University Of Diyala
College Of Engineering
Computer Engineering Department

COMPUTER ARCHITECTURE I

PART 6: MICROPROGRAMMED CONTROL

Asst. Prof. Ahmed Salah Hameed

Second stage

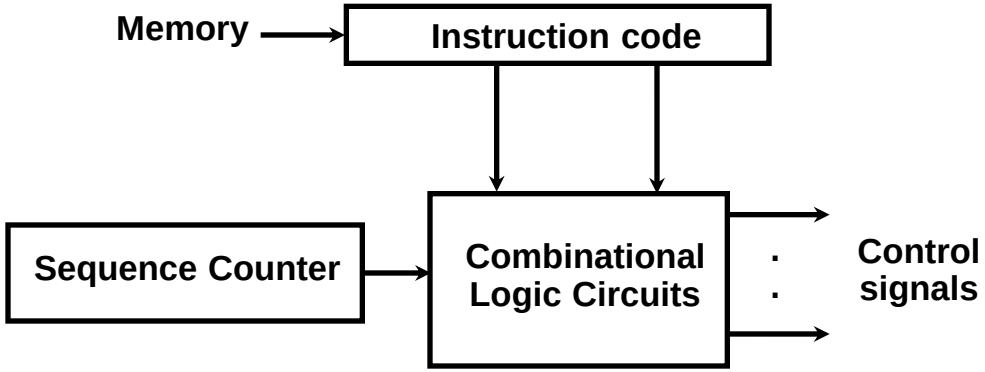
2022-2023

CONTENTS

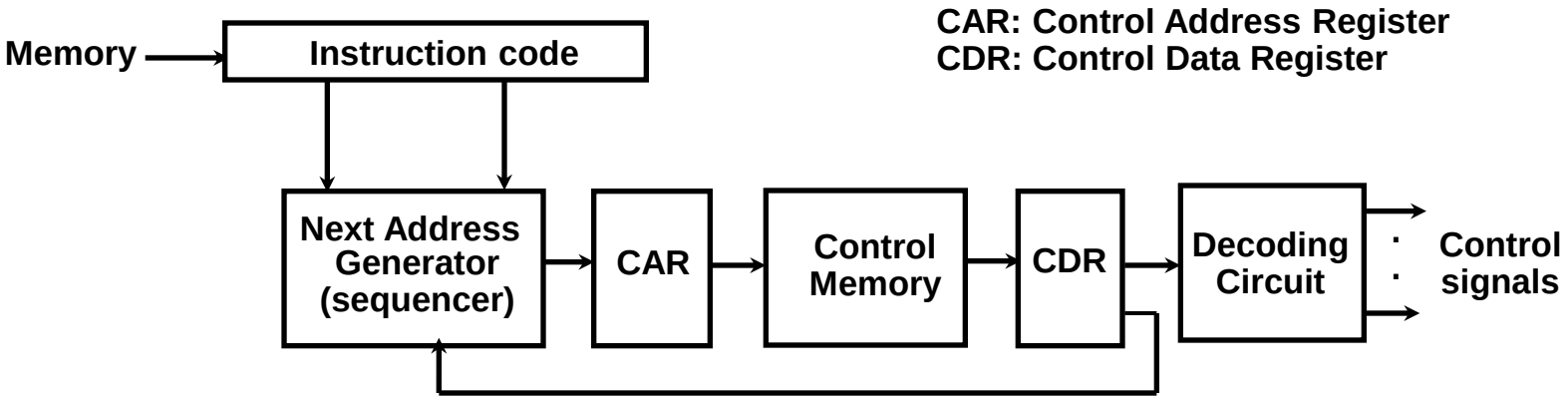
- Control Memory
- Address Sequencing
- Microprogram Example
- Design of Control Unit

CONTROL UNIT IMPLEMENTATION

- Hardwired



- Microprogrammed



MICROPROGRAMMED CONTROL UNIT

Control signals

- Group of bits used to select paths in multiplexers, decoders, arithmetic logic units

