencer (Microprogram Sequencer)

A Microprogram Control Unit that determines the Microinstruction Address to be executed in the next clock cycle

- **In-line Sequencing**
- **Branch**
- **Conditional Branch**
- **Subroutine**
- **Loop**
- **Instruction OP-code mapping**

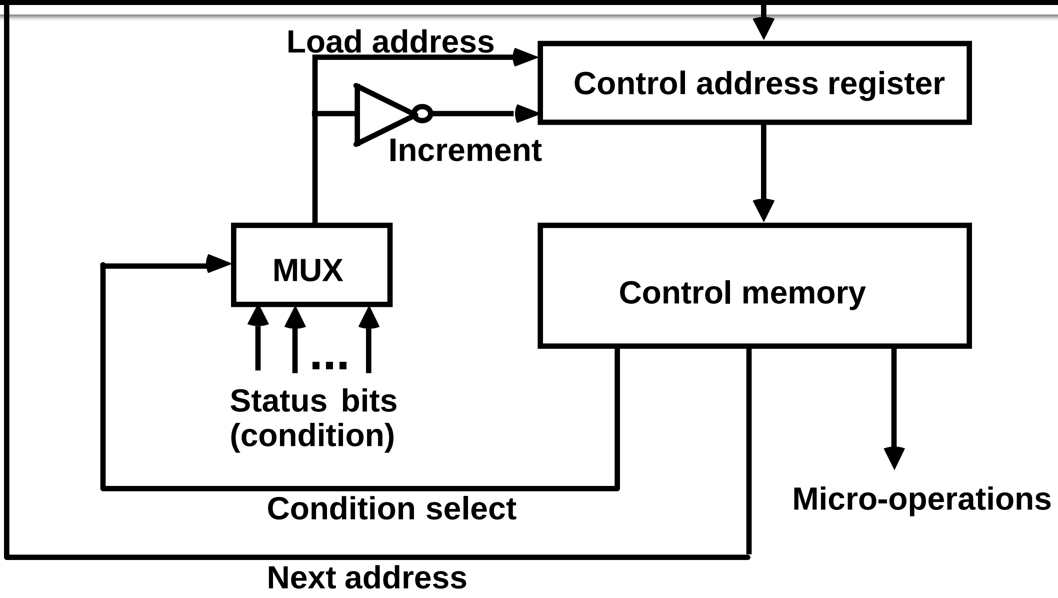
MICROINSTRUCTION SEQUENCING



Sequencing Capabilities Required in a Control Storage

- Incrementing of the control address register
- Unconditional and conditional branches
- A mapping process from the bits of the machine instruction to an address for control memory
- A facility for subroutine call and return

CONDITIONAL BRANCH



Conditional Branch

If *Condition* is true, then *Branch* (address from
the next address field of the current microinstruction)
else *Fall Through*

