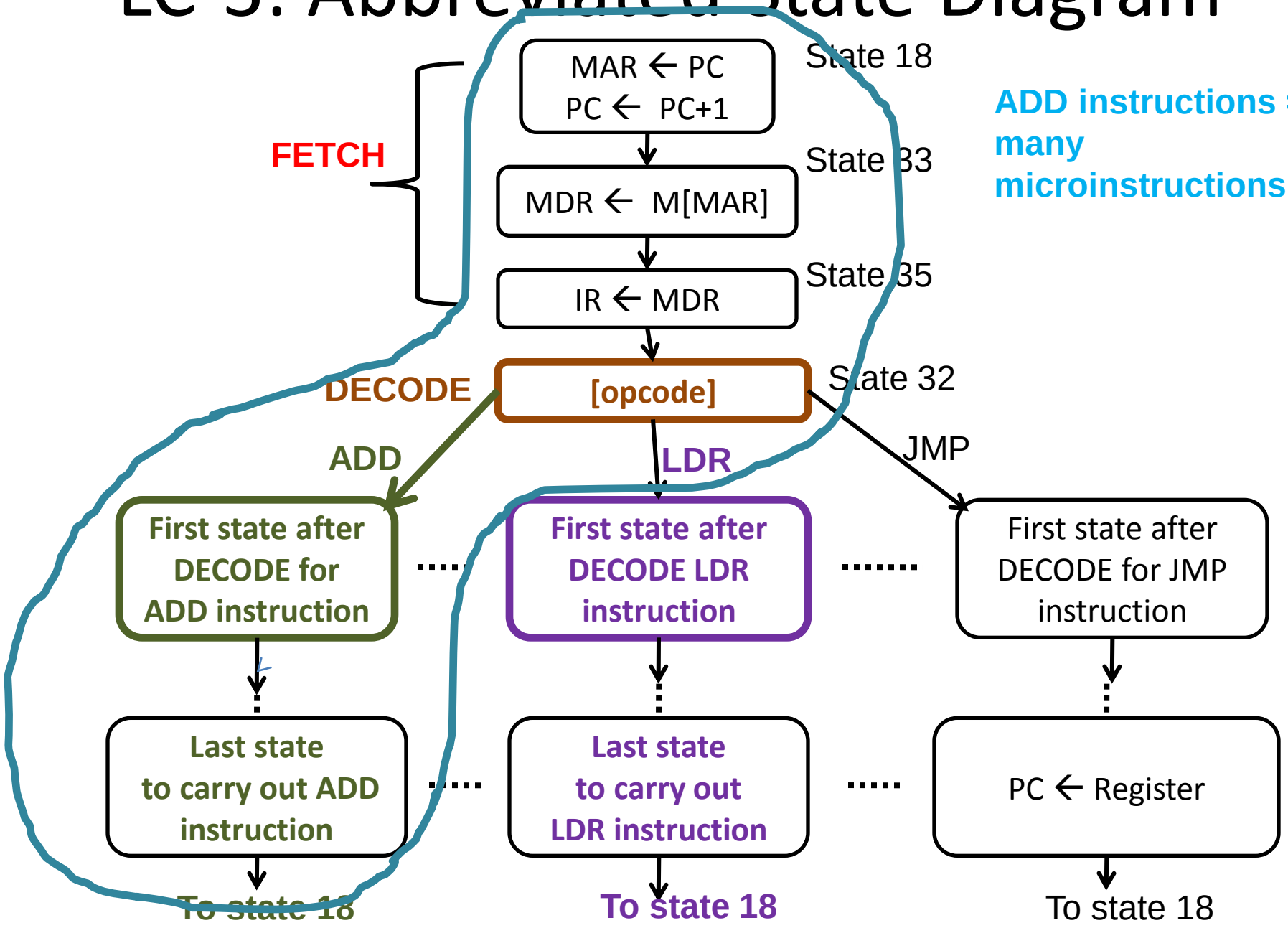
ECE290 Fall 2012
Lecture 22

Dr. Zbigniew Kalbarczyk

Today

- LC-3 Micro-sequencer (the control store)
- LC-3 Micro-programmed control memory
- LC-3 Micro-instruction format
- LC -3 Micro-sequencer (the circuitry)
- LC -3 Micro-sequencer control
- Determining next state