Control variables

- Binary variables specify microoperations
- Certain microoperations initiated while others idle

Control word

- String of 1’s and 0’s represent control variables

MICROPROGRAMMED CONTROL UNIT

Control memory

- Memory contains control words

Microinstructions

- Control words stored in control memory
- Specify control signals for execution of microoperations

Microprogram

- Sequence of microinstructions

CONTROL MEMORY

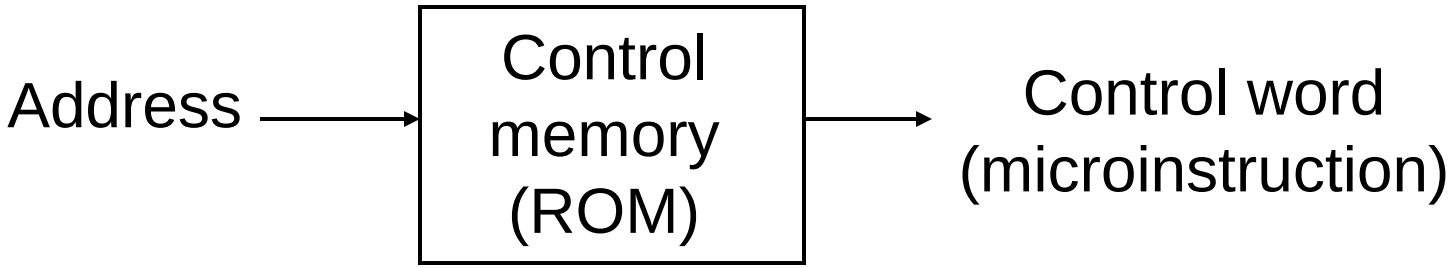
Read-only memory (ROM)

Content of word in ROM at given address specifies microinstruction

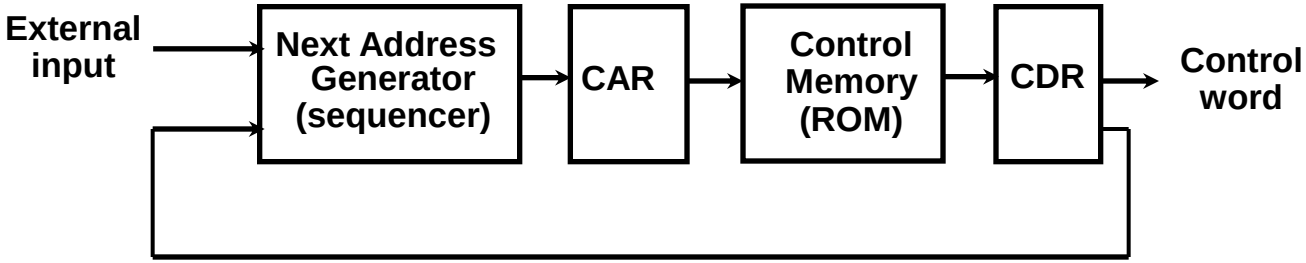
Each computer instruction initiates series of microinstructions (microprogram) in control memory

These microinstructions generate microoperations to

- Fetch instruction from main memory
- Evaluate effective address
- Execute operation specified by instruction
- Return control to fetch phase for next instruction



MICROPROGRAMMED CONTROL ORGANIZATION



Control memory

- Contains microprograms (set of microinstructions)
- Microinstruction contains
  - Bits initiate microoperations
  - Bits determine address of next microinstruction

Control address register (CAR)

- Specifies address of next microinstruction

MICROPROGRAMMED CONTROL ORGANIZATION

Next address generator (microprogram sequencer)

