ICS 复习

Important parts

- Appendix A
  - Exclude exception
- Appendix C
  - Exclude exception

About chapter 10

- **Stack machines, Zero-address machine**
  - PPT 8-2
  - Chapter 10.2
- **Simulate a calculator with LC-3**

```
(1)     push    25
(2)     push    17
(3)     add
(4)     push    3
(5)     push    2
(6)     add
(7)     multiply
(8)     pop     E
```

OpMult, which will pop two values from the stack, multiply them, and push the result onto the stack.

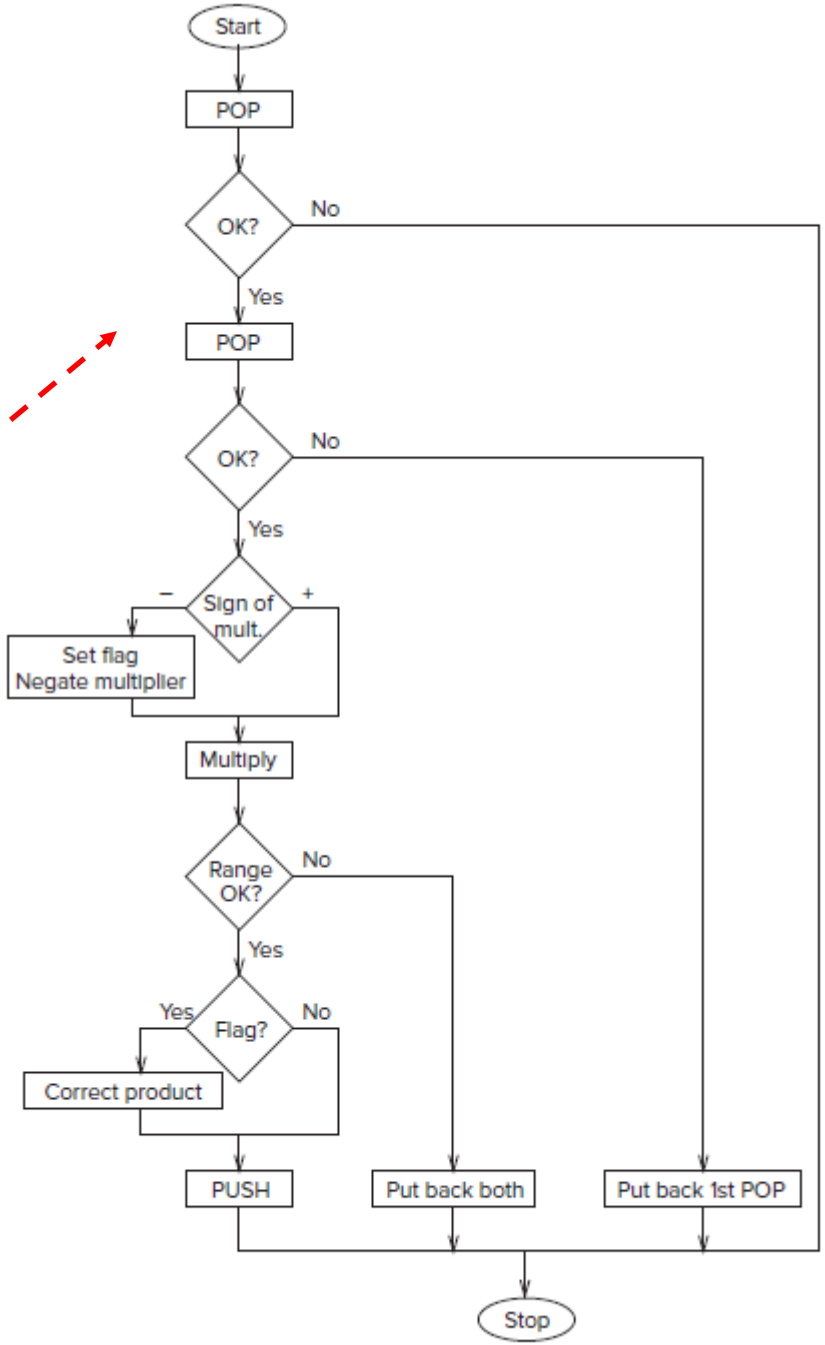


Figure 10.11 Flowchart for the OpMult subroutine.

About chapter 10

- **Data Type Conversion**
  - PPT 9-2
  - Chapter 10.1

Appendix C :Microarchitecture of the LC-3

- Time is divided into clock cycles.
- The cycle time of a microprocessor is the duration of a clock cycle.
  - A common cycle time for a microprocessor today is 0.33 nanoseconds, which corresponds to 3 billion clock cycles each second. We say that such a microprocessor is operating at a frequency of 3 gigahertz, or 3 GHz.
- We say, “at each instant of time,” but we really mean during each clock cycle.

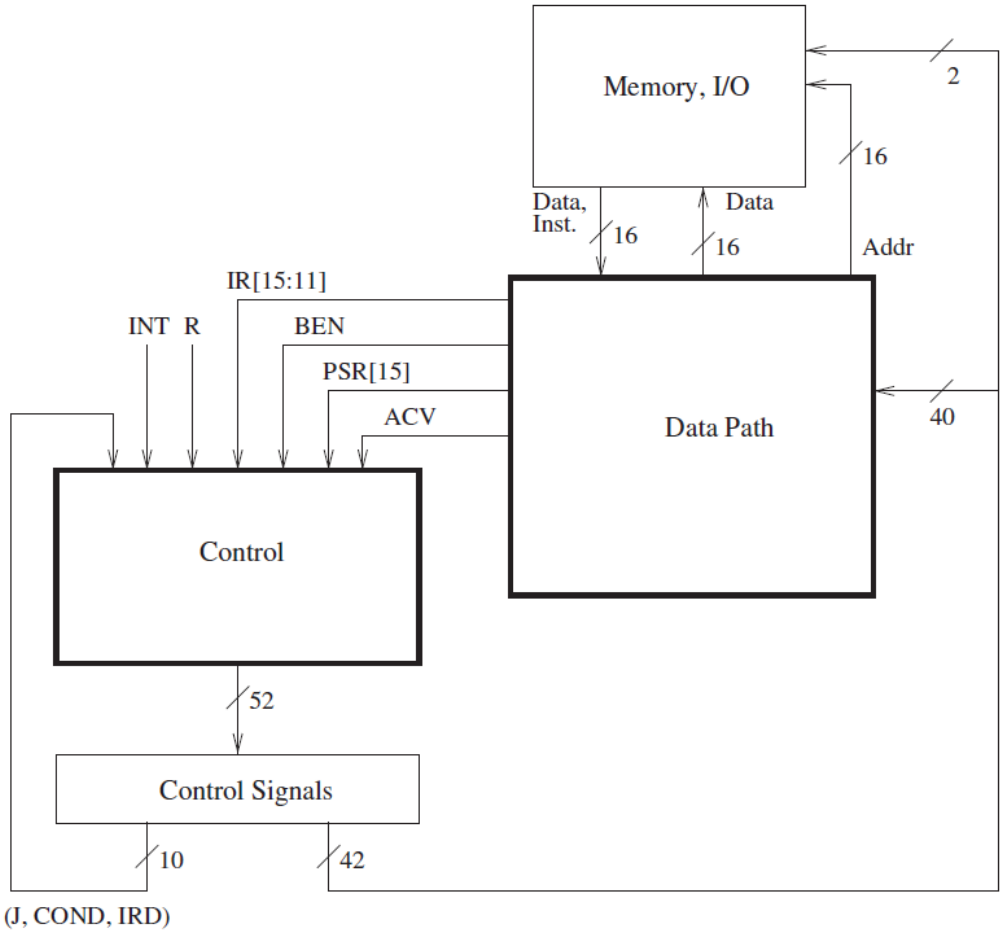


Figure C.1 Microarchitecture of the LC-3, major components.

Appendix C :Microarchitecture of the LC-3

- The control signals needed in the “next” clock cycle depend on the following:
  - 1. The control signals that are present during the current clock cycle.
  - 2. The LC-3 instruction that is being executed.
  - 3. The privilege mode of the program that is executing, and whether the processor has the right to access a particular memory location.
  - 4. If that LC-3 instruction is a BR, whether the conditions for the branch have been met (i.e., the state of the relevant condition codes).
  - 5. Whether or not an external device is requesting that the processor be interrupted.
  - 6. If a memory operation is in progress, whether it is completing during this cycle.

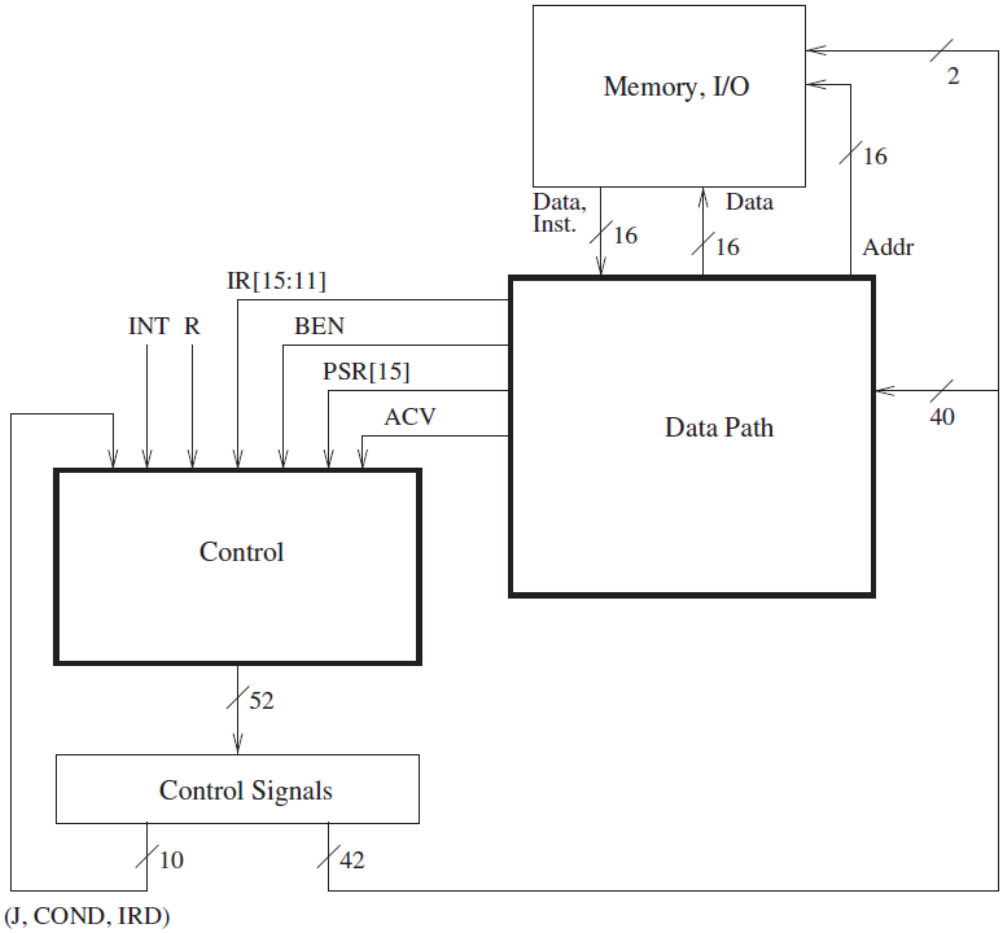


Figure C.1 Microarchitecture of the LC-3, major components.