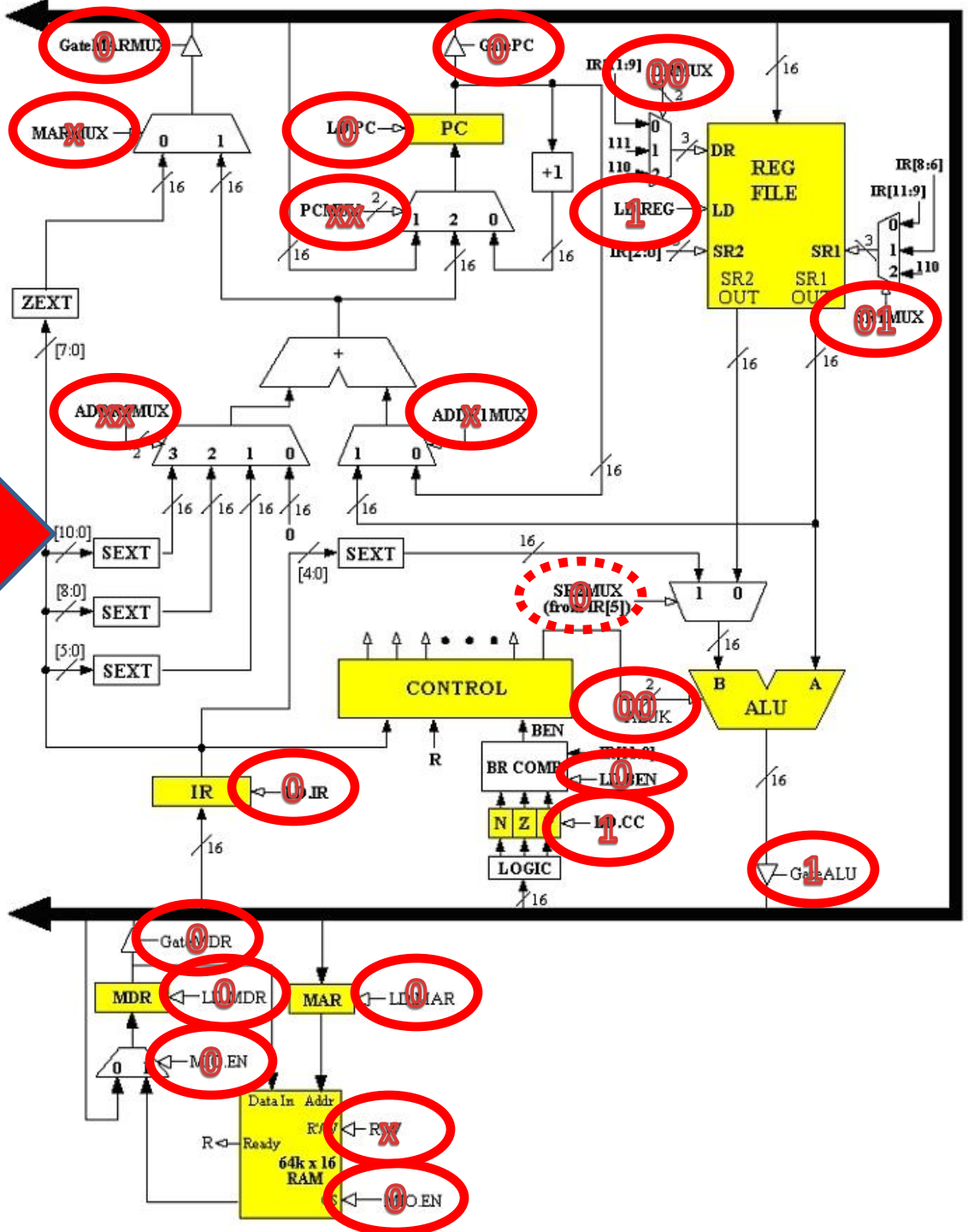
LC-3: Abbreviated State Diagram



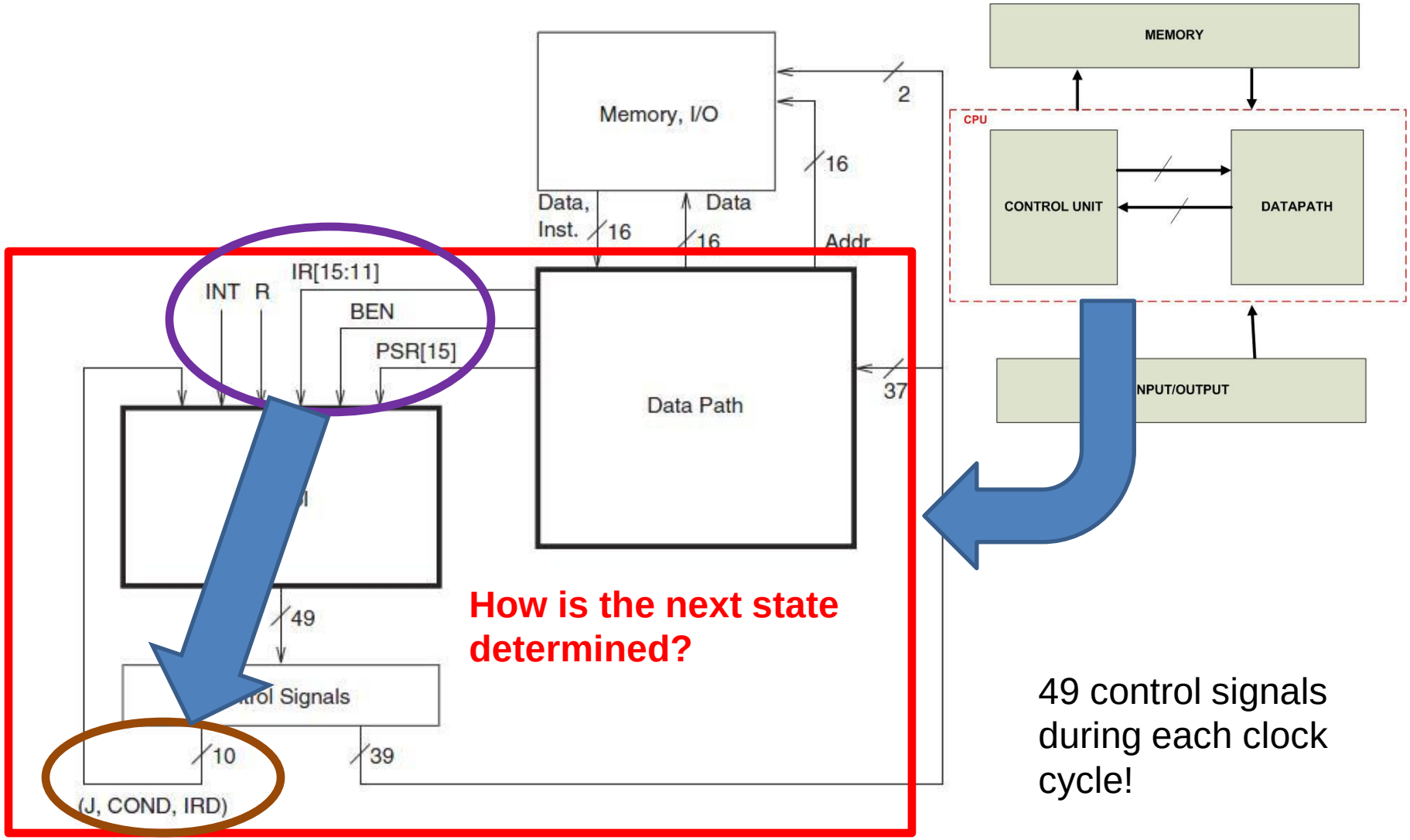
From th

- Control Signals for **one microinstruction** of **ADD instruction**
- Microinstruction:
  - DR ← SR1 +SR2; Load NZP
  - **This does not include fetch, decode, and other execution phase microinstructions needed to perform ADD instruction!!!**