- Determines address sequence for control memory

Microprogram sequencer functions

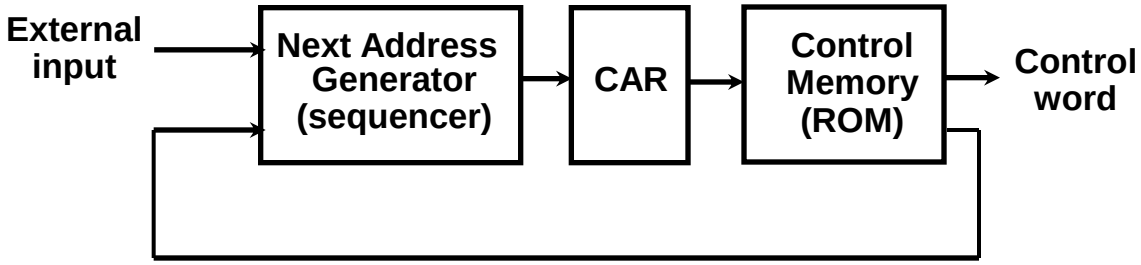
- Increment CAR by one
- Transfer external address into CAR
- Load initial address into CAR to start control operations

MICROPROGRAMMED CONTROL ORGANIZATION

Control data register (CDR)- or pipeline register

- Holds microinstruction read from control memory
- Allows execution of microoperations specified by control word simultaneously with generation of next microinstruction

Control unit can operate without CDR



MICROPROGRAM ROUTINES

Routine

- Group of microinstructions stored in control memory

Each computer instruction has its own microprogram routine to generate microoperations that execute the instruction

MICROPROGRAM ROUTINES

Subroutine

- Sequence of microinstructions used by other routines to accomplish particular task

Example

- Subroutine to generate effective address of operand for memory reference instruction

Subroutine register (SBR)

- Stores return address during subroutine call

CONDITIONAL BRANCHING

Branching from one routine to another depends on status bit conditions

Status bits provide parameter info such as

- Carry-out of adder
- Sign bit of number
- Mode bits of instruction

Info in status bits can be tested and actions initiated based on their conditions: 1 or 0

Unconditional branch

- Fix value of status bit to 1

MAPPING OF INSTRUCTION

Each computer instruction has its own microprogram routine stored in a given location of the control memory

