Chapter 4

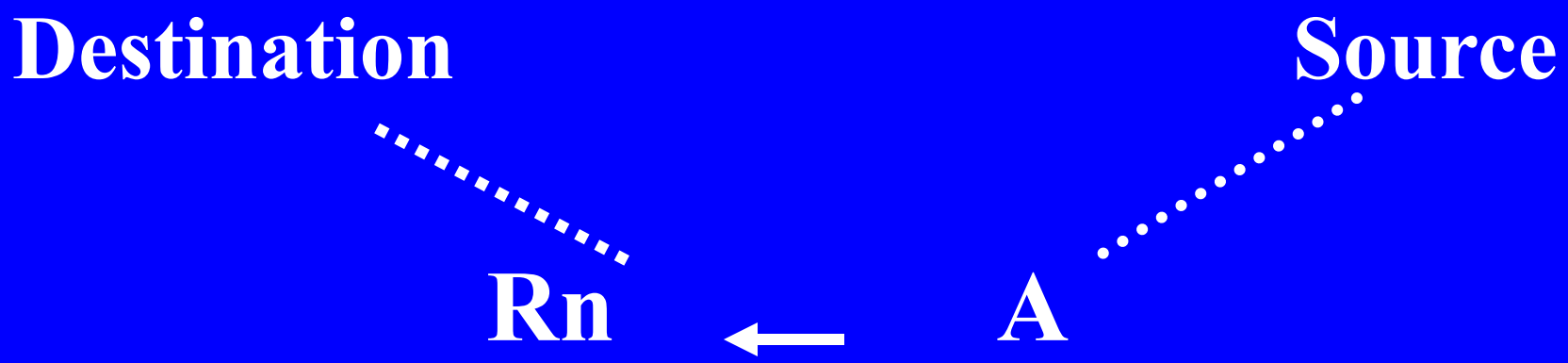
8051 Family Microcontrollers Instruction Set