One Microinstruction



LC-3: Micro-Architecture

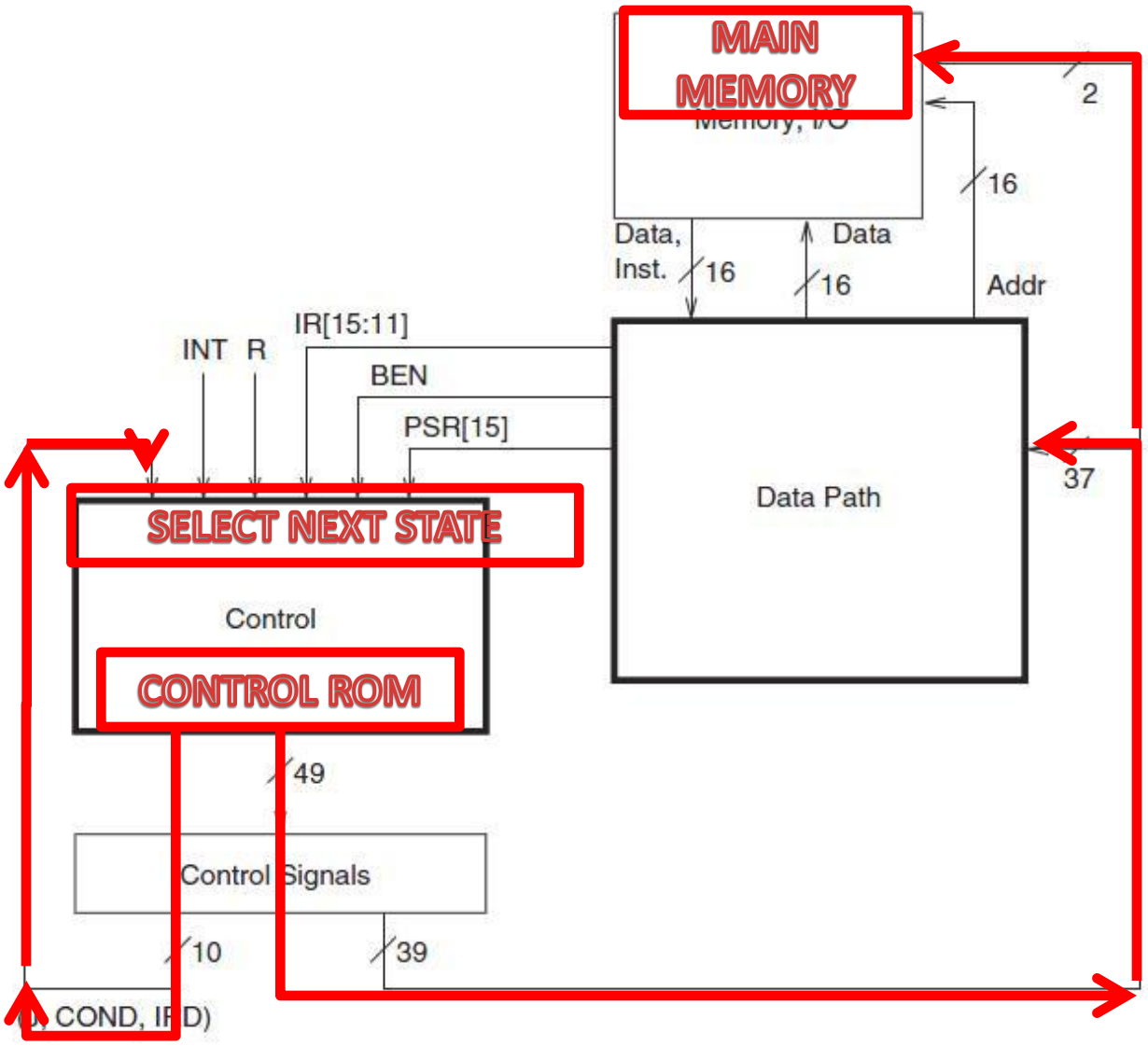


Instructions vs Microinstructions

- Instruction: assembly or binary representation of an instruction
  - Each instruction corresponds to one or more microinstructions needed to go through fetch, decode and execute phases
- Each microinstruction consists of bits corresponding to
  - Next state information (check the state diagram arrows between states)
  - Control word (control signals we just went through)

LC-3 Micro-instructions

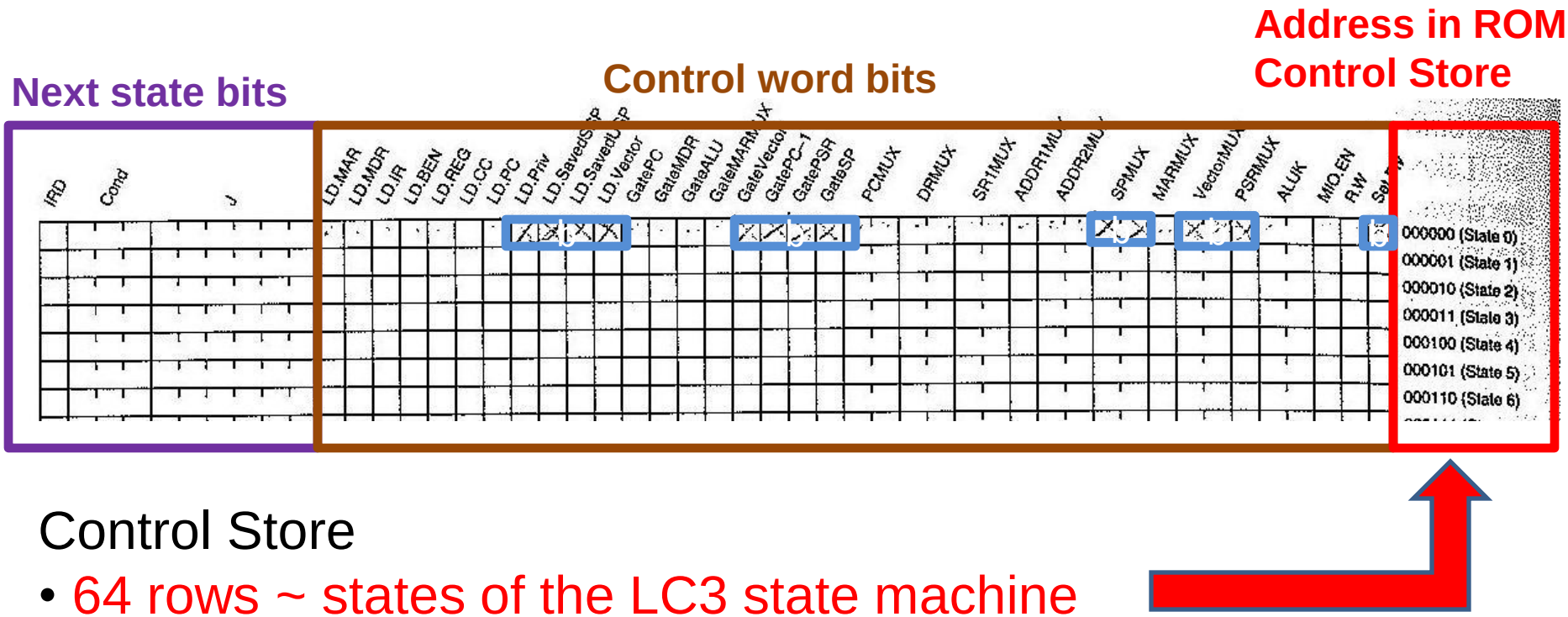
- Micro-instruction
  - Next state information
  - Control word



