University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 120: Introduction to Computing

Extending LC-3 ISA

Can we add a new instruction to LC-3 ISA?

Yes!

As long as it can be carried out by the existing datapath.

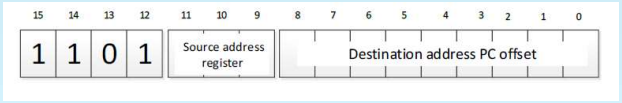
There are already 15 instructions and there is just one opcode (#13) not in use. So, let's use it.

Instruction Encoding

Let's add a new instruction, called **CP**.

This instruction copies a value from one memory location whose address is stored in the register specified by IR[11:9] bits to another memory location whose address is specified as a PC-relative offset by IR[8:0] bits.

The binary encoding of this new instruction is:



Instruction RTL and FSM

M[PC + SEXT(IR[8:0])] ← M[R(IR[11:9])]

Generate source address:

MAR ← R(IR[11:9])

Read source value:

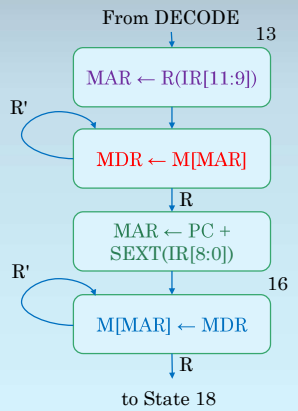
MDR ← M[MAR]

Generate destination address:

MAR ← PC + SEXT(IR[8:0])

Write value to the destination:

M[MAR] ← MDR

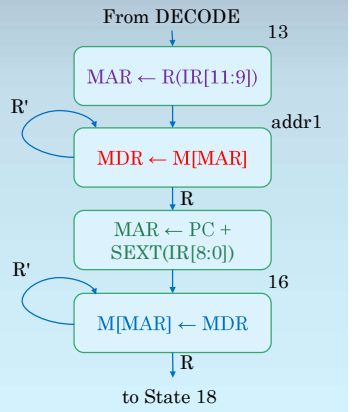


Microsequencing bits for CP instruction

CP1 does not branch.

The next state is CP2 (addr1)

state #		J	COND	IRD
13	CP1	addr1	000	0
	CP2			
	CP3			
	CP4			

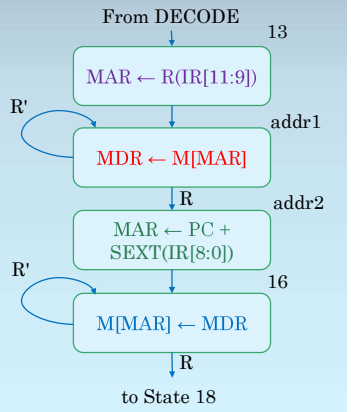


Microsequencing bits for CP instruction

CP2 branches on **R** (memory ready).

The next states are CP2 (addr1) and CP3 (addr2).

state #		J	COND	IRD
13	CP1	addr1	000	0
addr1	CP2	addr1	001	0
addr2	CP3			
	CP4			

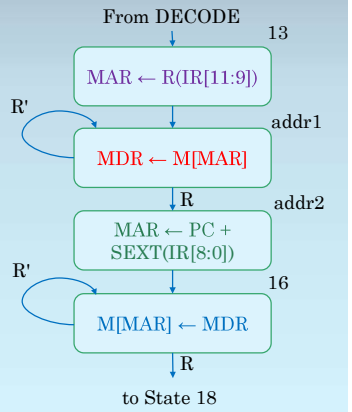


Microsequencing bits for CP instruction

CP3 does not branch.

The next state is CP4 (16).

state #		J	COND	IRD
13	CP1	addr1	000	0
addr1	CP2	addr1	001	0
addr2	CP3	16	000	0
16	CP4			

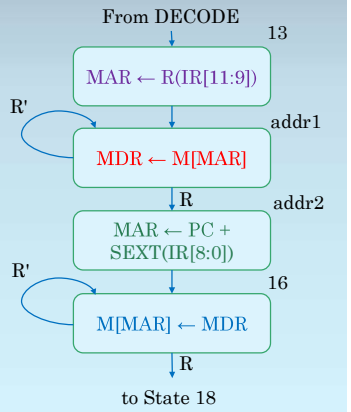


Microsequencing bits for CP instruction

CP4 branches on **R** (memory ready).

The next states are CP4 (16) and FETCH1 (18).

state #		J	COND	IRD
13	CP1	addr1	000	0
addr1	CP2	addr1	001	0
addr2	CP3	16	000	0
16	CP4	16	001	0



Microinstruction addresses

addr1 and addr2 must be unused.

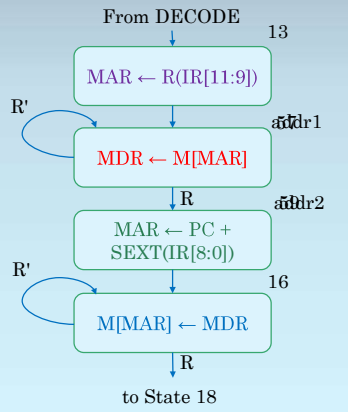
addr1 and addr2 must satisfy this relation:

$$addr2 = addr1 + \#2$$

or

$$b_5b_4b_3b_21b_0 = b_5b_4b_3b_20b_0 \text{ OR } 000010$$

For example, addr1 = 111001_2 and addr2 = 111011_2



Microsequencing bits for CP instruction

Finally

state #		J	COND	IRD
001101	CP1	111001	000	0
111001	CP2	111001	001	0
111011	CP3	010000	000	0
010000	CP4	010000	001	0