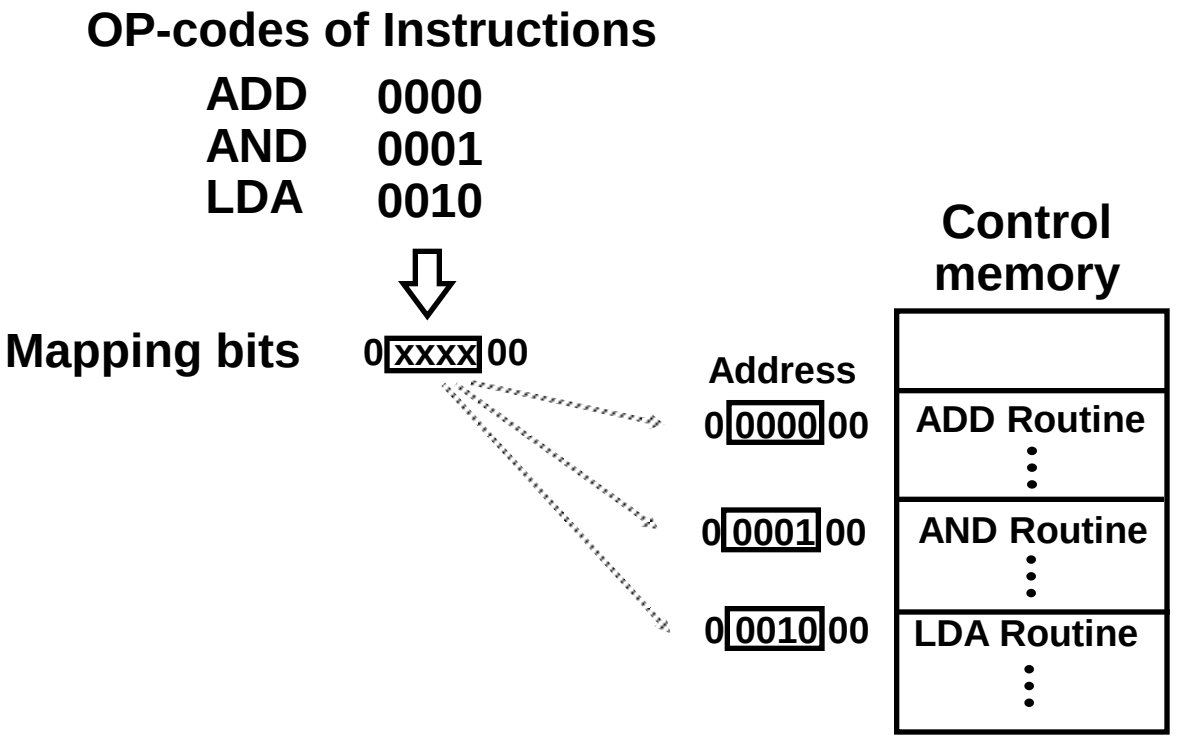
Mapping

- Transformation from instruction code bits to address in control memory where routine is located