Appendix C :Microarchitecture of the LC-3

1. J[5:0], COND[2:0], and IRD—ten bits of control signals provided by the current clock cycle.

2. IR[15:12], which identifies the opcode, and IR[11:11], which differentiates JSR from JSRR (i.e., the addressing mode for the target of the subroutine call).

3. PSR[15], bit [15] of the Processor Status Register, which indicates whether the current program is executing with supervisor or user privileges,

4. ACV, a signal that informs the processor that a process operating in User mode is trying to access a location in privileged memory. ACV stands for Access Control Violation. When asserted, it denies the process access to the privileged memory location.

5. BEN to indicate whether or not a BR should be taken.

6. INT to indicate that some external device of higher priority than the executing process requests service.

7. R to indicate the end of a memory operation.

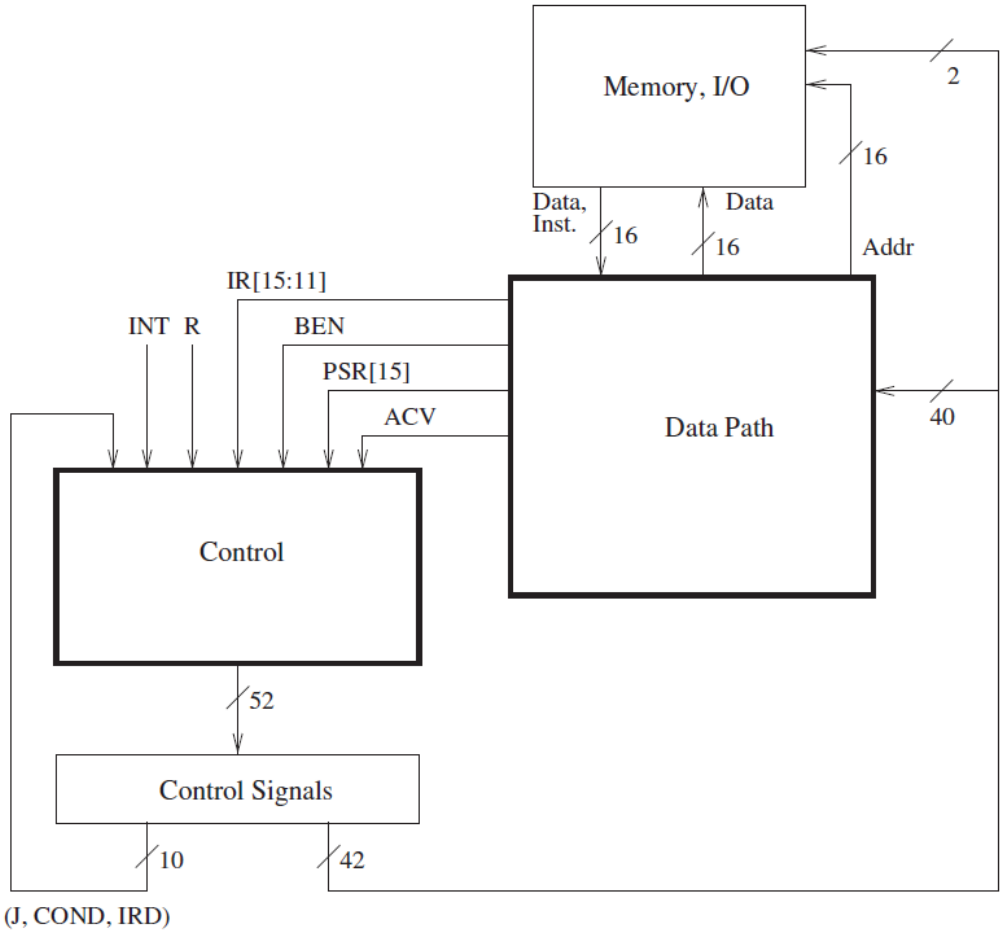
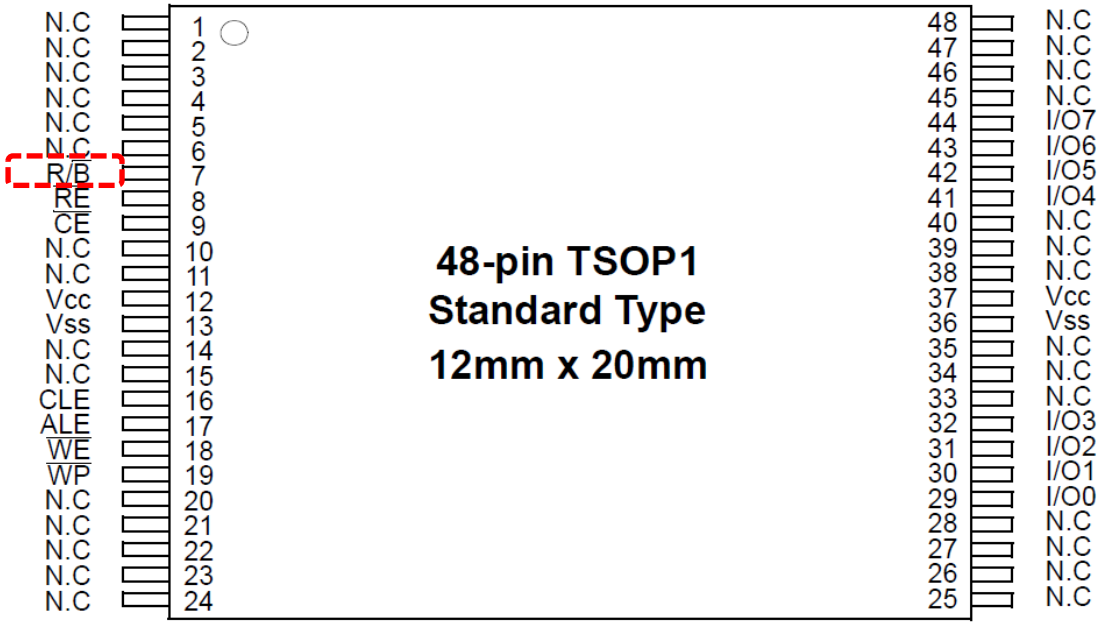


Figure C.1 Microarchitecture of the LC-3, major components.

About R

A flash memory chip

K9F2G08U0A-PCB0/PIB0



PIN DESCRIPTION

Pin Name	Pin Function
$I/O_0 \sim I/O_7$	**DATA INPUTS/OUTPUTS** The I/O pins are used to input command, address and data, and to output data during read operations. The I/O pins float to high-z when the chip is deselected or when the outputs are disabled.
CLE	**COMMAND LATCH ENABLE** The CLE input controls the activating path for commands sent to the command register. When active high, commands are latched into the command register through the I/O ports on the rising edge of the $\overline{WE}$ signal.
ALE	**ADDRESS LATCH ENABLE** The ALE input controls the activating path for address to the internal address registers. Addresses are latched on the rising edge of $\overline{WE}$ with ALE high.
$\overline{CE}$	**CHIP ENABLE** The $\overline{CE}$ input is the device selection control. When the device is in the Busy state, $\overline{CE}$ high is ignored, and the device does not return to standby mode in program or erase operation.
$\overline{RE}$	**READ ENABLE** The $\overline{RE}$ input is the serial data-out control, and when active drives the data onto the I/O bus. Data is valid tREA after the falling edge of $\overline{RE}$ which also increments the internal column address counter by one.
$\overline{WE}$	**WRITE ENABLE** The $\overline{WE}$ input controls writes to the I/O port. Commands, address and data are latched on the rising edge of the $\overline{WE}$ pulse.
$\overline{WP}$	**WRITE PROTECT** The $\overline{WP}$ pin provides inadvertent program/erase protection during power transitions. The internal high voltage generator is reset when the $\overline{WP}$ pin is active low.
R/$\overline{B}$	**READY/BUSY OUTPUT** The R/$\overline{B}$ output indicates the status of the device operation. When low, it indicates that a program, erase or random read operation is in process and returns to high state upon completion. It is an open drain output and does not float to high-z condition when the chip is deselected or when outputs are disabled.
Vcc	**POWER** Vcc is the power supply for device.
Vss	**GROUND**
N.C	**NO CONNECTION** Lead is not internally connected.

READY/BUSY OUTPUT

The R/B output indicates the status of the device operation. When low, it indicates that a program, erase or random read operation is in process and returns to high state upon completion.

Appendix C :Microarchitecture of the LC-3

- During each clock cycle,
  - 42 of these control signals determine the processing of information in the data path
  - the other **10** control signals combine with the **10** bits of additional information to determine which set of control signals will be required in the next clock cycle.
- These 52 control signals specify the state of the control structure of the LC-3 microarchitecture