Instructions are in Main Memory

Microinstructions are in control store: Control ROM Memory

Control Store for LC3 Architecture



Control Store

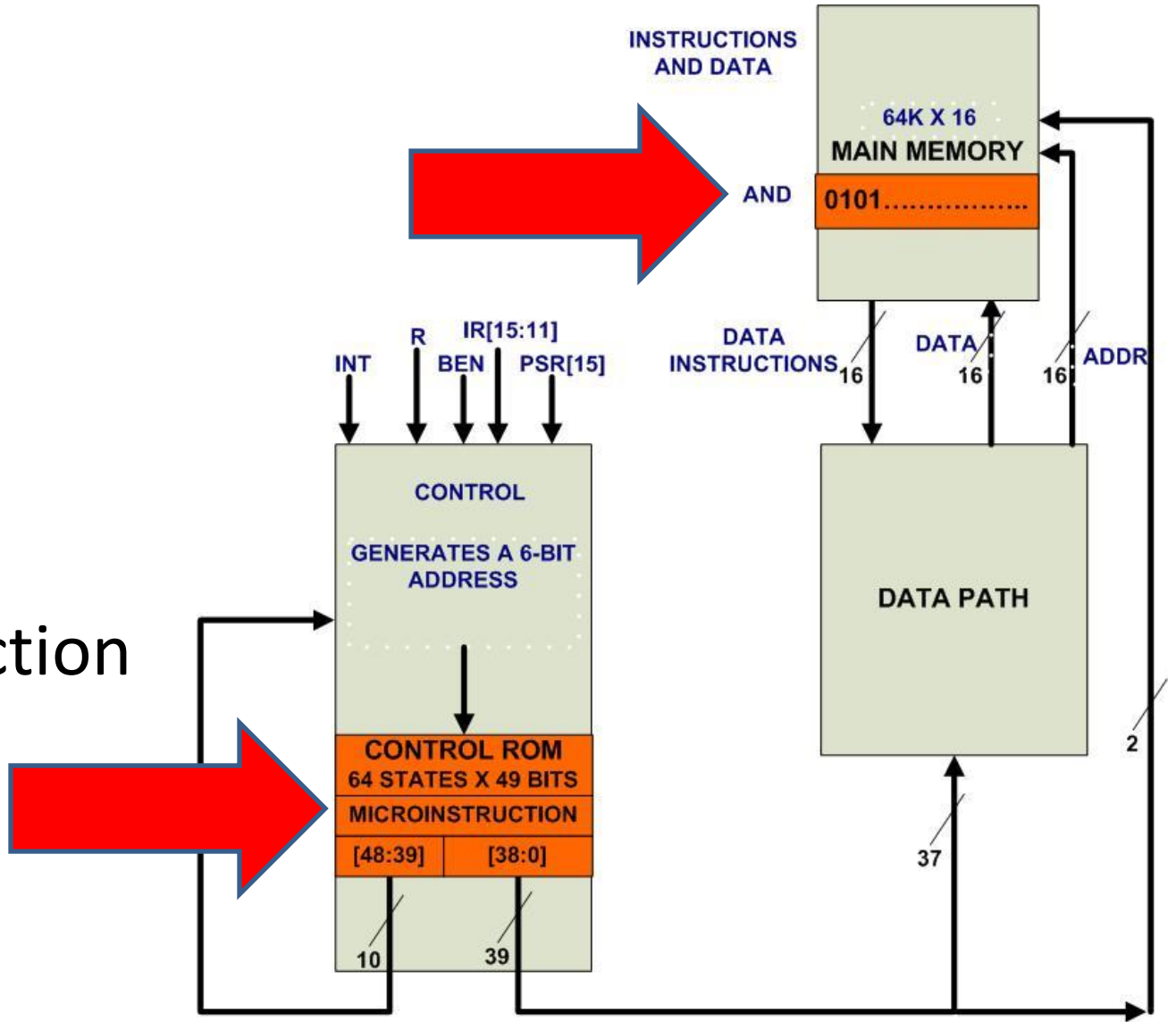
- 64 rows ~ states of the LC3 state machine
- 49 columns ~ 10 sequence control (signals from the current microinstruction) and 39 data path control
  - 39 control signals = 25 operation execution + 14 interrupt execution

THE CONTROL STORE: MICRO-SEQUENCING

ADDR	48								39	38 control signals 0	
	0	0	1	0	0	1	0	0	1	0	BR1
	0										
2	0	0	0	0	0	1	1	0	0	1	MAR ← PC + sext(off9) LD1
18	0	1	0	1	1	0	0	0	0	1	fetch1
22	0	0	0	0	0	1	0	0	1	0	BR2
	0										
	0										
25	0	0	0	1	0	1	1	0	0	1	MDR ← M[MAR] LD2
	0										
27	0	0	0	0	0	1	0	0	1	0	DR ← MDR, set CC LD3
32	1	X	X	X	X	X	X	X	X	X	decode
33	0	0	0	1	1	0	0	0	0	1	fetch2
	0										
35	0										fetch3
	0										

Overview: Instruction Versus Micro-instruction

- Instruction
- Micro-Instruction



Review: Status Signals from Data Path to Control

- **IR[15:11]** are the opcode bits
- **PSR[15**] is the bit [15] of the Processor Status Register, which indicates whether the current program is executing with supervisor or user privileges.
- **BEN** is to indicate whether or not a BR (conditional branch) should be taken.
- **INT** is to indicate that some external device of higher priority than the executing process requests service.
- **R** is to indicate the end of a memory operation.
- (J,COND, IRD) to provide signals from the current microinstruction