MAPPING OF INSTRUCTION

Example

- Mapping 4-bit operation code to 7-bit address

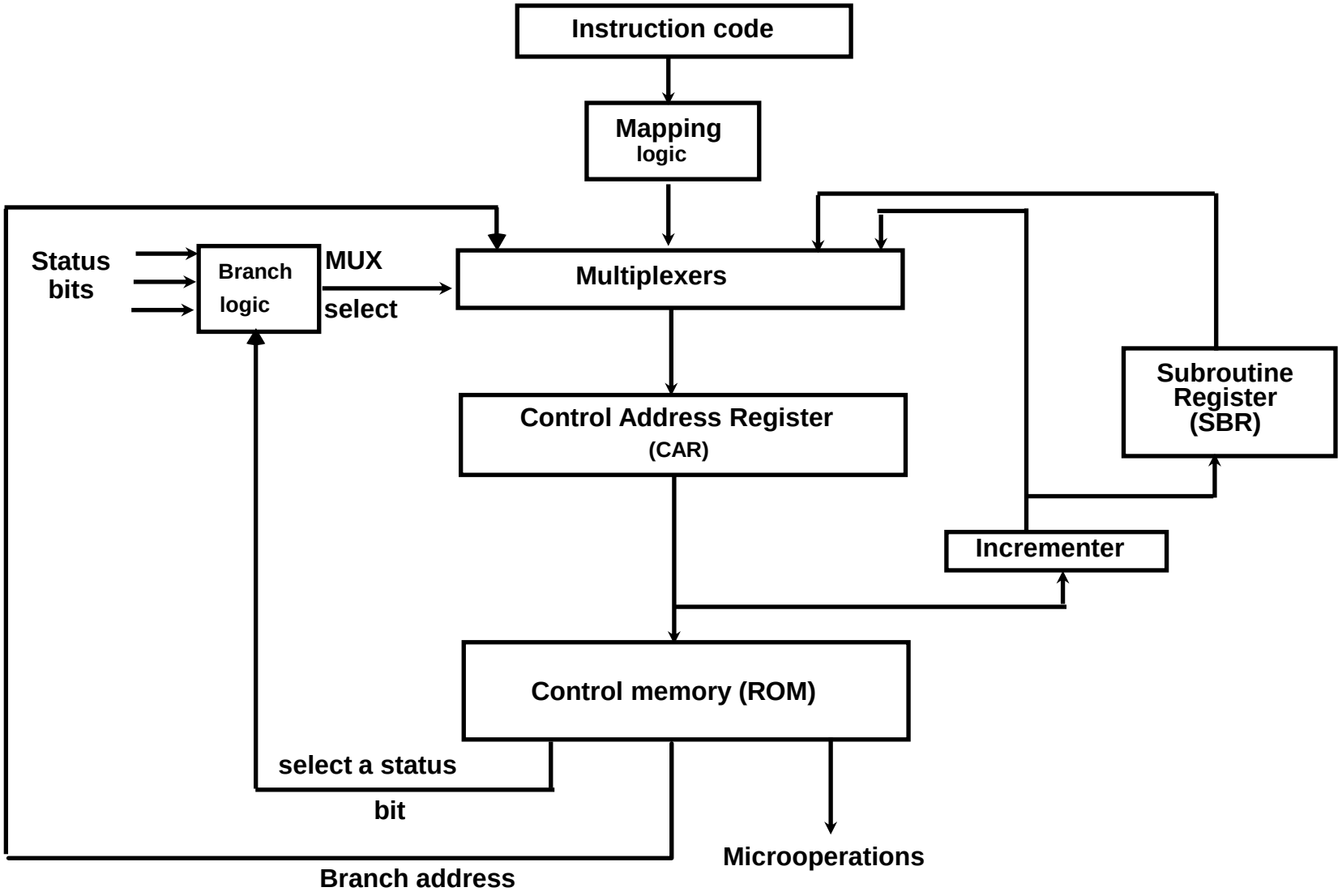


ADDRESS SEQUENCING

Address sequencing capabilities required in control unit

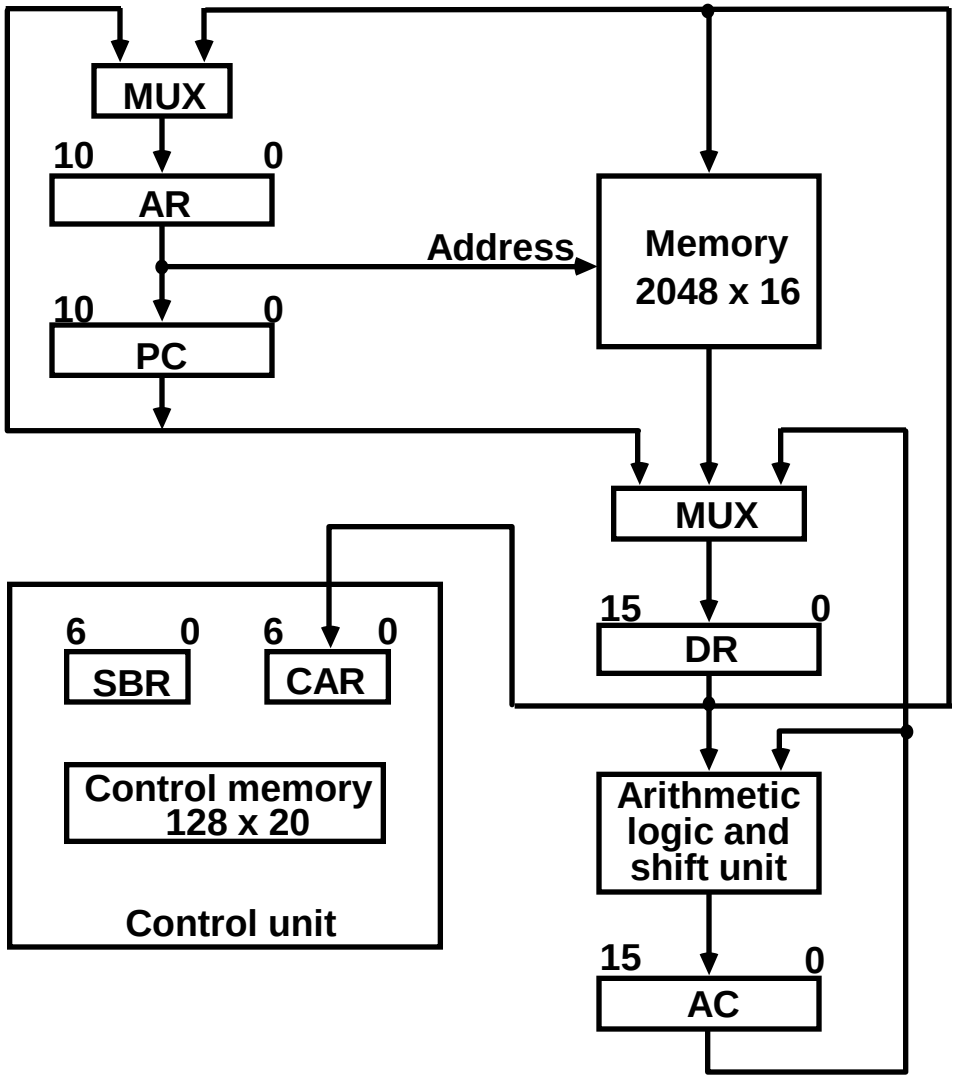
- Incrementing CAR
- Unconditional or conditional branch, depending on status bit conditions
- Mapping from bits of instruction to address for control memory
- Facility for subroutine call and return