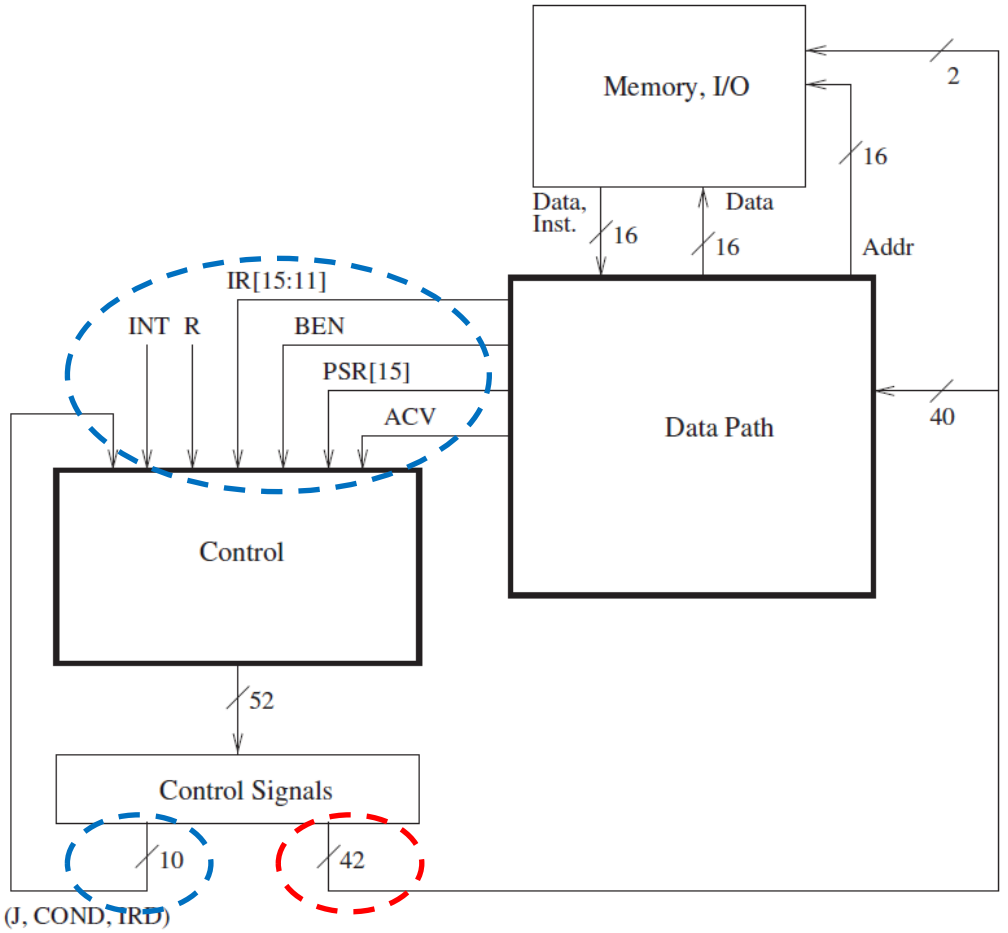


Figure C.1 Microarchitecture of the LC-3, major components.

The state machine

- [INT]
- [ACV]
- set CC

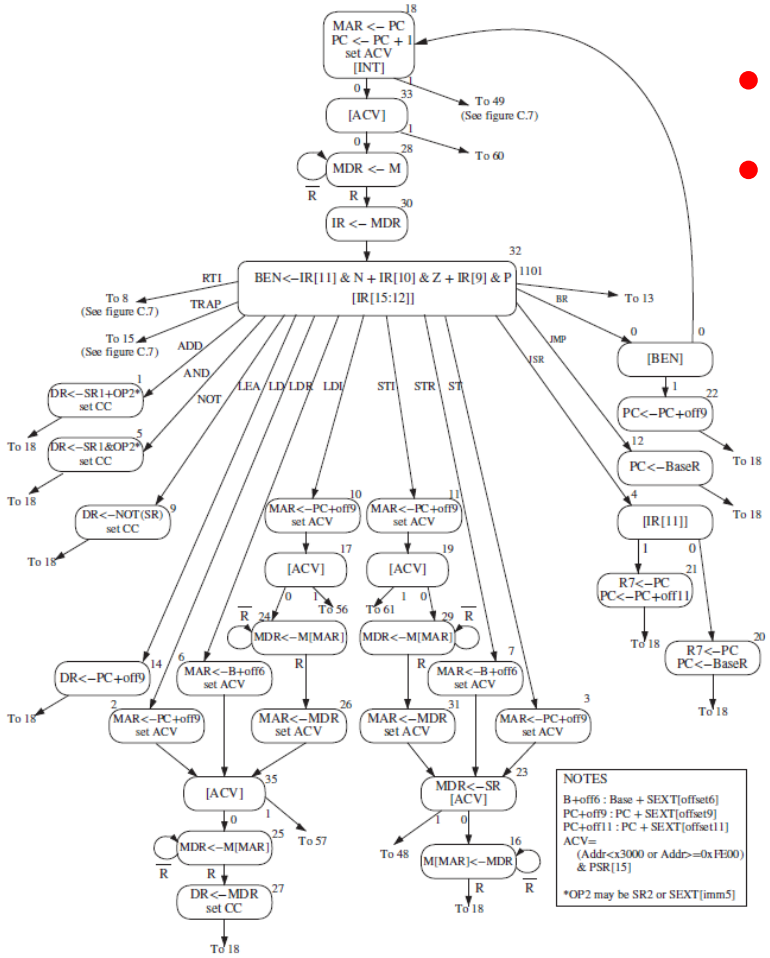


Figure C.2 A state machine for the LC-3.

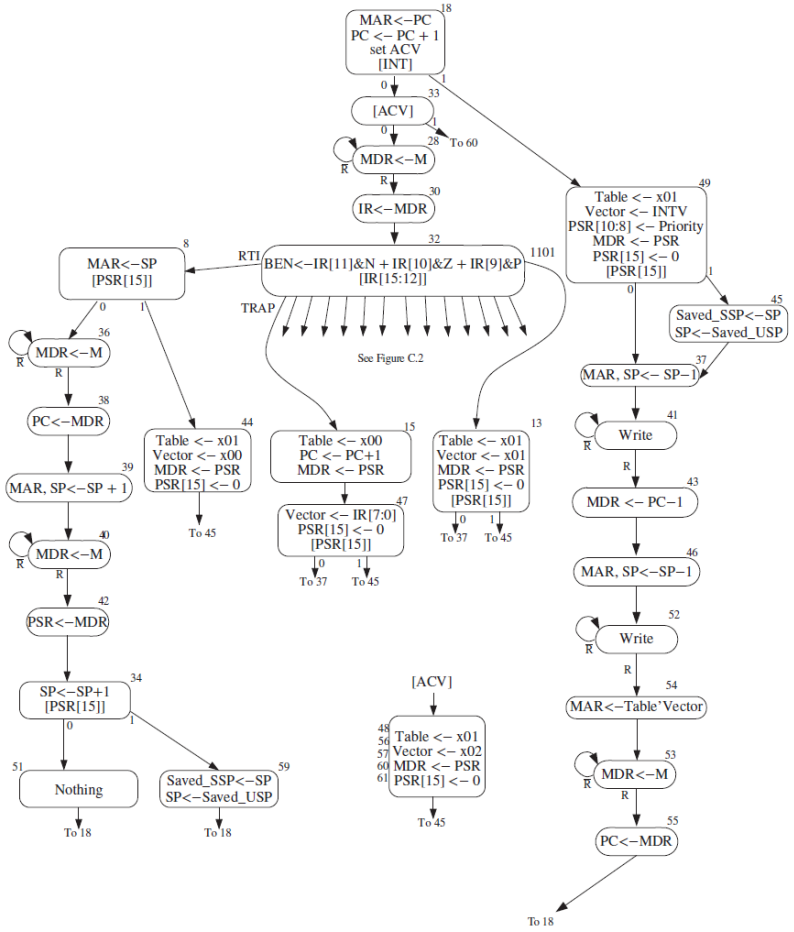
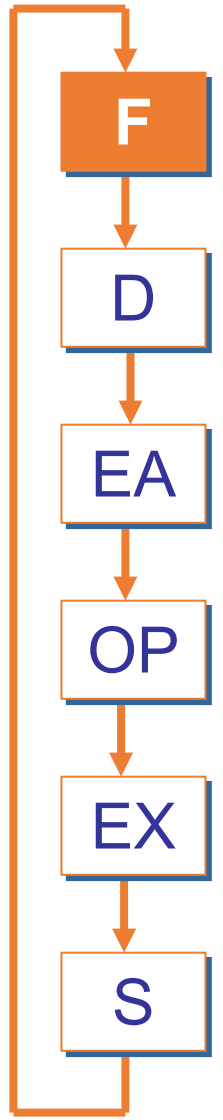


Figure C.7 LC-3 state machine showing interrupt control.



The state machine

- The state machine describes what happens during each **clock cycle** in which the computer is running.
- Each state is **active for exactly one clock cycle** before control passes to the next state.
- Each node in the state machine corresponds to the **activity** that the processor carries out during a single clock cycle.

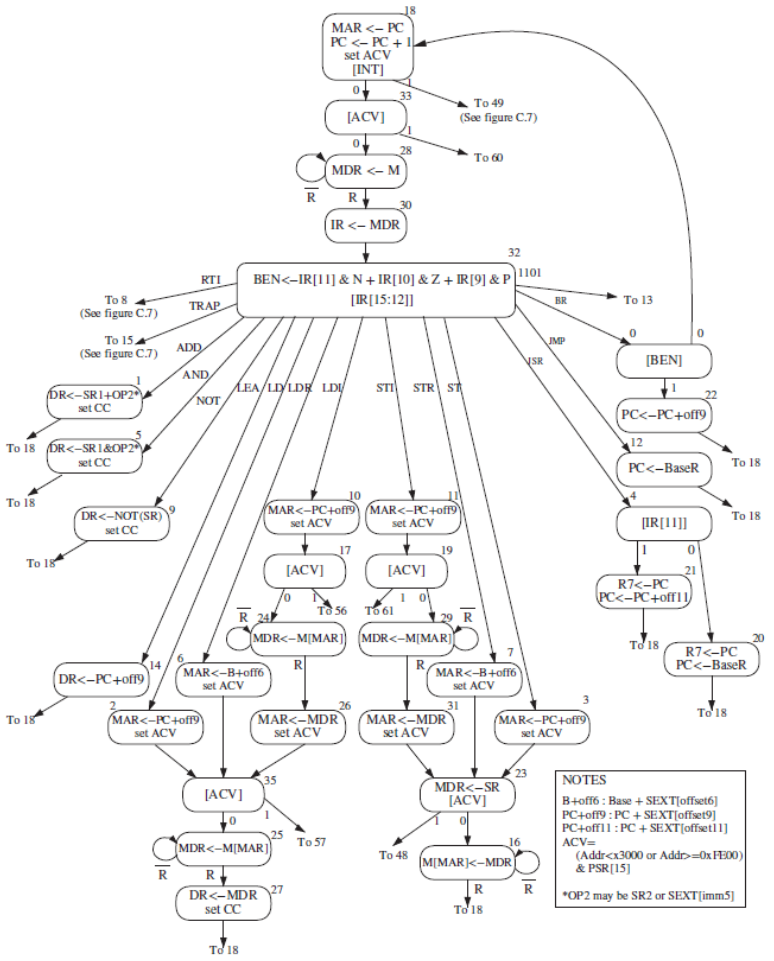


Figure C.2 A state machine for the LC-3.

The state machine

- the **FETCH** phase of the instruction cycle
  - In **state 18**, the **MAR** is loaded with the address contained in **PC**, and the PC is **incremented** in reparation for the FETCH of the next LC-3 instruction after the current instruction finishes its instruction cycle.
  - If the content of **MAR** specifies privileged memory, and PSR[15] = 1, indicating User mode, the access of the instruction will not be allowed. That would be an access control violation, so **ACV** is set.
  - Finally, if there is no interrupt request present (**INT** = 0), the flow passes to **state 33**. Or else, the flow passes to **state 49**.