Lesson 3

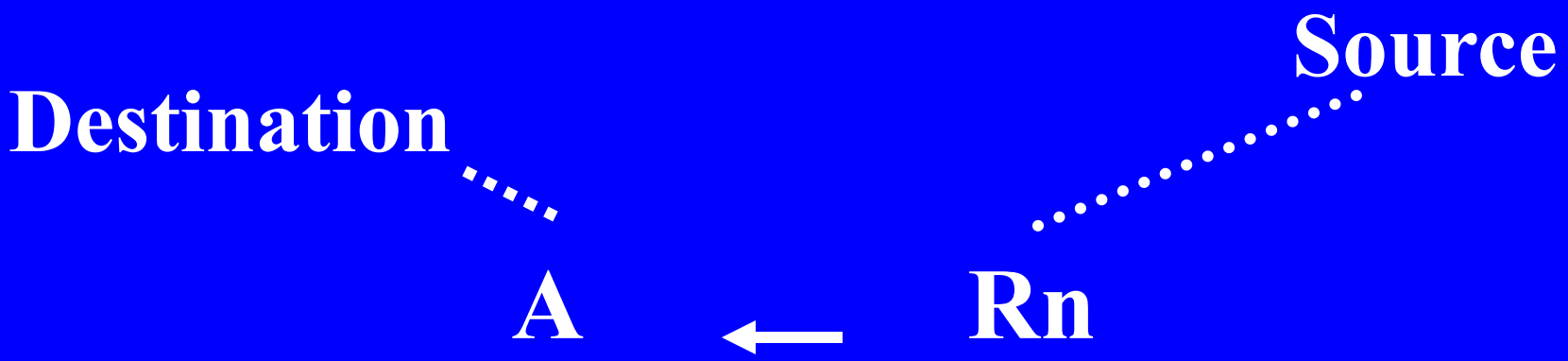
Data Transfer Instructions

Move byte between accumulator (an SFR) and register at a register bank

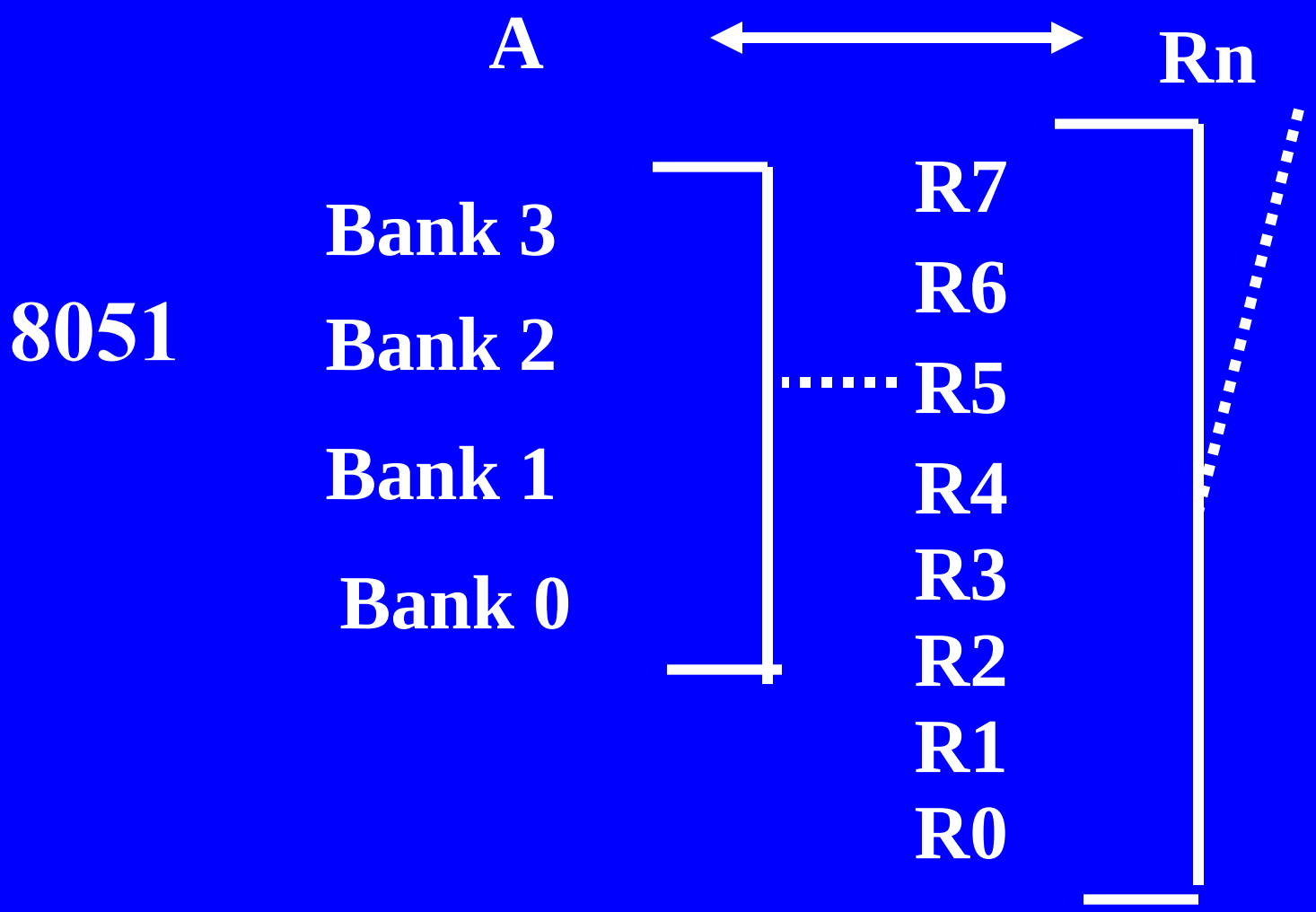
MOV Rn, A



MOV A, Rn



MOV A, Rn; MOV Rn, A



Rn

Rn: n = 0,1, 2, 3,4, 5, 6,or 7,

n is as per 3 bits coexisting with 5-bits with opcode

Register n is at the bank defined by RS1 and RS0 at PSW