LC3: Control Structure

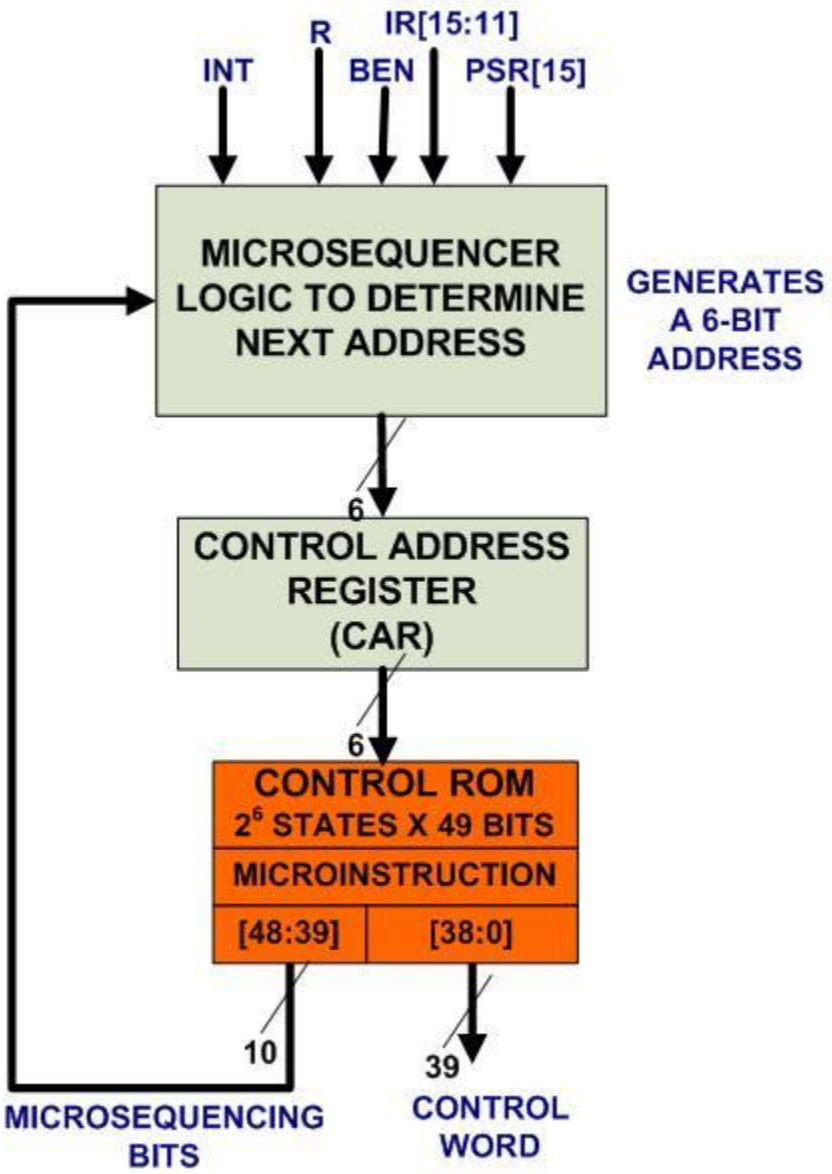
Micro-programmed Control Unit

Why 6 bit address?

A: Addressing 52 states need 6 bit address

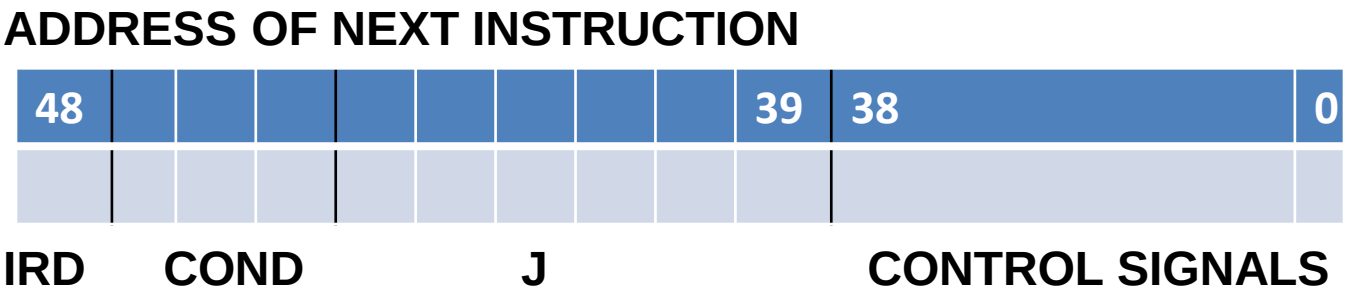
Why do we need CAR?

A: Similar to MAR – need to store the address



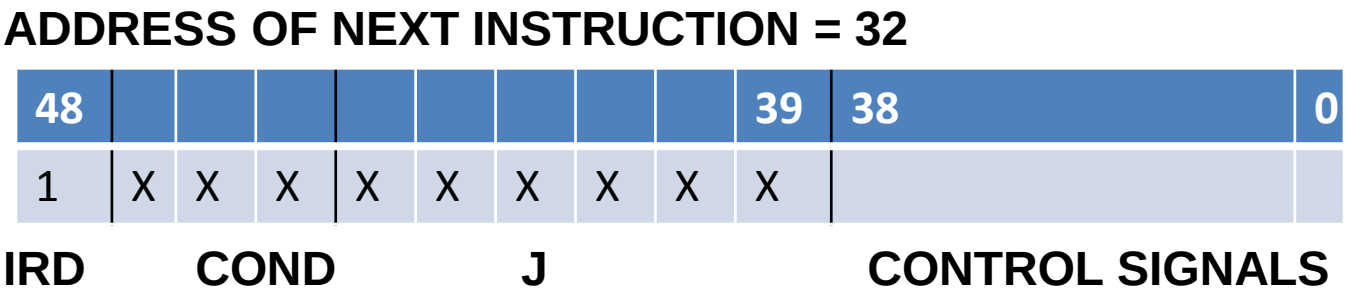
LC3 Micro-Instruction Format

- Each Micro-instruction has the default next state location encoded in J[5:0]
- This next state location can be modified depending on the values of IRD and COND[2:0]



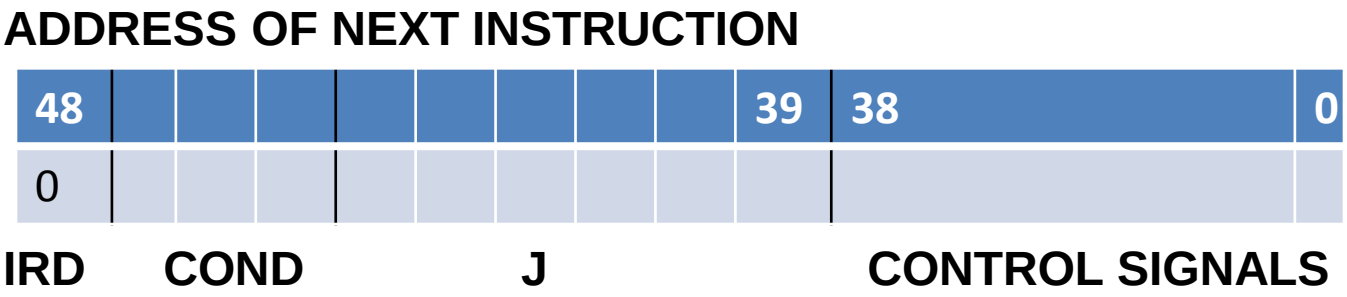
LC3 Micro-Instruction Format: IRD=1

- If IRD = 1
  - This is the control vector for state 32
  - The next address will be 00 concatenated with IR[15:12] (00 || IR[15:12])