ADDRESS SEQUENCING



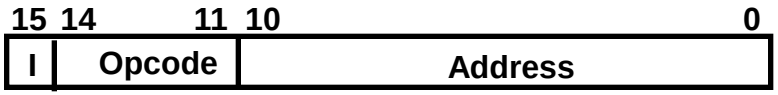
MICROPROGRAM EXAMPLE

Computer Configuration



MICROPROGRAM EXAMPLE

Computer instruction format

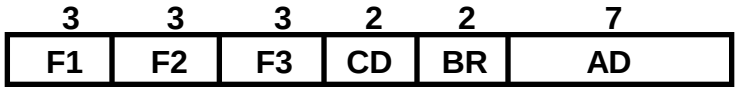


Four computer instructions

Symbol	OP-code	Description
ADD	0000	AC ← AC + M[EA]
BRANCH	0001	if (AC < 0) then (PC ← EA)
STORE	0010	M[EA] ← AC
EXCHANGE	0011	AC ← M[EA], M[EA] ← AC

EA is the effective address

Microinstruction Format



F1, F2, F3: Microoperation fields
CD: Condition for branching
BR: Branch field
AD: Address field

MICROINSTRUCTION FIELDS

F1	Microoperation	Symbol
000	None	NOP
001	AC ← AC + DR	ADD
010	AC ← 0	CLRAC
011	AC ← AC + 1	INCAC
100	AC ← DR	DRTAC
101	AR ← DR(0-10)	DRTAR
110	AR ← PC	PCTAR
111	M[AR] ← DR	WRITE

F2	Microoperation	Symbol
000	None	NOP
001	AC ← AC - DR	SUB
010	AC ← AC ∨ DR	OR
011	AC ← AC ∧ DR	AND
100	DR ← M[AR]	READ
101	DR ← AC	ACTDR
110	DR ← DR + 1	INCDR
111	DR(0-10) ← PC	PCTDR