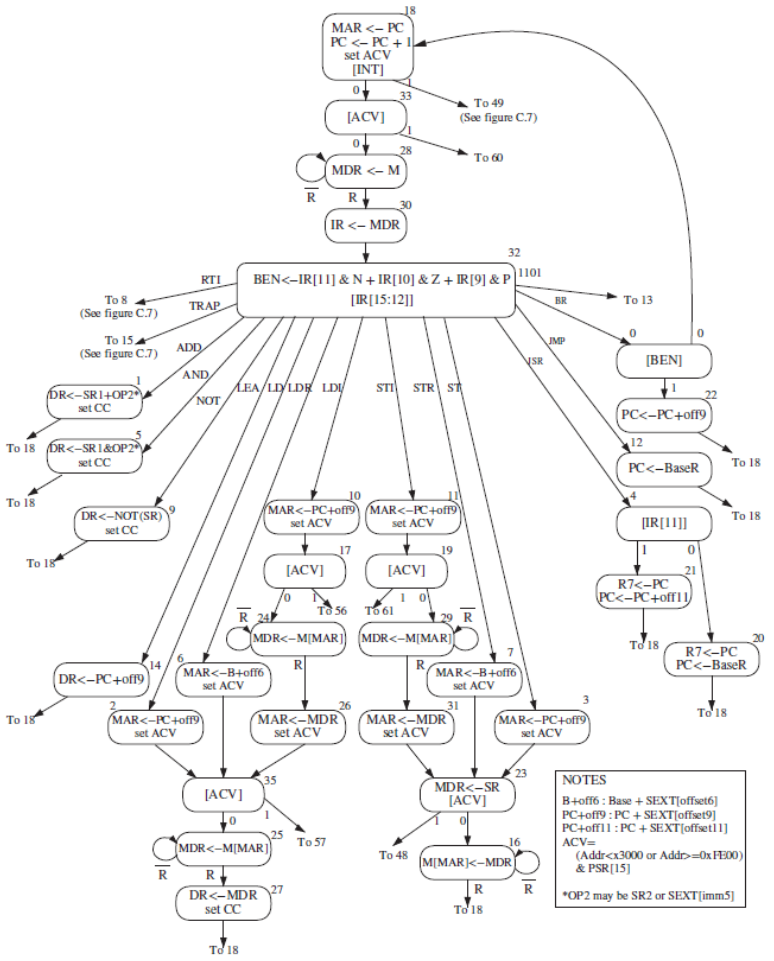


Figure C.2 A state machine for the LC-3.

The state machine

- the **FETCH** phase of the instruction cycle
  - From **state 33**, control passes to **state 60** if the processor is trying to access privileged memory while in User mode, or to **state 28**, if the memory access is allowed, that is, if there is no ACV violation.
  - In **state 28,** since the MAR contains the address of the instruction to be processed, this instruction is read from memory and loaded into the MDR. Since this memory access can **take multiple cycles**, this state continues to execute until a **ready signal** from the memory (**R**) is asserted, indicating that the memory access has completed. Thus, the MDR contains the valid contents of the memory location specified by MAR.
  - The state machine then moves on to **state 30**, where the instruction is loaded into the instruction register (**IR**), completing the **fetch** phase of the **instruction cycle**.

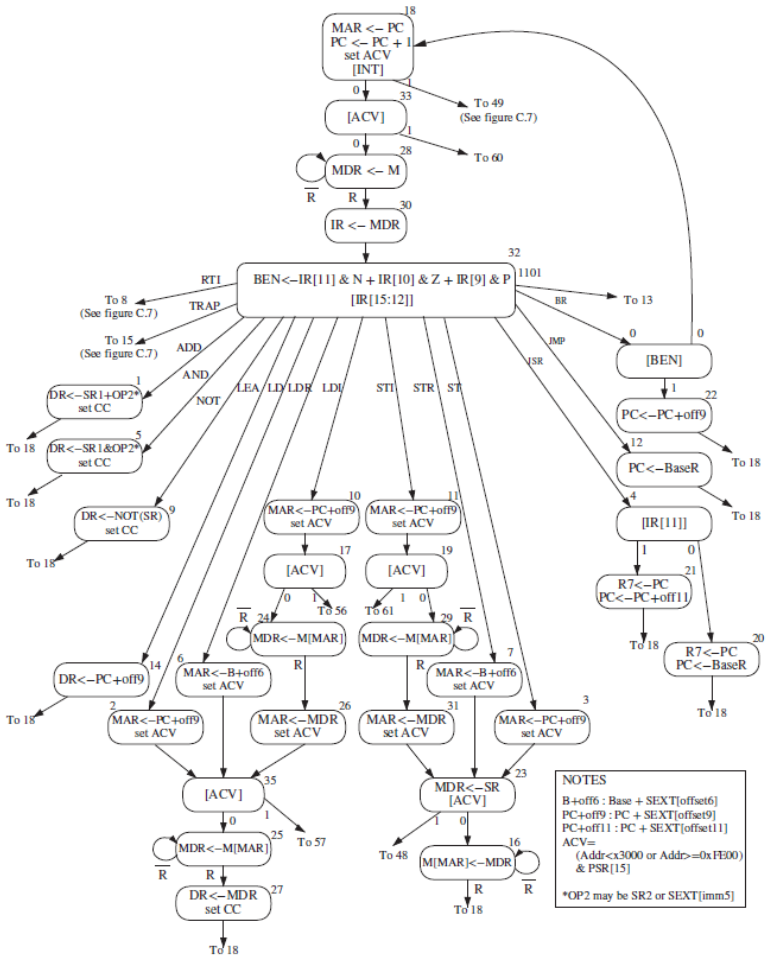


Figure C.2 A state machine for the LC-3.

The state machine

- The state machine then moves to **state 32**, where **DECODE** takes place.
  - there are 16 arcs emanating from state 32, each one corresponding to bits [15:12] of the LC-3 instruction.
  - the arc from **the last state** of each instruction cycle (i.e., the state that completes the processing of that LC-3 instruction) takes us to **state 18**

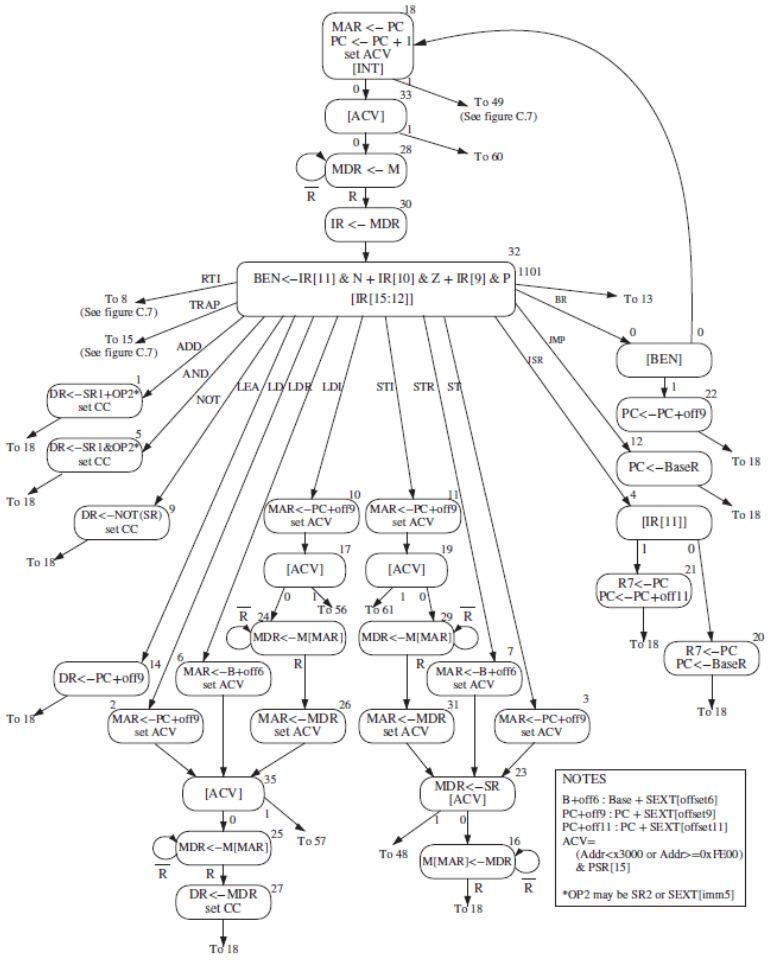


Figure C.2 A state machine for the LC-3.

The data path

- The **data path** consists of **all components** that actually **process the information** during each clock cycle—
  - the **functional units** that operate on the information,
  - the **registers** that store information at the end of one cycle so it will be available for further use in subsequent cycles,
  - and the **buses and wires** that carry information from one point to another in the data path.

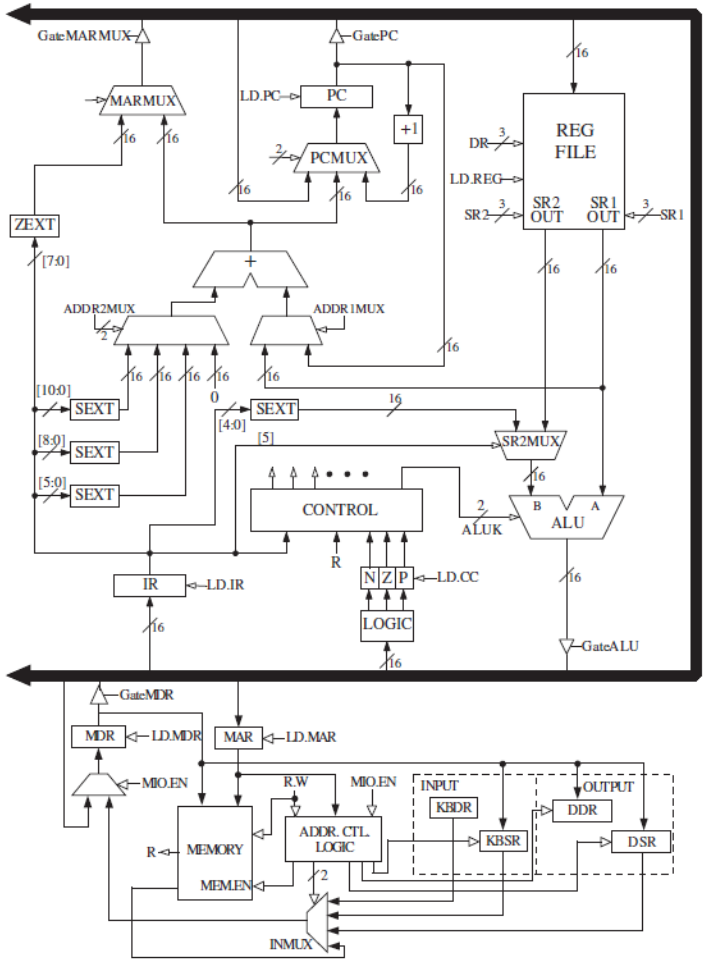


Figure C.3 The LC-3 data path.