- 00 means Bank 0, -01 bank 1
- 10 bank 2- 11 bank 3

Instruction Execution

1 clock cycle

STEP 1 **Fetch opcode-bits**

STEP 2 **Fetch bits for getting the operand(s)**

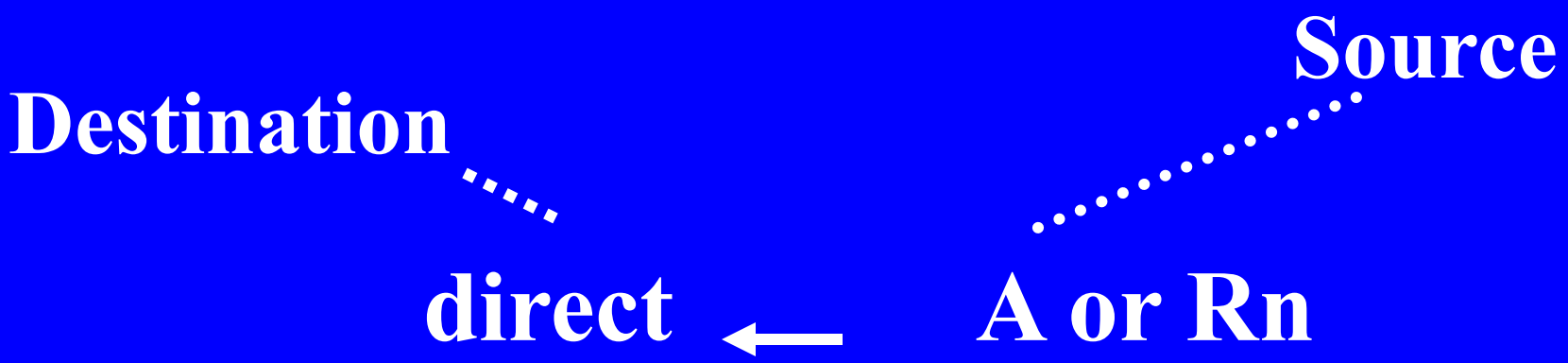
Register addressing mode

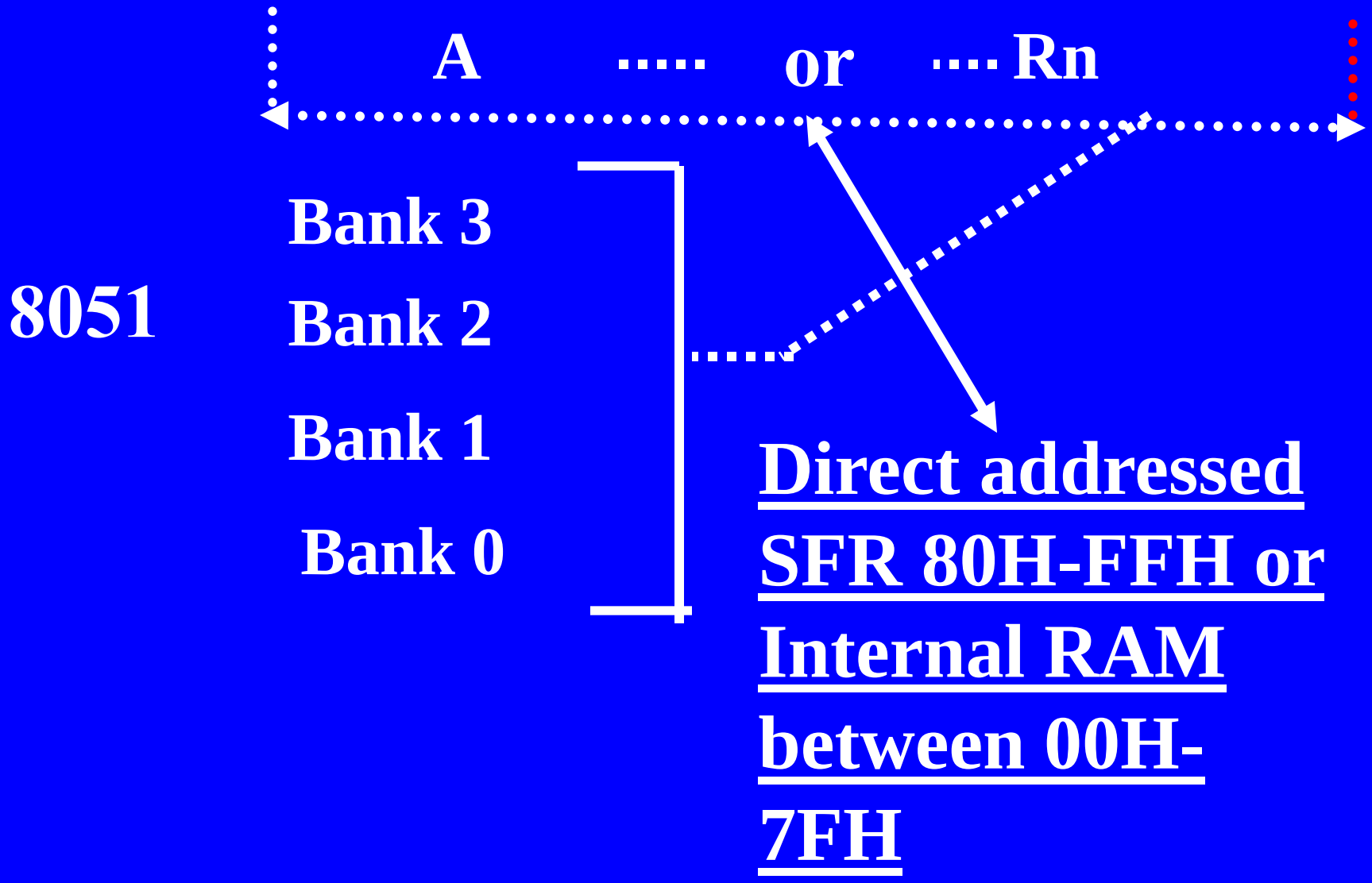
Time

Move byte from an SFR/Internal RAM to another *direct*

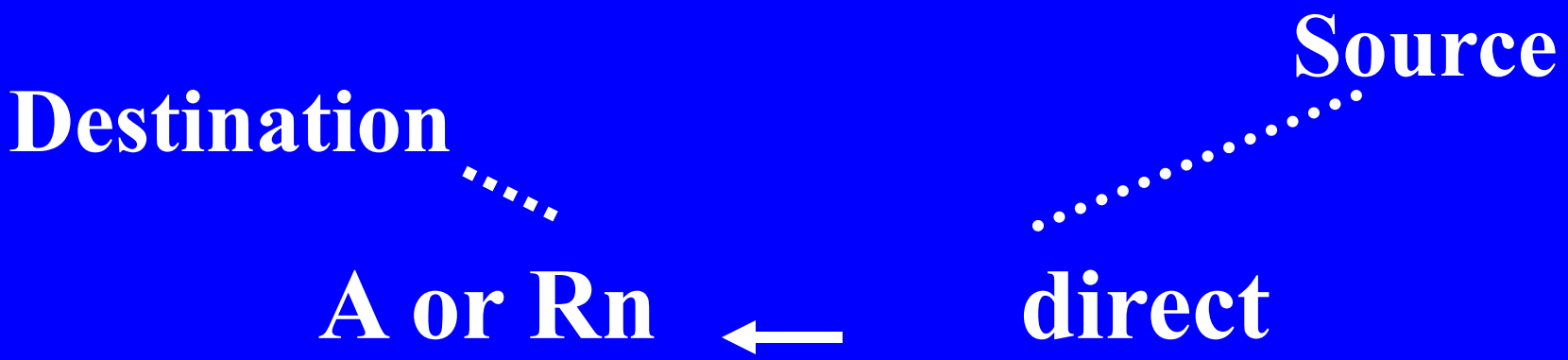