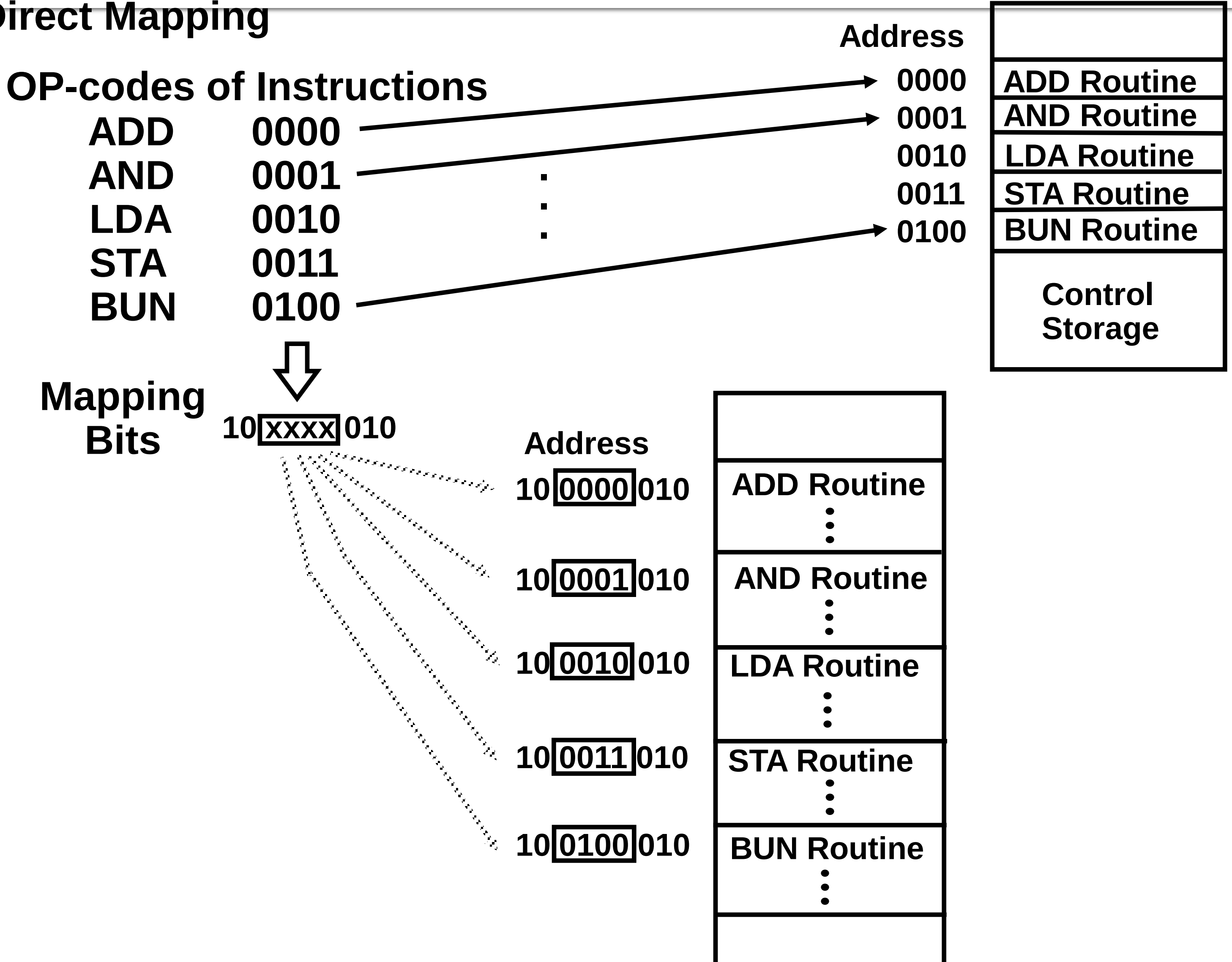
Conditions to Test: O(overflow), N(negative),
Z(zero), C(carry), etc.

Unconditional Branch

Fixing the value of one status bit at the input of the multiplexer to 1

MAPPING OF INSTRUCTIONS

Direct Mapping



OP-codes of Instructions

ADD	0000
AND	0001
LDA	0010
STA	0011
BUN	0100

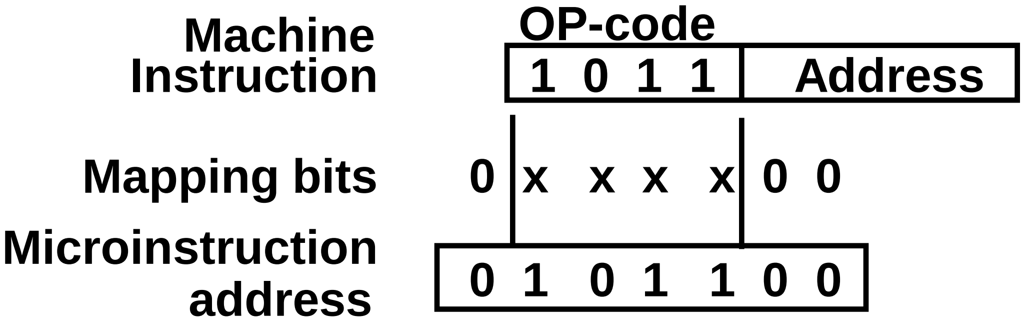
Address	
0000	ADD Routine
0001	AND Routine
0010	LDA Routine
0011	STA Routine
0100	BUN Routine
	Control Storage

Mapping Bits 10 xxxx 010

Address	
10 0000 010	ADD Routine
10 0001 010	AND Routine
10 0010 010	LDA Routine
10 0011 010	STA Routine
10 0100 010	BUN Routine

MAPPING OF INSTRUCTIONS TO MICROROUTINES

Mapping from the OP-code of an instruction to the address of the Microinstruction which is the starting microinstruction of its execution microprogram