Table C.1 Data Path Control Signals

Signal Name	Signal Values	
LD.MAR/1:	NO, LOAD	
LD.MDR/1:	NO, LOAD	
LD.IR/1:	NO, LOAD	
LD.BEN/1:	NO, LOAD	
LD.REG/1:	NO, LOAD	
LD.CC/1:	NO, LOAD	
LD.PC/1:	NO, LOAD	
LD.Priv/1:	NO, LOAD	
LD.Priority/1:	NO, LOAD	
LD.SavedSSP/1:	NO, LOAD	
LD.SavedUSP/1:	NO, LOAD	
LD.ACV/1:	NO, LOAD	
LD.Vector/1:	NO, LOAD	
GatePC/1:	NO, YES	
GateMDR/1:	NO, YES	
GateALU/1:	NO, YES	
GateMARMUX/1:	NO, YES	
GateVector/1:	NO, YES	
GatePC-1/1:	NO, YES	
GatePSR/1:	NO, YES	
GateSP/1:	NO, YES	
PCMUX/2:	PC+1	;select pc+1
	BUS	;select value from bus
	ADDER	;select output of address adder
DRMUX/2:	11.9	;destination IR[11:9]
	R7	;destination R7
	SP	;destination R6
SR1MUX/2:	11.9	;source IR[11:9]
	8.6	;source IR[8:6]
	SP	;source R6
ADDR1MUX/1:	PC, BaseR	
ADDR2MUX/2:	ZERO	;select the value zero
	offset6	;select SEXT[IR[5:0]]
	PCoffset9	;select SEXT[IR[8:0]]
	PCoffset11	;select SEXT[IR[10:0]]
SPMUX/2:	SP+1	;select stack pointer+1
	SP–1	;select stack pointer–1
	Saved SSP	;select saved Supervisor Stack Pointer
	Saved USP	;select saved User Stack Pointer
MARMUX/1:	7.0	;select ZEXT[IR[7:0]]
	ADDER	;select output of address adder
TableMUX/1:	x00, x01	
VectorMUX/2:	INTV	
	Priv.exception	
	Opc.exception	
	ACV.exception	
PSRMUX/1:	Individual settings, BUS	
ALUK/2:	ADD, AND, NOT, PASSA	
MIO.EN/1:	NO, YES	
R.W/1:	RD, WR	
Set.Priv/1:	0	;Supervisor mode
	1	;User mode

The data path

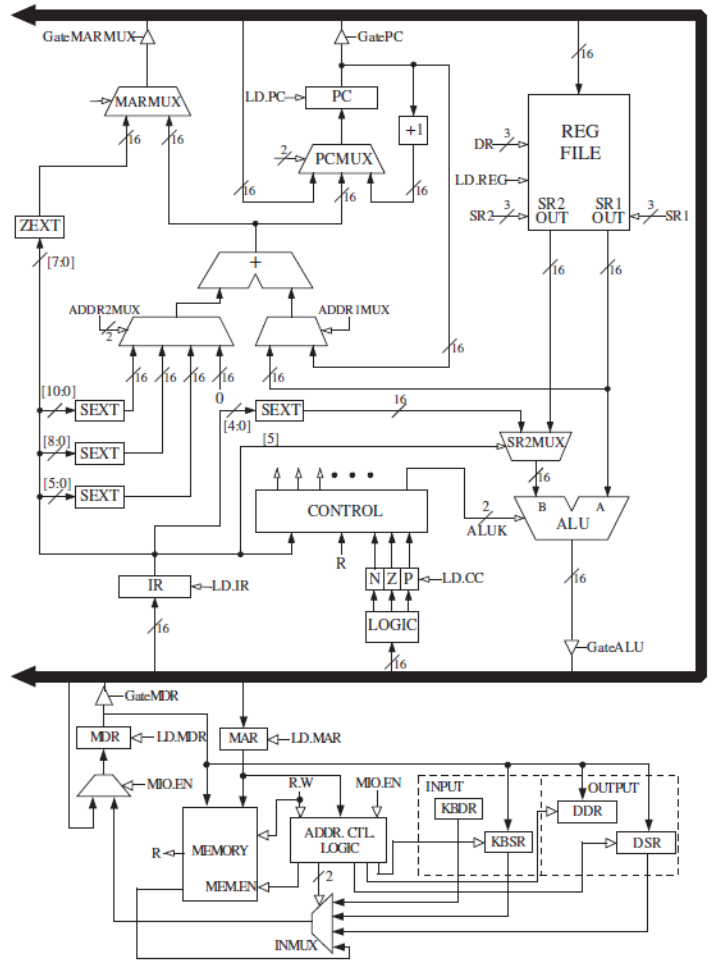


Figure C.3 The LC-3 data path.

- **LD.PC** (p134)
  - In order for the PC to be, the finite state machine must assert the PCMUX select lines to choose the output of the box labeled +1 and must also **assert** the LD.PC signal to **load the output of the PCMUX into** the PC at the end of the current cycle.
- **ALUK**
  - ALUK consists of two bits, it can have one of four values. Which value it has during any particular clock cycle depends on whether the ALU is required to ADD, AND, NOT, or simply pass one of its inputs to the output during that clock cycle (PASSA).

Additional logic required to provide control signals

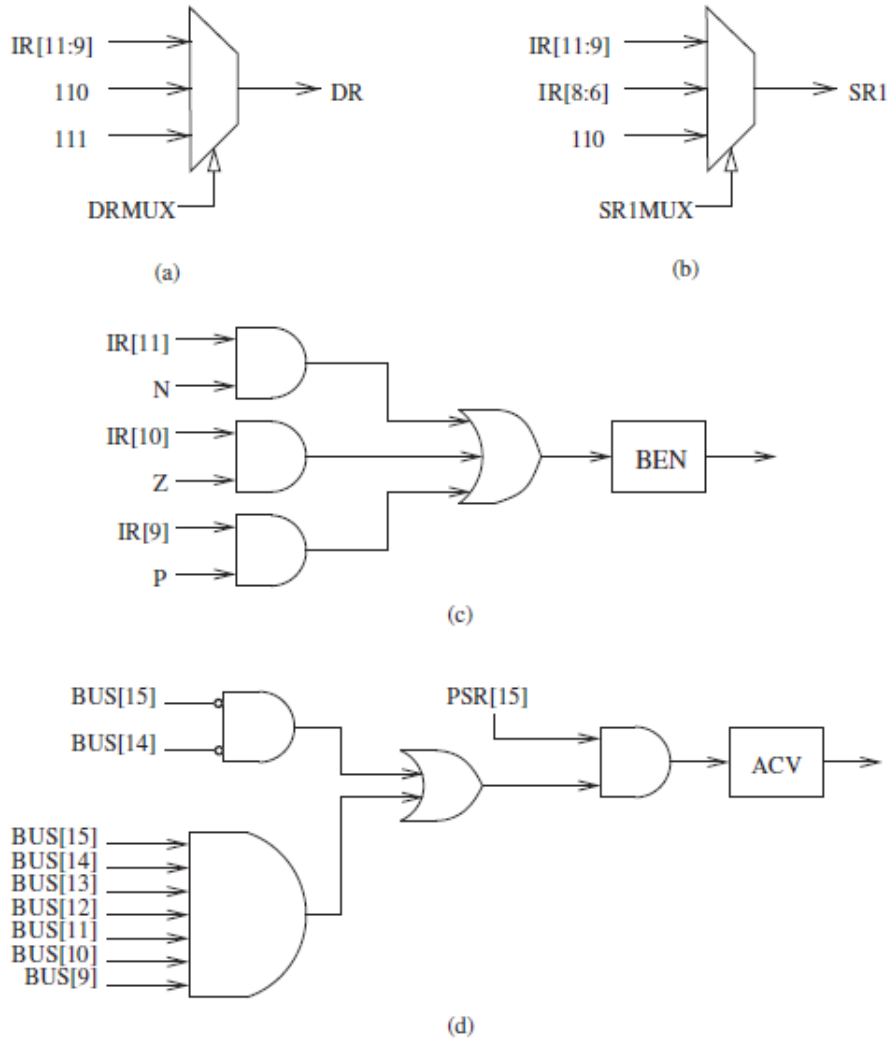
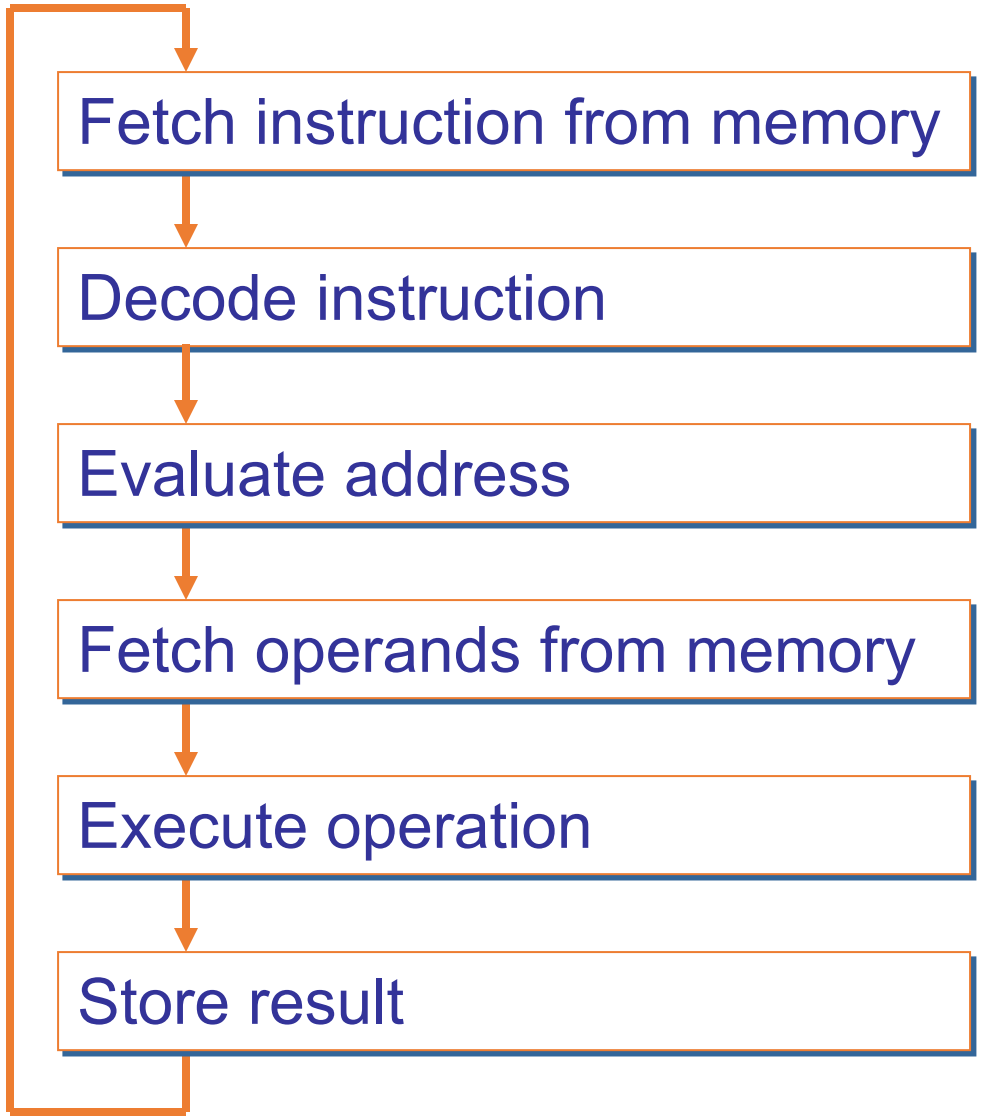
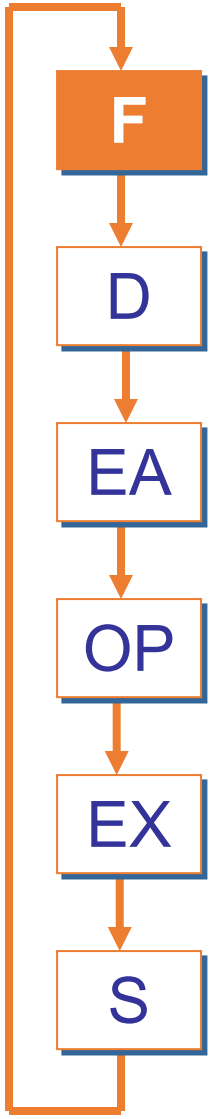