MOV A, direct; MOV Rn, direct; MOV direct, A; MOV direct, Rn

An SFR or Internal RAM between 00H-7FH




A
or
Rn
8051
Bank 3
Bank 2
Bank 1
Bank 0
Direct addressed SFR 80H-FFH or Internal RAM between 00H-7FH

An SFR or Internal RAM between 00H-7FH



Instruction Execution

1 clock cycle

STEP 1 **Fetch opcode-bits**

STEP 2 **Fetch direct**

direct addressing mode

Time

MOV direct, direct

MOV direct, direct

direct ⟷ direct

8051

Direct addressed SFR or Internal RAM between 00H-7FH

Direct addressed SFR or Internal RAM between 00H-7FH

Instruction Execution

2 clock cycles

STEP 1 **Fetch opcode-bits**

STEP 2 **Fetch direct**

STEP 3 **Fetch direct**

Time

both the operands using direct addressing modes

Move *indirect*

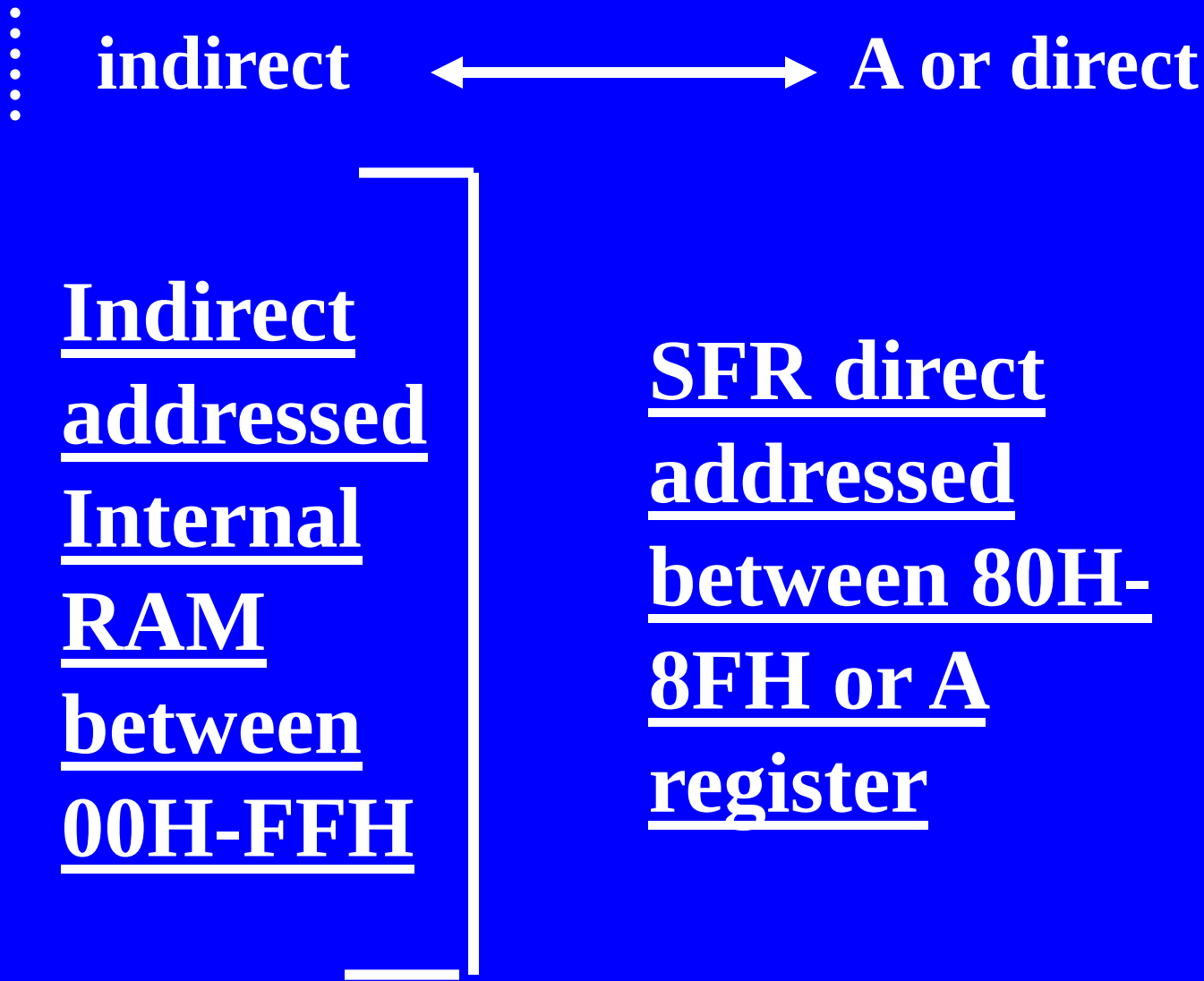
MOV A, @Ri;
MOV @Ri, A;
MOV direct, @Ri;
MOV @Ri, direct