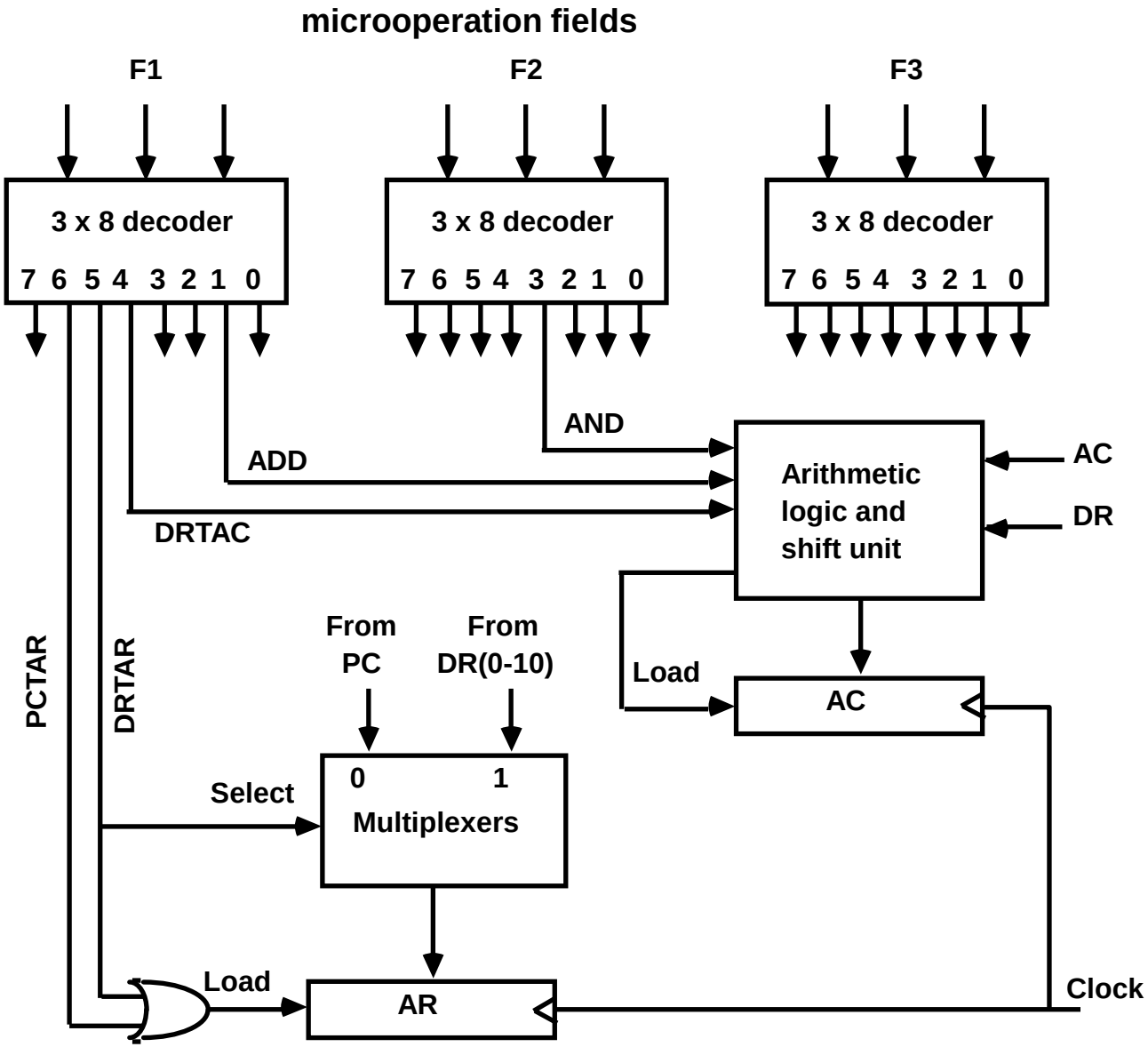
F3	Microoperation	Symbol
000	None	NOP
001	AC ← AC ⊕ DR	XOR
010	AC ← AC’	COM
011	AC ← shl AC	SHL
100	AC ← shr AC	SHR
101	PC ← PC + 1	INCPC
110	PC ← AR	ARTPC
111	Reserved	

MICROINSTRUCTION FIELDS

CD	Condition	Symbol	Comments
00	Always = 1	U	Unconditional branch
01	DR(15)	I	Indirect address bit
10	AC(15)	S	Sign bit of AC
11	AC = 0	Z	Zero value in AC

BR	Symbol	Function
00	JMP	CAR ← AD if condition = 1
		CAR ← CAR + 1 if condition = 0
01	CALL	CAR ← AD, SBR ← CAR + 1 if condition = 1
		CAR ← CAR + 1 if condition = 0
10	RET	CAR ← SBR (Return from subroutine)
11	MAP	CAR(2-5) ← DR(11-14), CAR(0,1,6) ← 0

DESIGN OF CONTROL UNIT



MICROPROGRAM SEQUENCER