Figure C.6 Additional logic required to provide control signals.

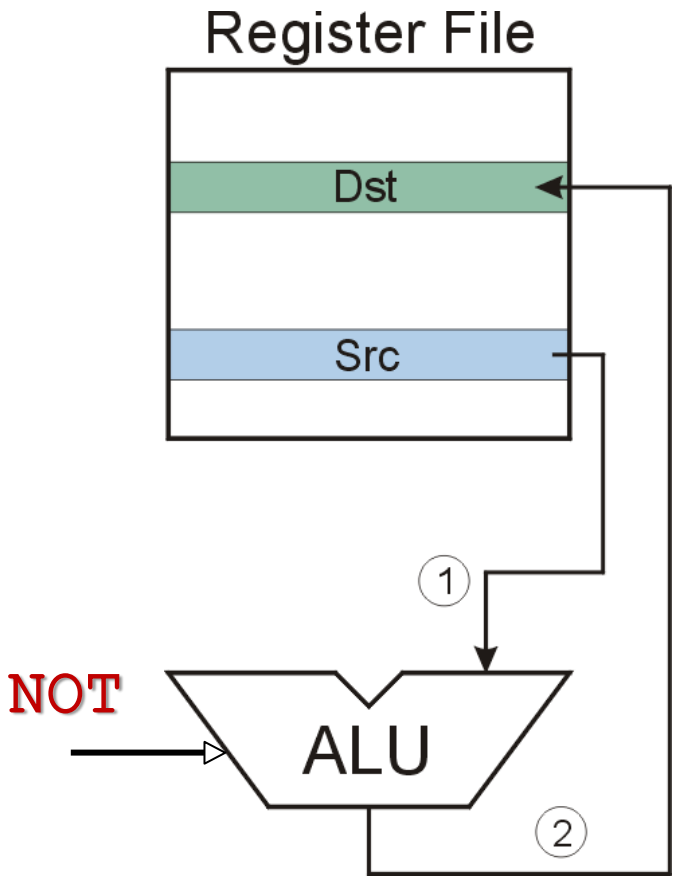
Instruction Cycle (chapter 4.3 & chapter 5.6)

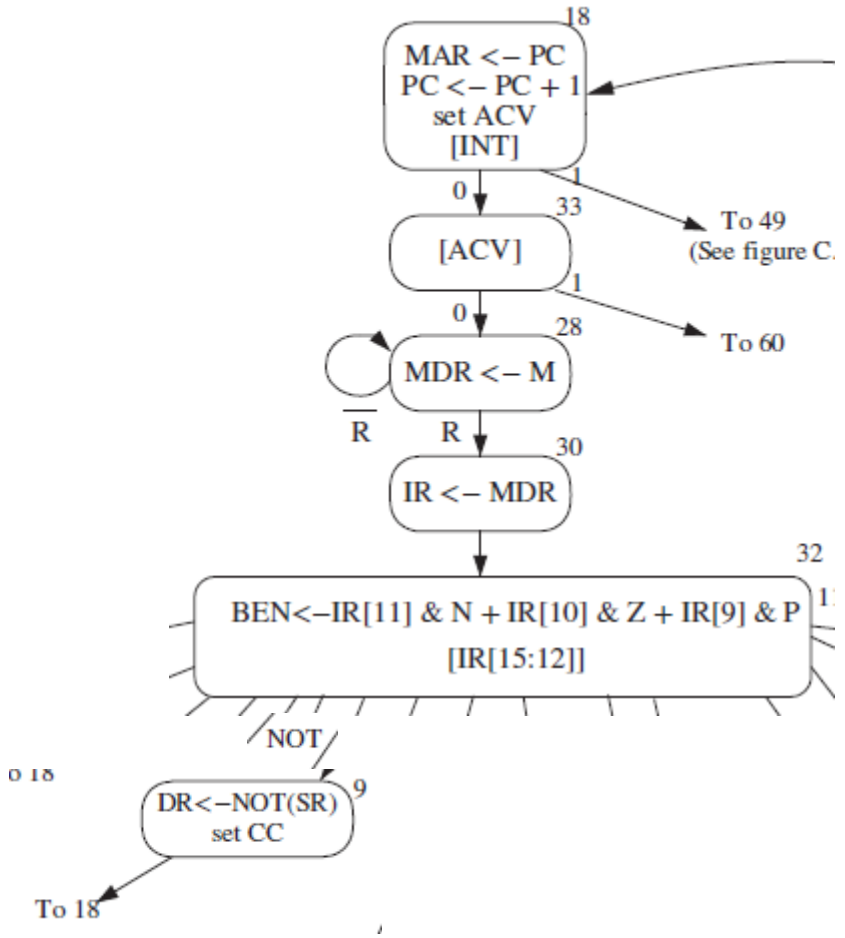
-----including 6 phases, each phase requiring 0 or more steps.



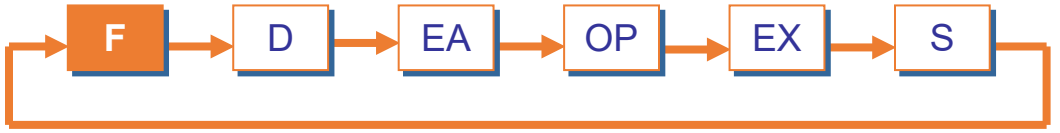


NOT (Register)





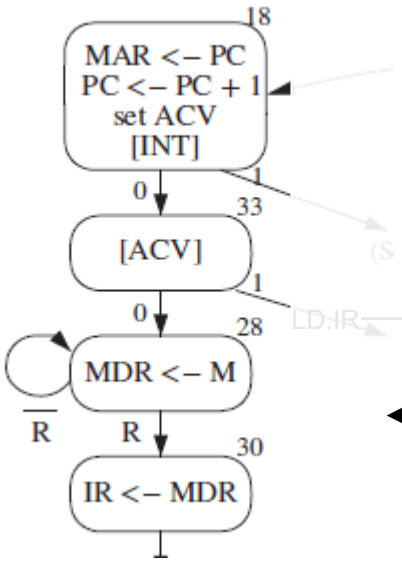
NOT (Register):