An *indirect* addressed Internal RAM between 00H-FFH

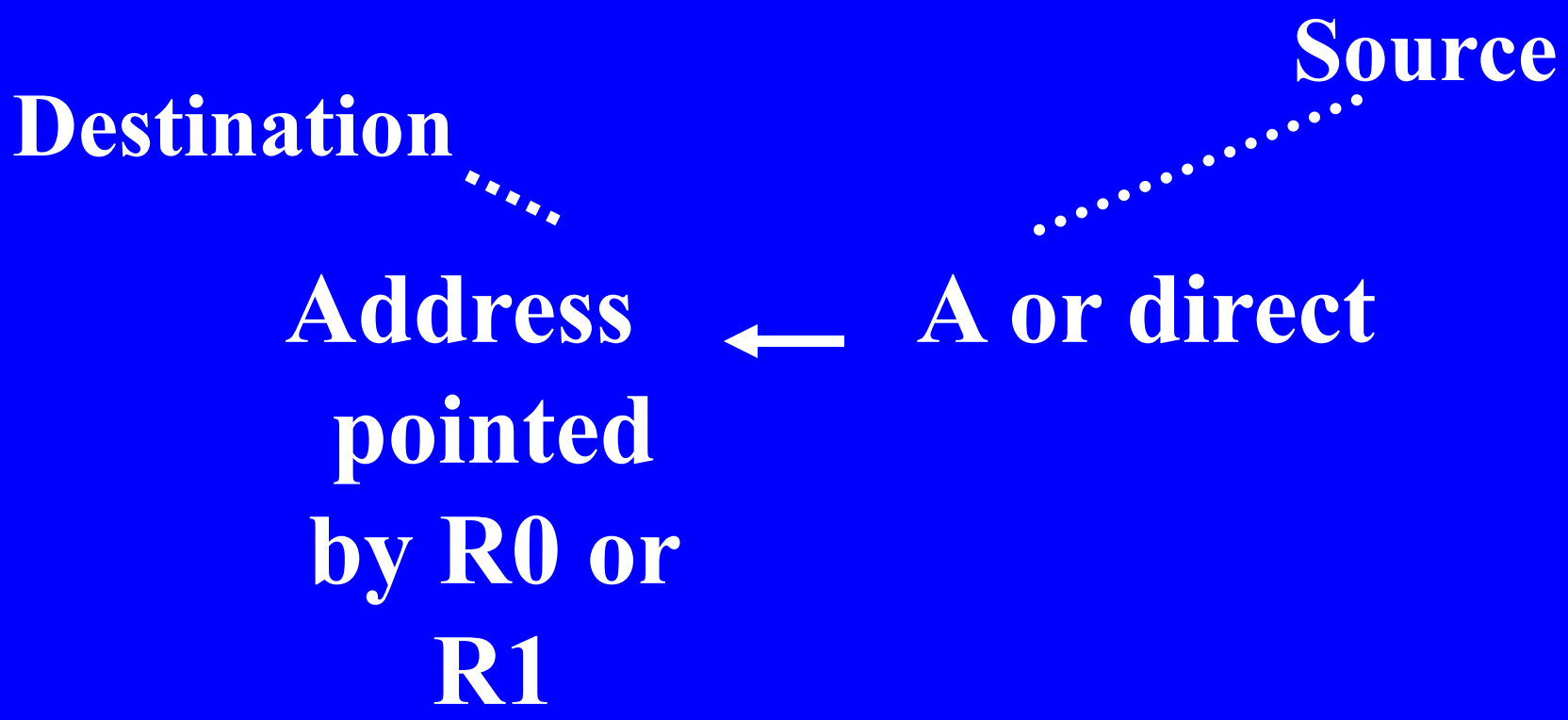
Source

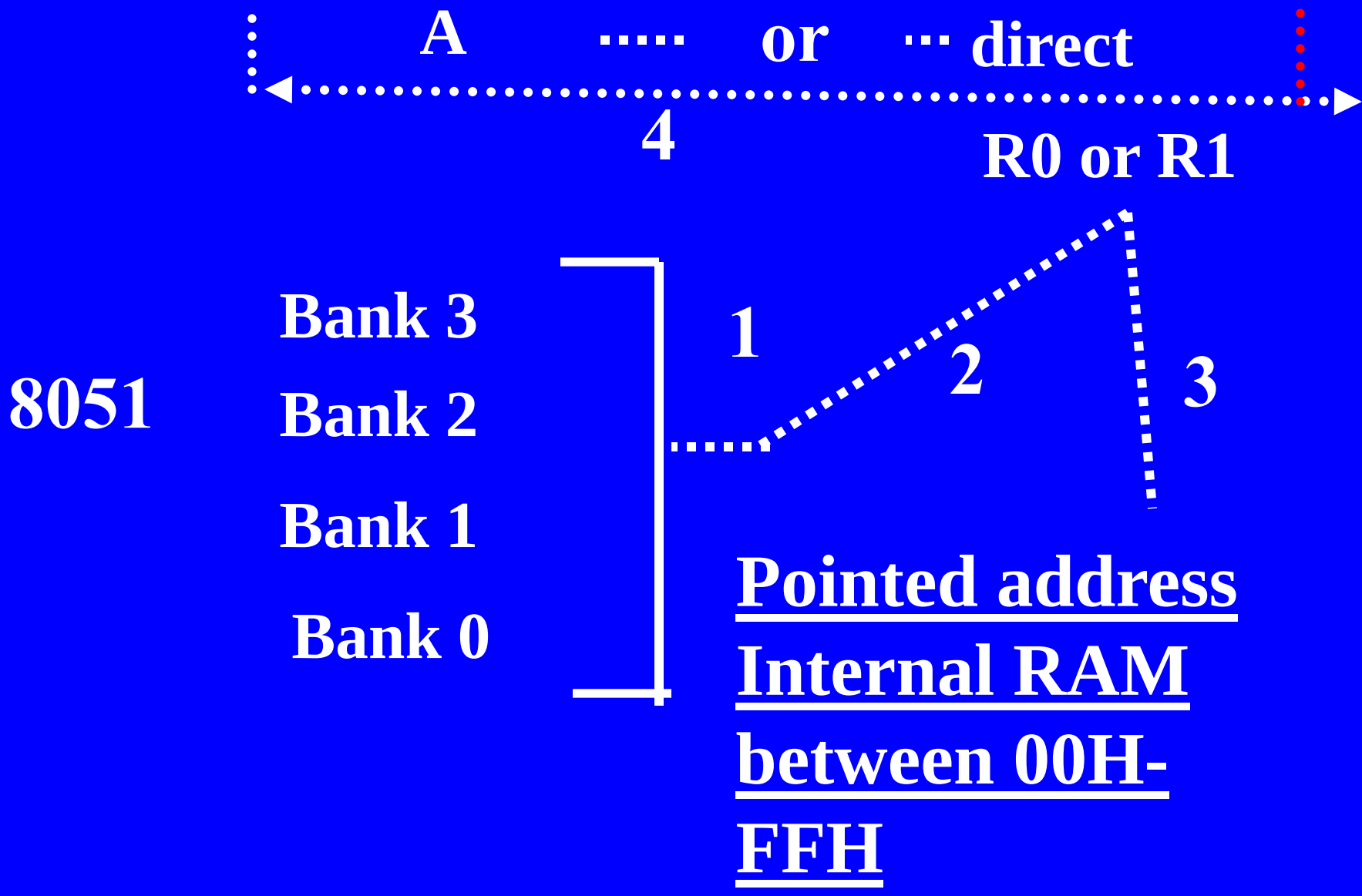
Destination

A or direct ← Address pointed by R0 or R1


indirect
A or direct
8051
Indirect addressed Internal RAM between 00H-FFH
SFR direct addressed between 80H-8FH or A register