Machine Instruction

OP-code

1 0 1 1 | Address

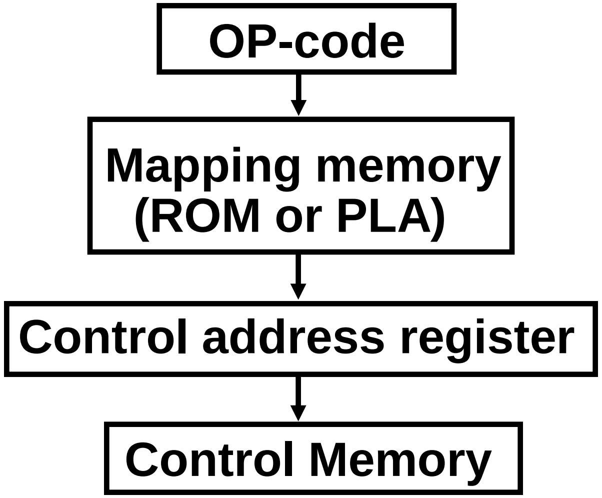
Mapping bits

0 x x x x 0 0

Microinstruction address

0 1 0 1 1 0 0

Mapping function implemented by ROM or PLA


OP-code

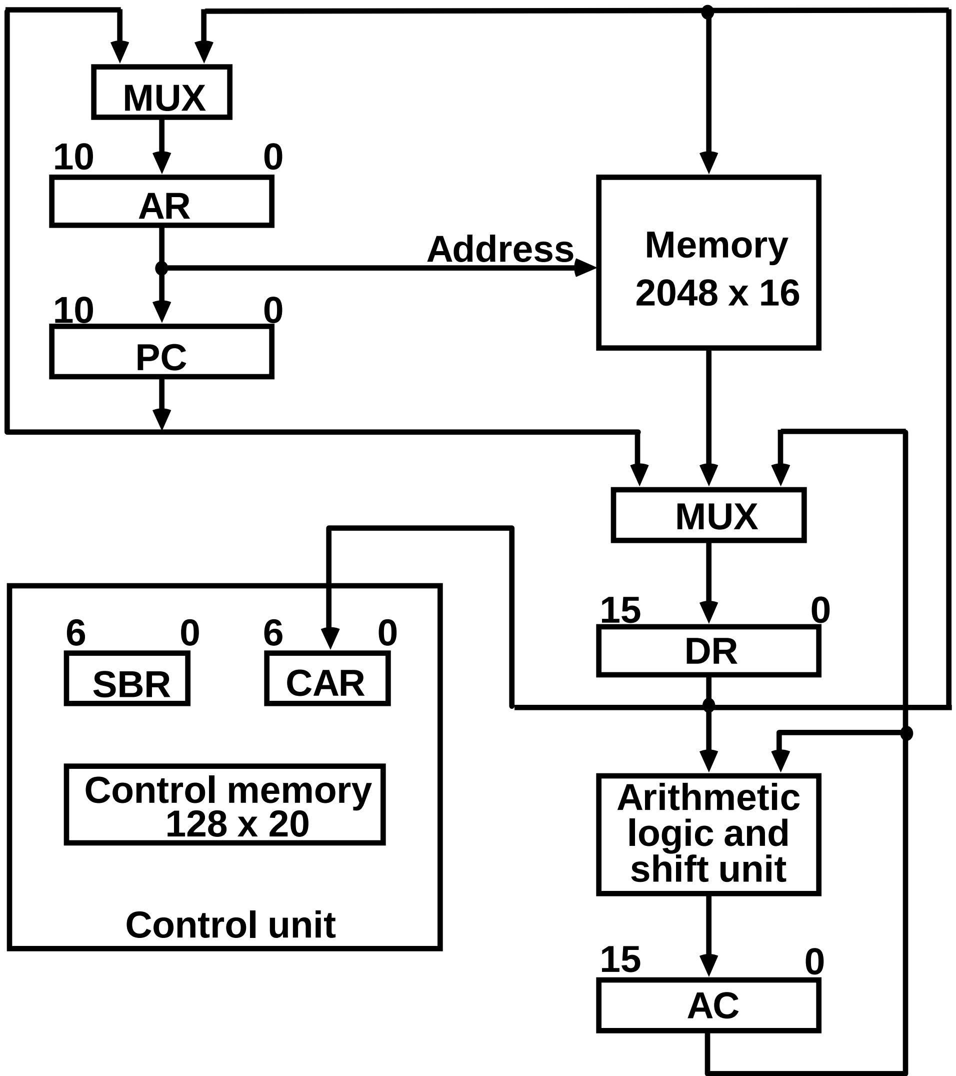
Mapping memory (ROM or PLA)

Control address register

Control Memory

MICROPROGRAM EXAMPLE

Computer Configuration


MUX
10
0
AR
Address
Memory
2048 x 16
10
0
PC
MUX
15
0
DR
6
0
SBR
6
0
CAR
Control memory
128 x 20
Control unit
Arithmetic
logic and
shift unit
15
0
AC