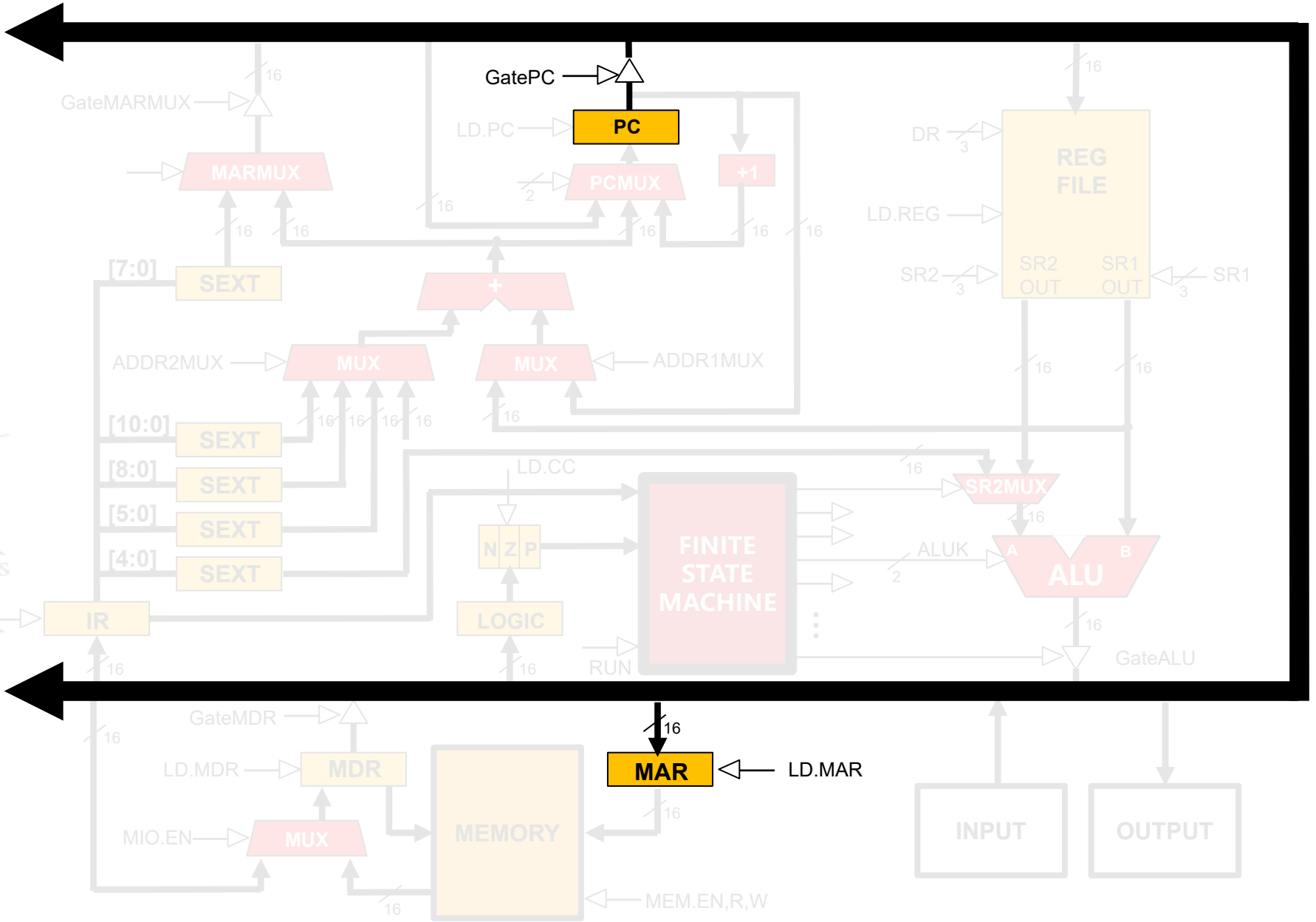


State 18

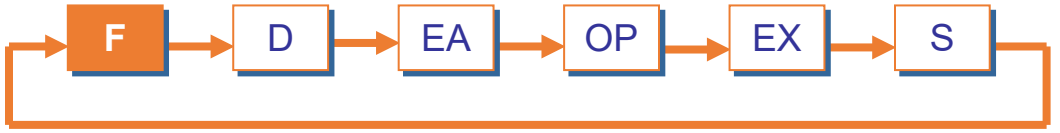
GatePC = YES

LD.MAR = LOAD





NOT (Register):



此处与教材不一致

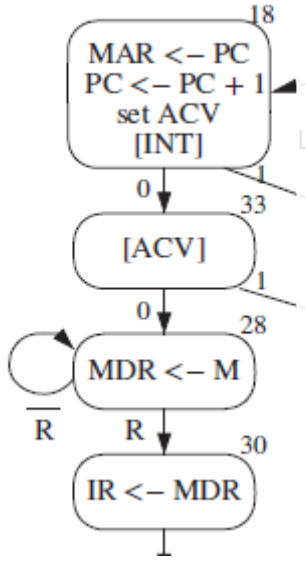
State 28

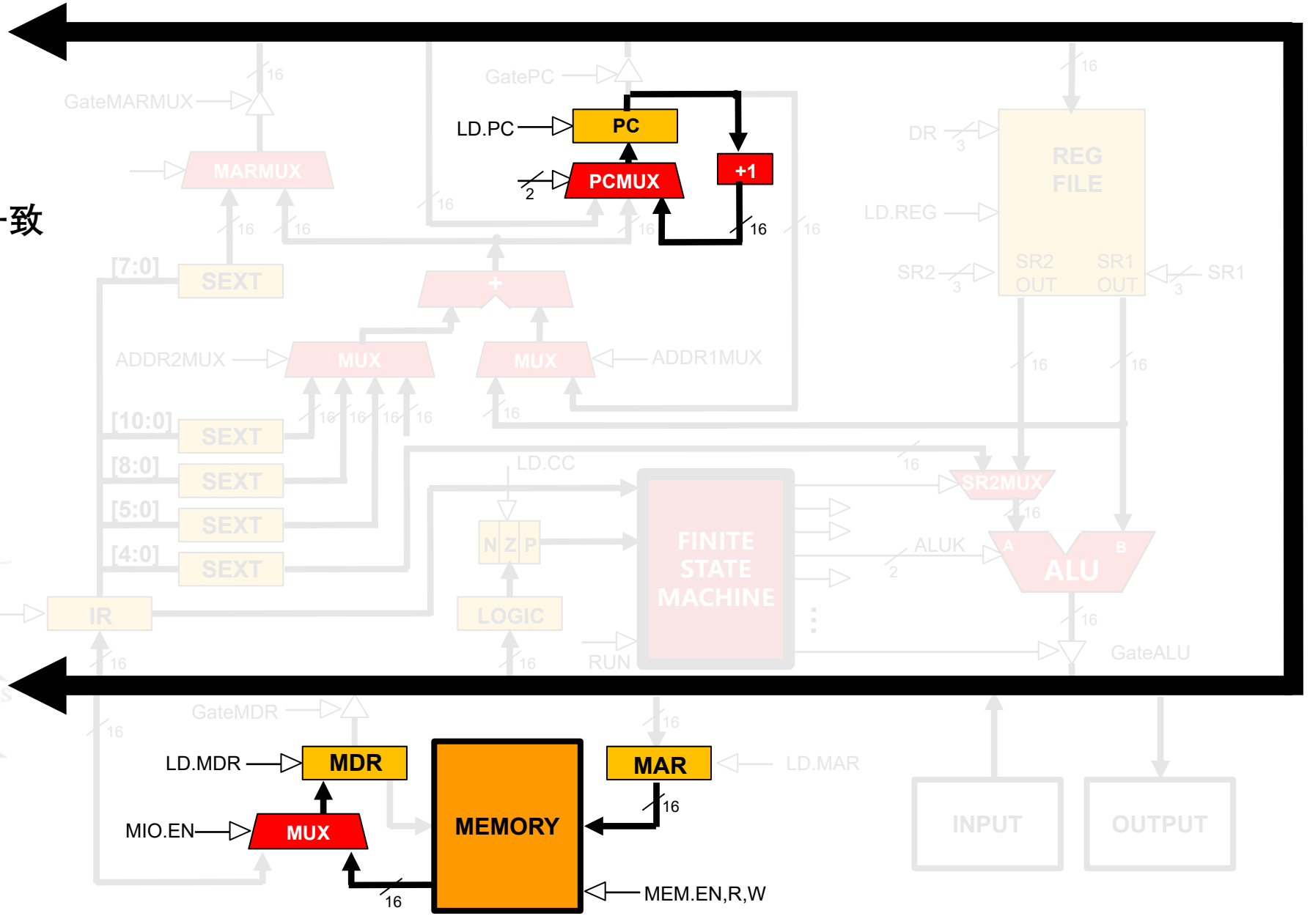
LD.PC = LOAD

LD.MDR = LOAD

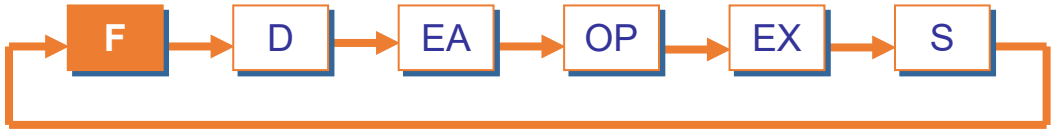
MIO.EN = YES

PCMUX = PC + 1

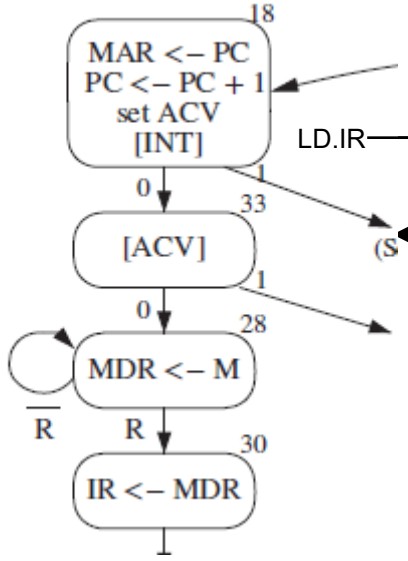




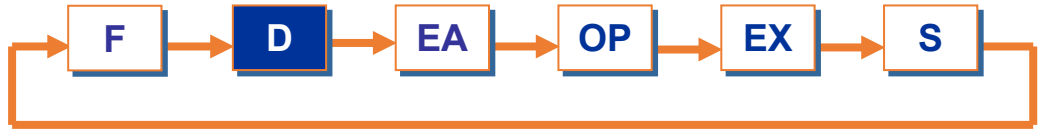
NOT (Register):



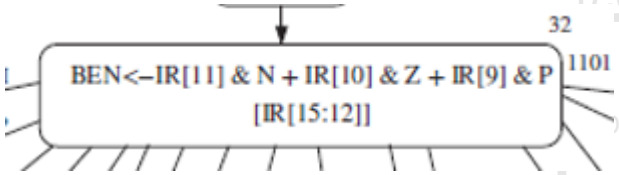
State 30
LD.IR = LOAD
GateMDR = YES

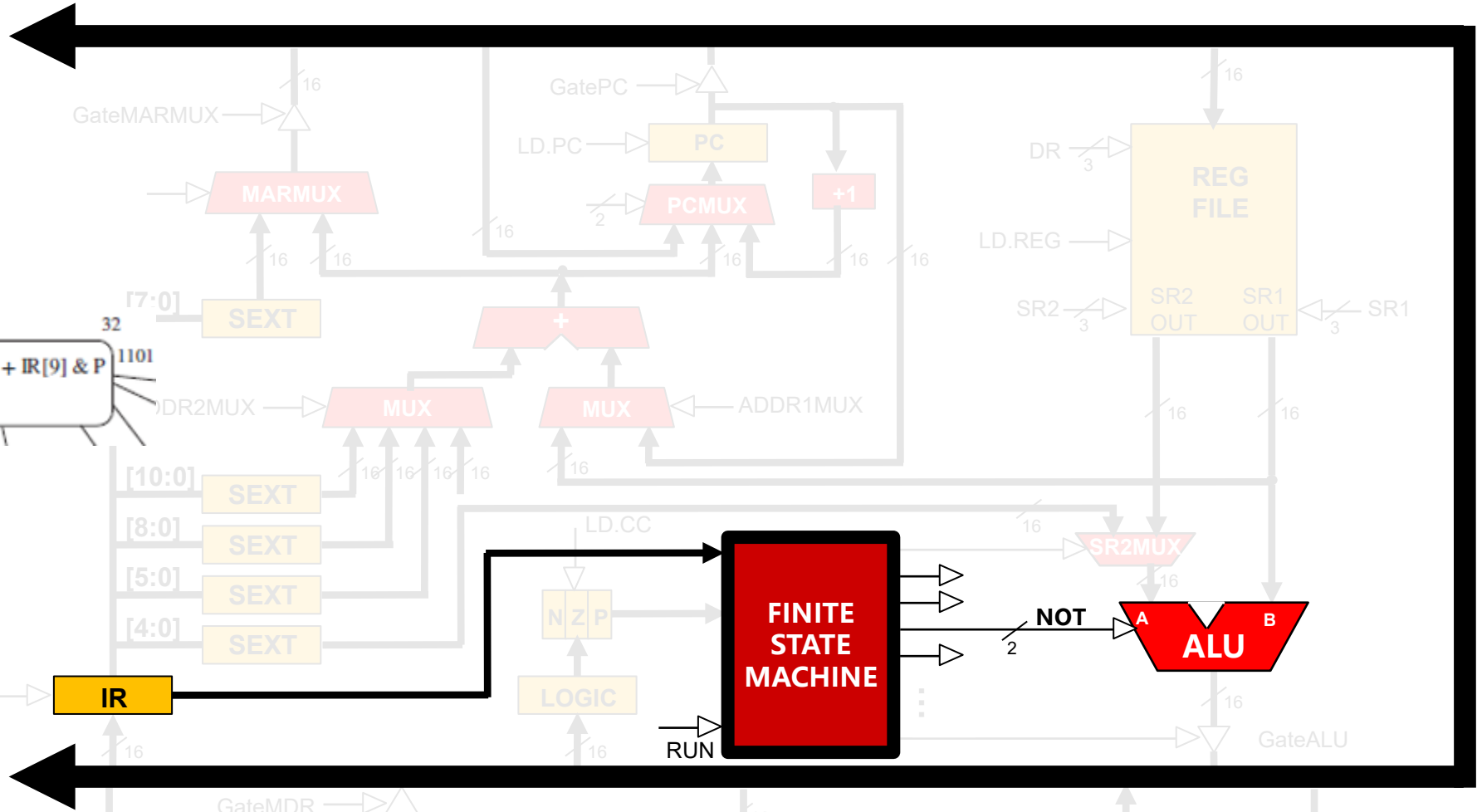


NOT (Register):