An *indirect* Internal RAM between 00H-FFH




A
or
direct
4
R0 or R1
8051
Bank 3
Bank 2
Bank 1
Bank 0
1
2
3
Pointed address Internal RAM between 00H-FFH

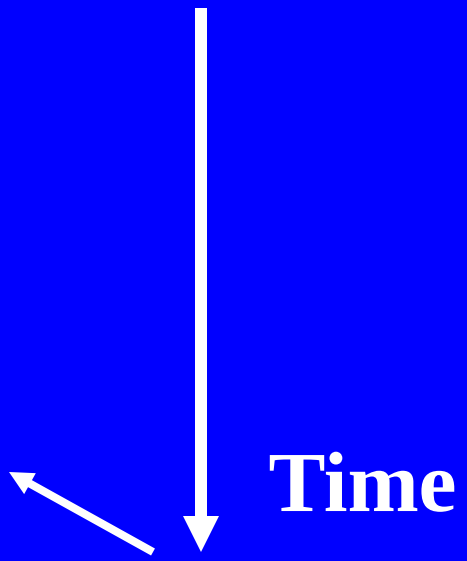
Instruction Execution

1 or 2 clock cycle(s)

STEP 1 **Fetch opcode-bits**

STEP 2 **Fetch bits for A or direct**

STEP 3 **Fetch bit for i 0 or 1**



indirect addressing mode one operand

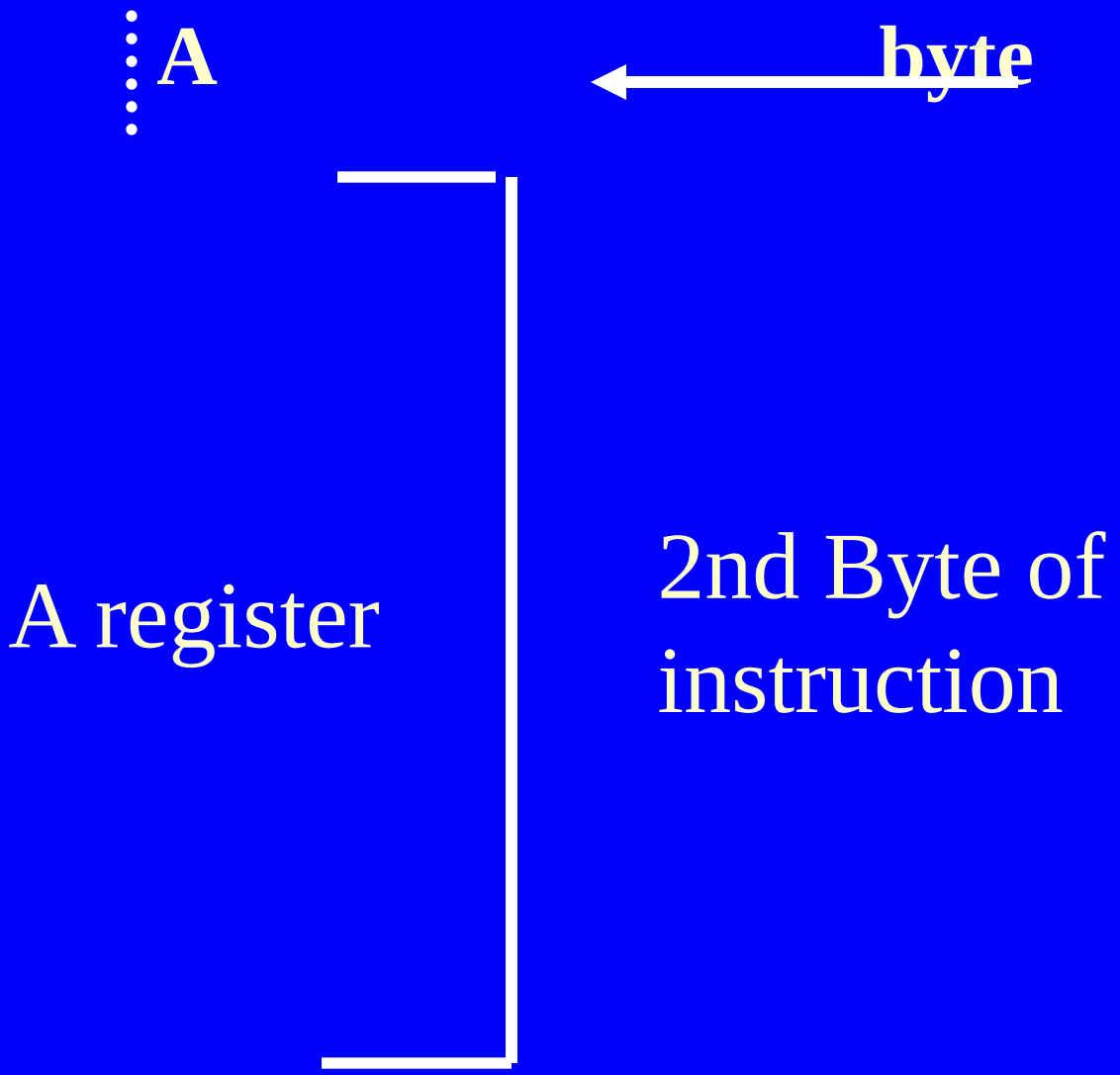
2 cycles when direct, 1 when A other operand