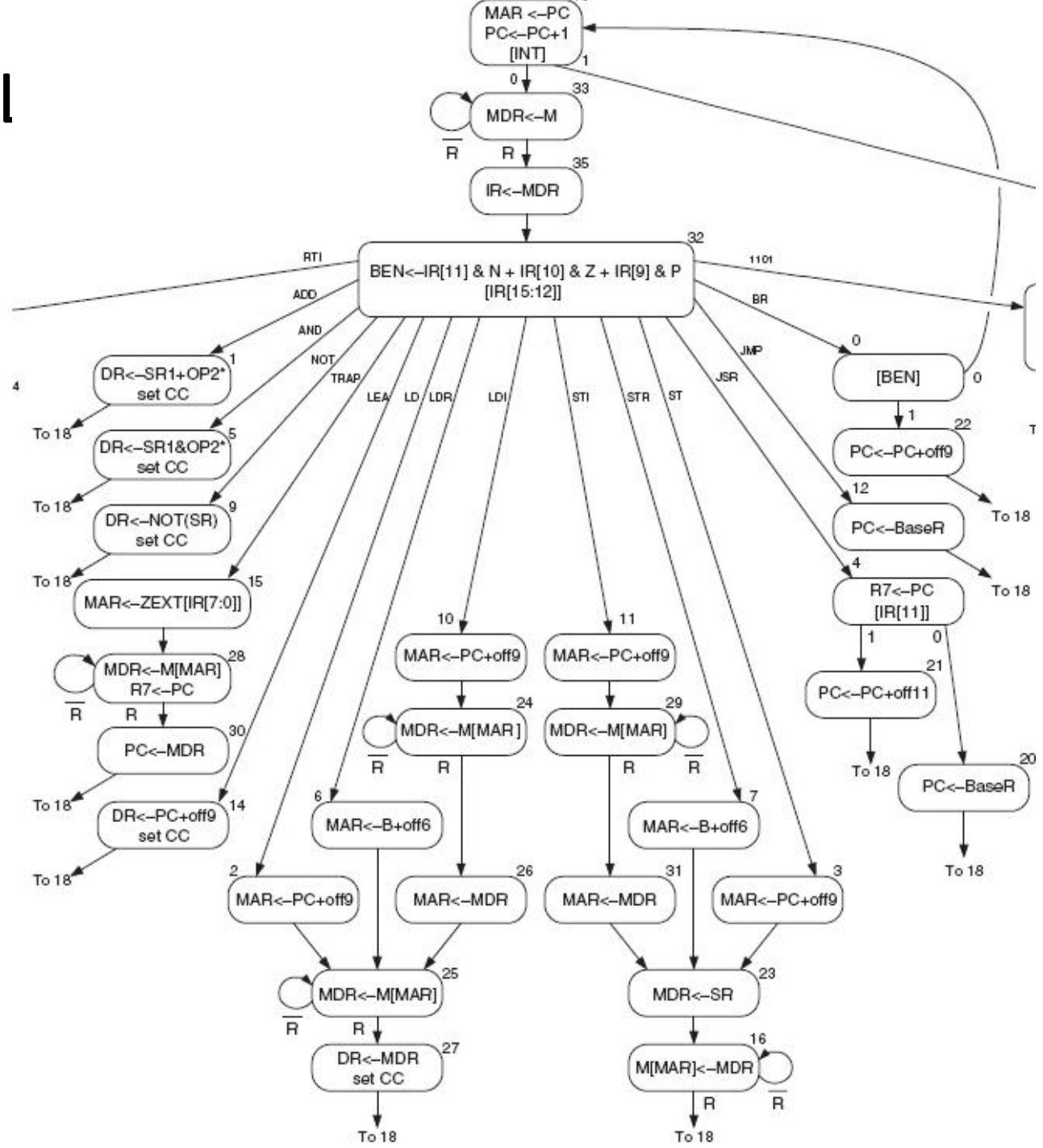
LC3 Micro-Instruction Format: IRD=0

- If IRD = 0
  - Get the branch address J[5:0] of micro-instruction from the control word
  - Use COND[2:0] and signals R, BEN, INT, IR[11] and PSR[15] to modify the branch address



I

- Every state requires at least one clock cycle
- State waiting for memory operations to complete may require more clock cycles



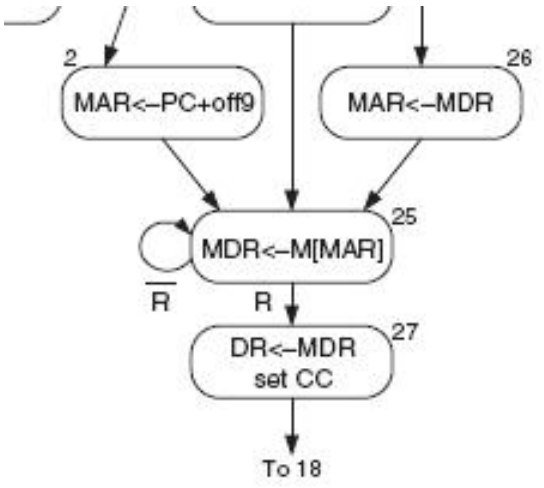
Example: Execute Phase of LD

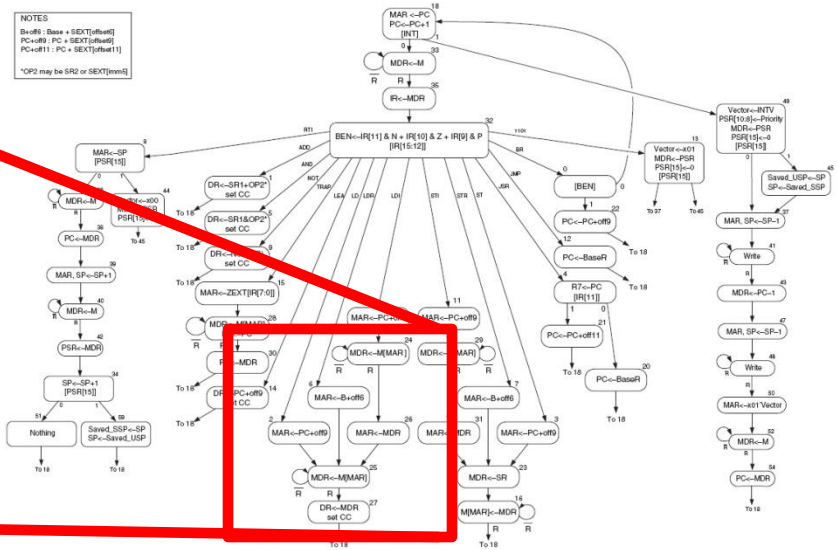
- State 2: MAR ← PC + sext(offset9)
- From State 2: always go to State 25 (011001)

ADDRESS OF NEXT INSTRUCTION

48	47	46	45	44	43	42	41	40	39	38	0
0	0	0	0	0	1	1	0	0	1		

IRD **COND** **J** **CONTROL SIGNALS**





Example: Execute Phase of LD

- State 25: MDR ← M[MAR]
- If R=0 (memory not ready) go to State 25
  - Next state (NS) = 011001 (25)
- If R=1 go to State 27 (011010)