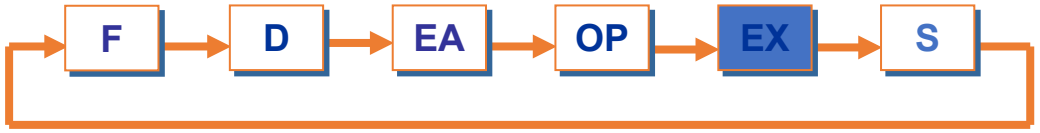


State 32
ALUK = NOT





NOT (Register):



参考p131-132

State machine省略了OP and S phases

State 9

SR1MUX =IR[8:6]

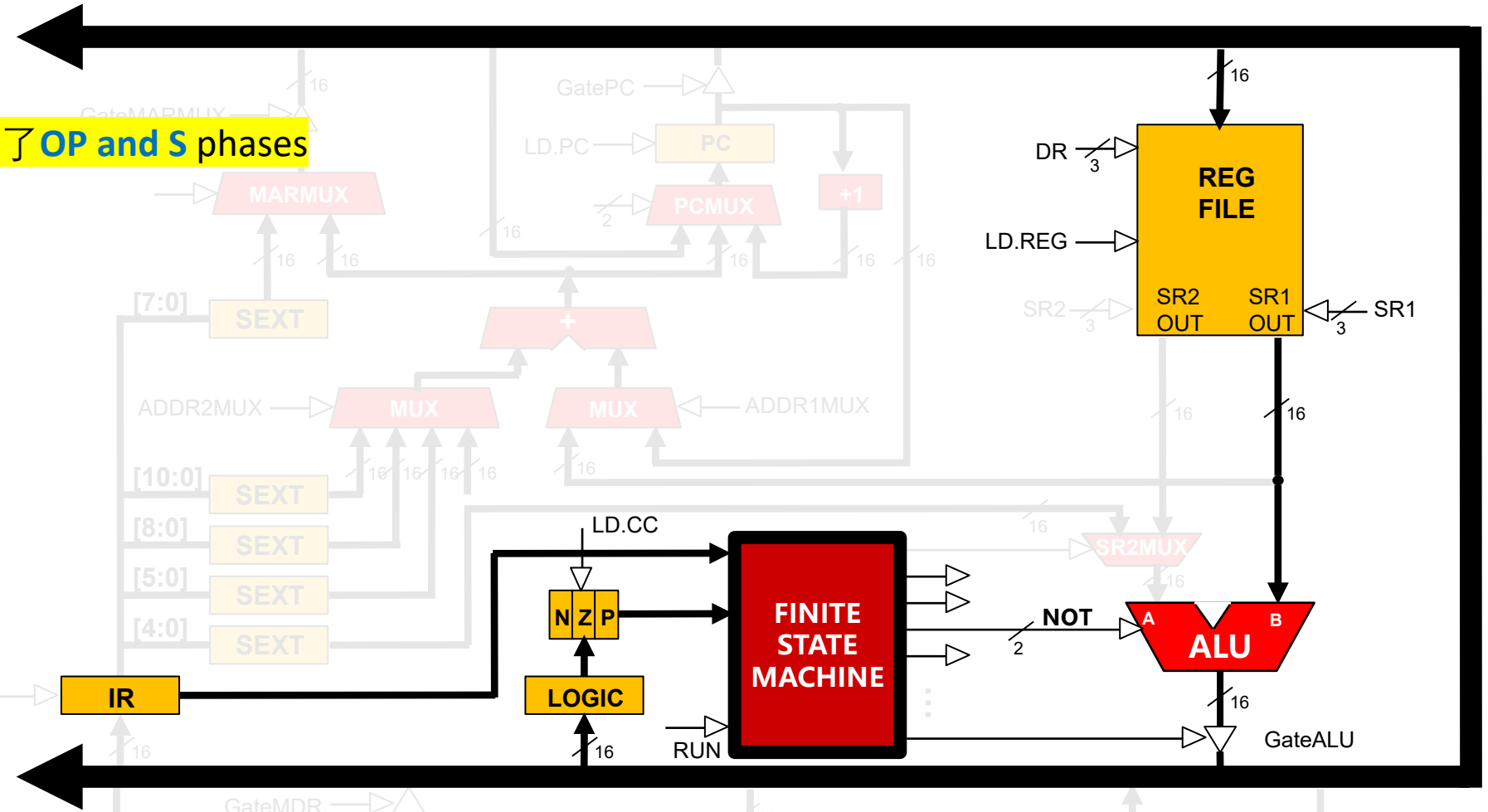
ALUK = NOT

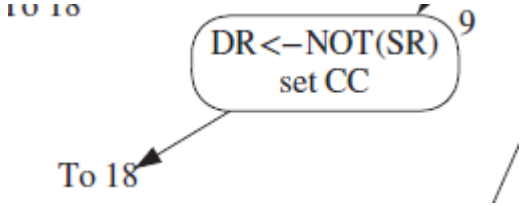
LD.REG=LOAD

GateALU = YES

DRMUX = IR[11:9]

LD.CC = LOAD





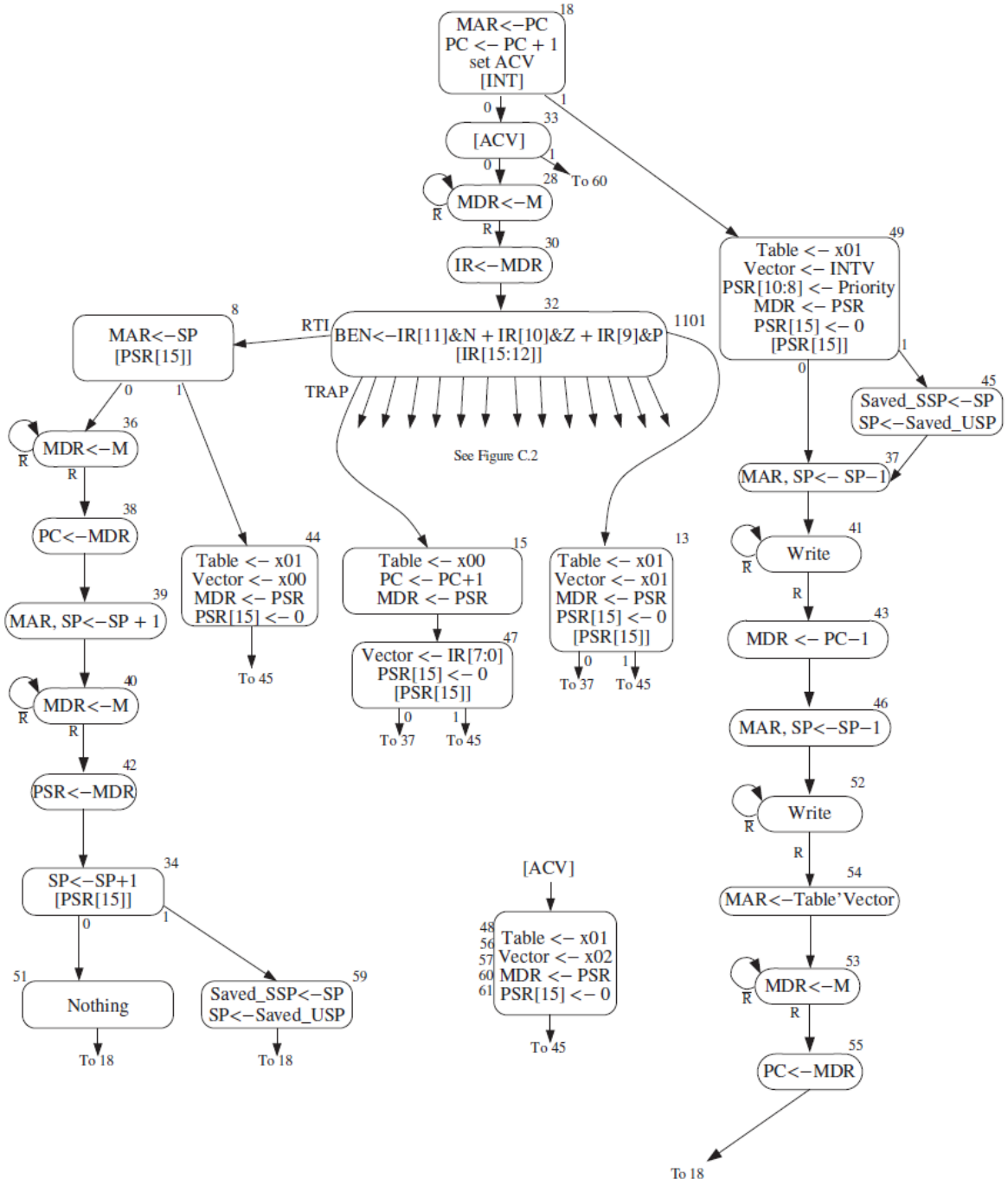
Interrupt and RTI

- **Interrupt**
  - State 18 is the only state in which the processor checks for interrupts
  - State 45, switch stack
  - State 41, write PSR
  - State 52, write PC
  - State 53, read new instruction

Table‘Vector: concatenating Table and Vector

Table : 0x00, trap

0x01, interrupt



TRAP and RTI

- RTI
  - State 8, exception detection
  - State 8, 36, 38, restore PC
  - State 39, 40, 42, restore PSR
  - State 59, switch stack