Control Address 25

48	47	46	45	44	43	42	41	40	39	38	0
0	0	0	1	0	1	1	0	0	1		

IRD **COND** **J** **CONTROL SIGNALS**

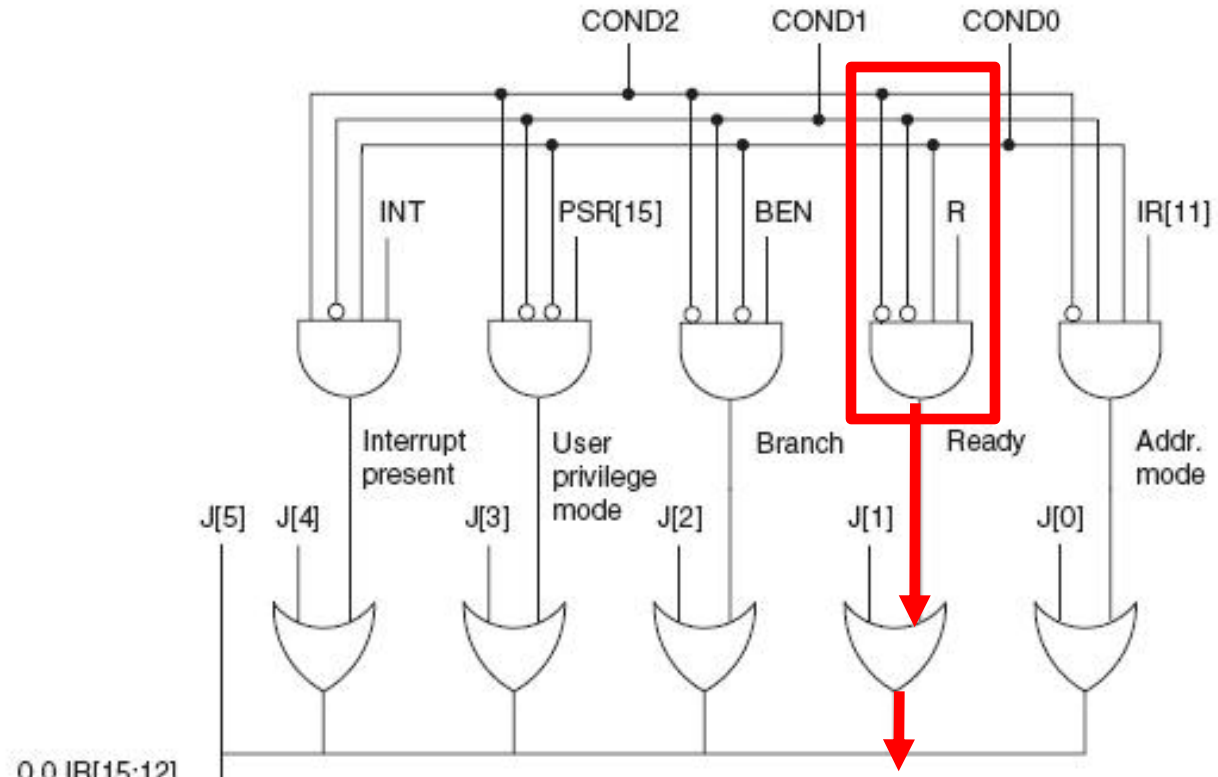
If R=1 then NS = 011011

Control Address 27

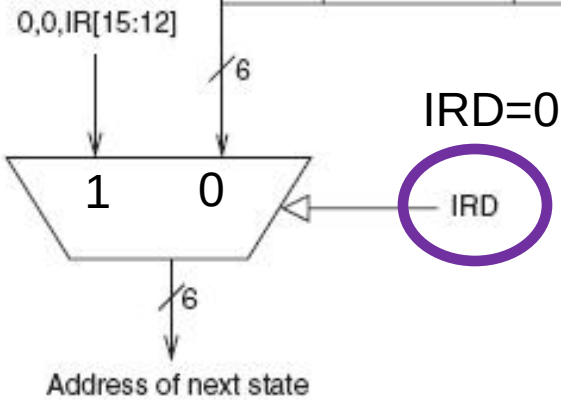
48	47	46	45	44	43	42	41	40	39	38	0
0	0	0	0	0	1	0	0	1	0		

IRD **COND** **J** **CONTROL SIGNALS**

Micro-sequencer Control



If R =1 then
Address of next state = (COND2)' (COND1)' COND0 R

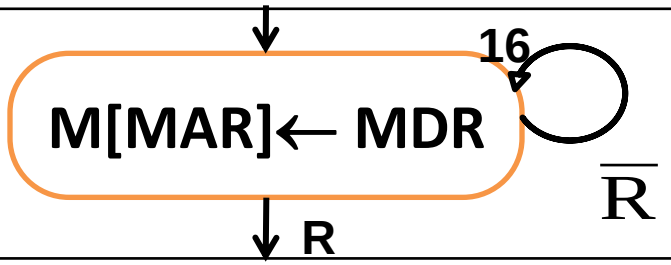
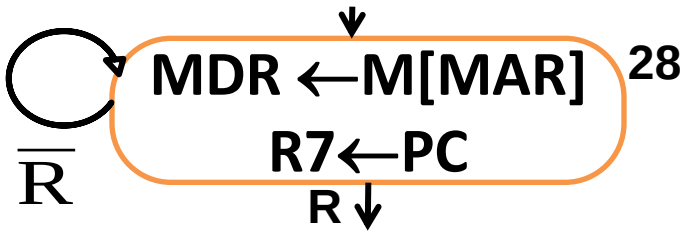
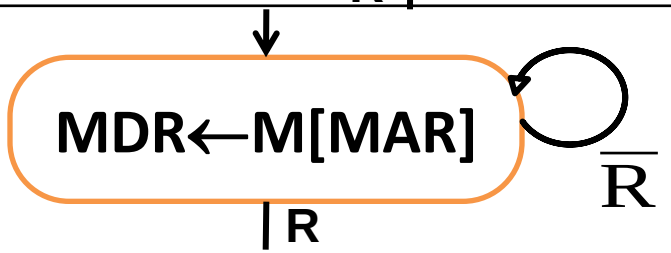


Control Address 25

48	47	46	45	44	43	42	41	40	39	38	0
0	0	0	1	0	1	1	0	0	1		

IRD **COND** **J** **CONTROL SIGNALS**

Other Examples

Current Addr	If R=0, go to	If R=1, go to	State Diagram
16	16 (10000) ↑	010010 18	
28	28 (11100)	011110 30	
24	24 (11000) ↑	011010 26	
29	29 (11101) ↑	011111 31	

Exercise

- For the word stored at address 33 in the control ROM, give the most leftmost 10 bits – the micro-sequencing (next state) bits

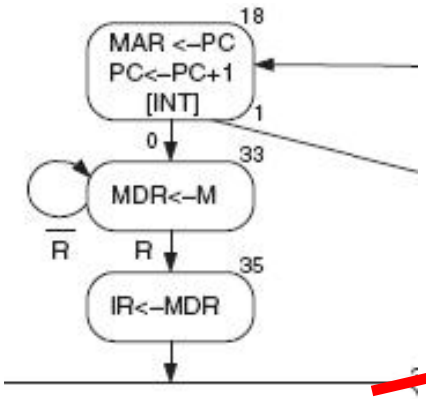
Control Address 33

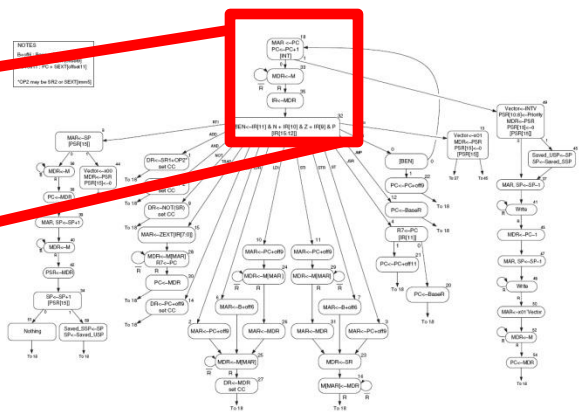
48	47	46	45	44	43	42	41	40	39	38	0
0	0	0	1	1	0	0	0	0	1		

IRD **COND** **J** **CONTROL SIGNALS**

If R=0 then next state=33

If R=1 then next state=35

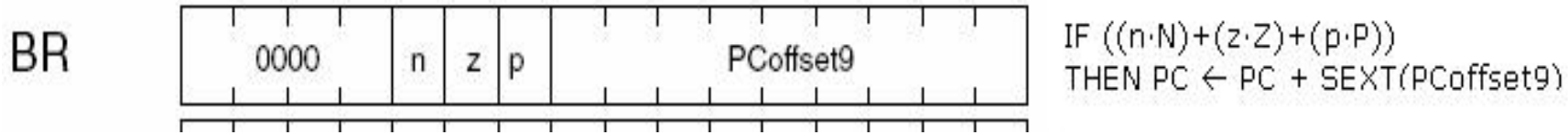




Micro-sequence Control

COND	TEST BIT COMPUTED	CONDITION TESTED	BRANCH
000	NO TEST	UNCONDITIONAL	NO CHANGE
001	R	MEMORY READ COMPLETED	J[1]=1
010	BEN	SUCESFUL CONDITIONAL BRANCH	J[2]=1
011	IR[11]	ADDRESSING MODE	J[0]=1
100	PSR[15]	PRIVILEGED MODE	J[3]=1
101	INT	INTERRUPT MODE	J[4]=1

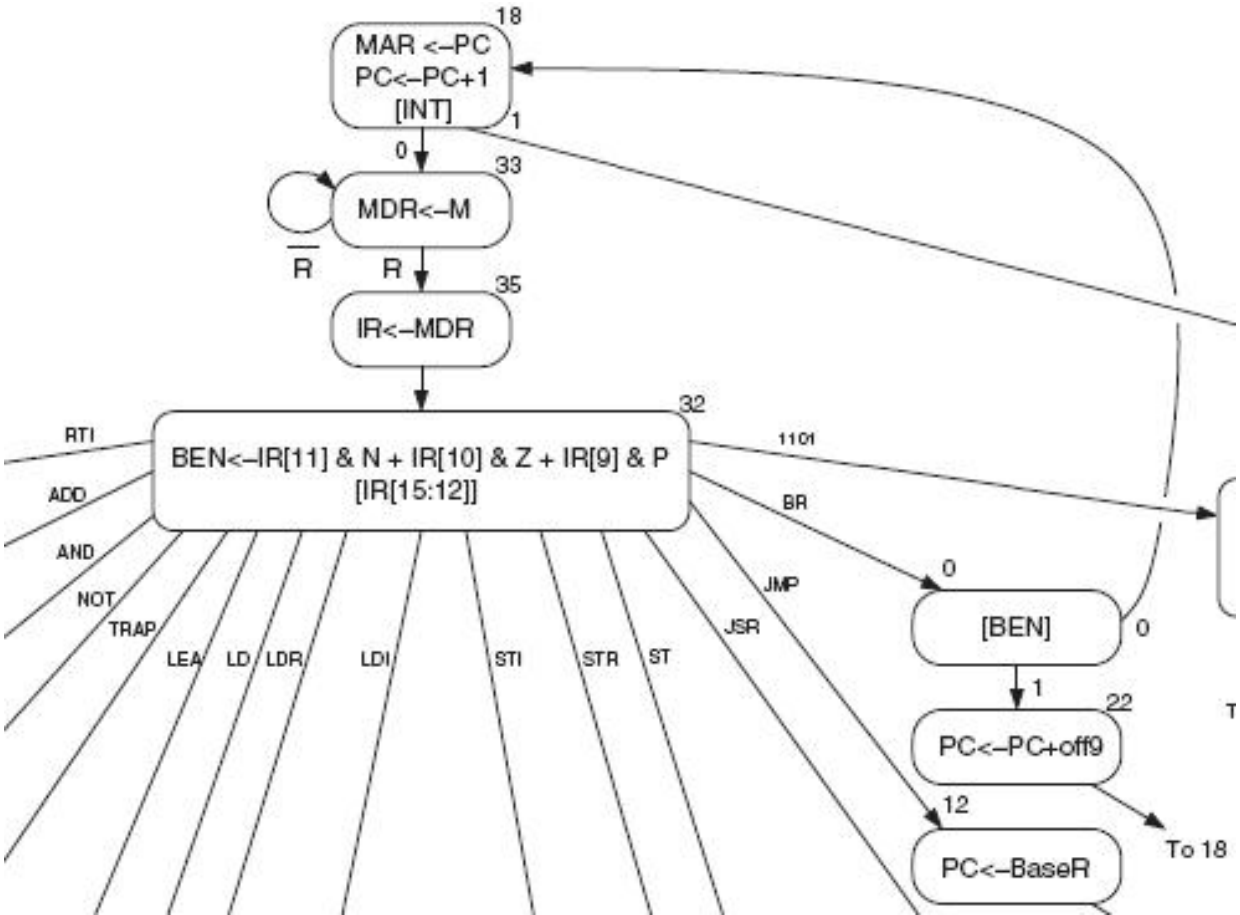
BR Instruction



- BEN: Branch Enable flip-flop
  - BEN = 1 if branch condition in BR instruction matches values of NZP
  - BEN = 0 otherwise

BR Instruction (cont.)

- In State 32 (instruction decode)
  - BEN ← IR[11] N + IR[10] Z + IR[9] P



Execute Phase of BRnz

- Example: BRnz has N=Z=1, P=0
  - Branches to PC + sext(offset9) if 1 N + 1 Z + 0 P = N + Z
- State 0
  - If BEN = 0 go to State 18 = 010010 (begin fetch)
  - If BEN = 1 go to State 22 = 010110 (22=18+4)

Control Address 18

48	47	46	45	44	43	42	41	40	39	38	0
0	0	1	0	0	1	0	0	1	0		

IRD **COND** **J** **CONTROL SIGNALS**

J[5]=0 J[4]=1 J[3]=0 J[2]=0+BEN J[1]=1 J[0]=0

Execute Phase of BRnz (cont.)

- State 22
  - PC ← PC + sext(offset9)
  - Go to State 18 (begin fetch)

Control Address 22

48	47	46	45	44	43	42	41	40	39	38	0
0	0	0	0	0	1	0	0	1	0		

IRD COND J CONTROL SIGNALS

Control Address 18

48	47	46	45	44	43	42	41	40	39	38	0
0	1	0	1	1	0	0	0	0	1		

IRD COND J CONTROL SIGNALS