Instruction Execution

1. No indirect addressing in 8051 instruction for an SFR

2. Only indirect addressing in 8052 instruction Internal RAM between 80H-FFH

Move *immediate*, MOV immediate DPTR


A
byte
A register
2nd Byte of instruction

MOV A,#data Example

1 clock cycle

Time

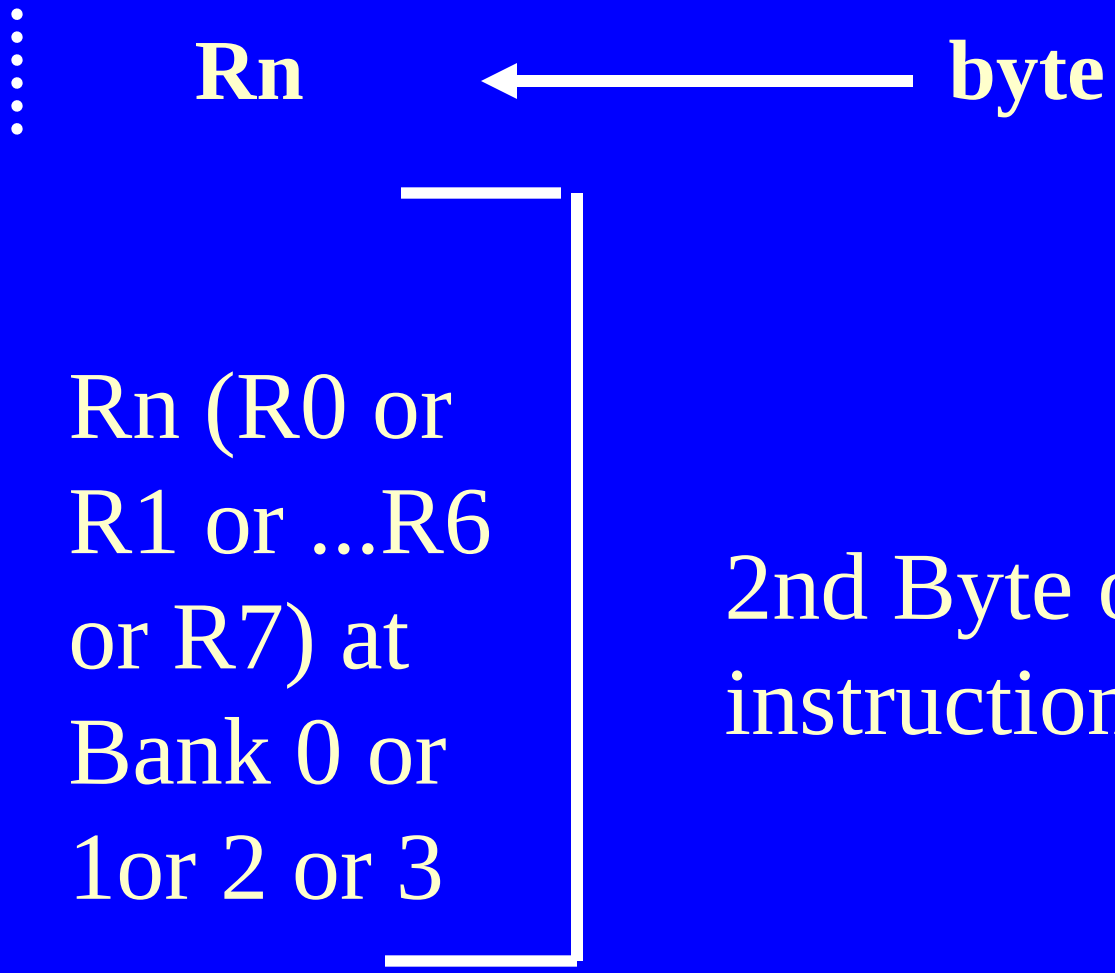
For moving 08H into A register

Fetch opcode and then byte for getting the operand for move

MOV A, #08H

Sign # specifies that 08H is the immediate succeeding byte is the operand

Codes in Memory-
74H, 08H


Rn
byte
8051
Rn (R0 or R1 or ...R6 or R7) at Bank 0 or 1or 2 or 3
2nd Byte of instruction

MOV Rn,#data

1 clock cycle

Time

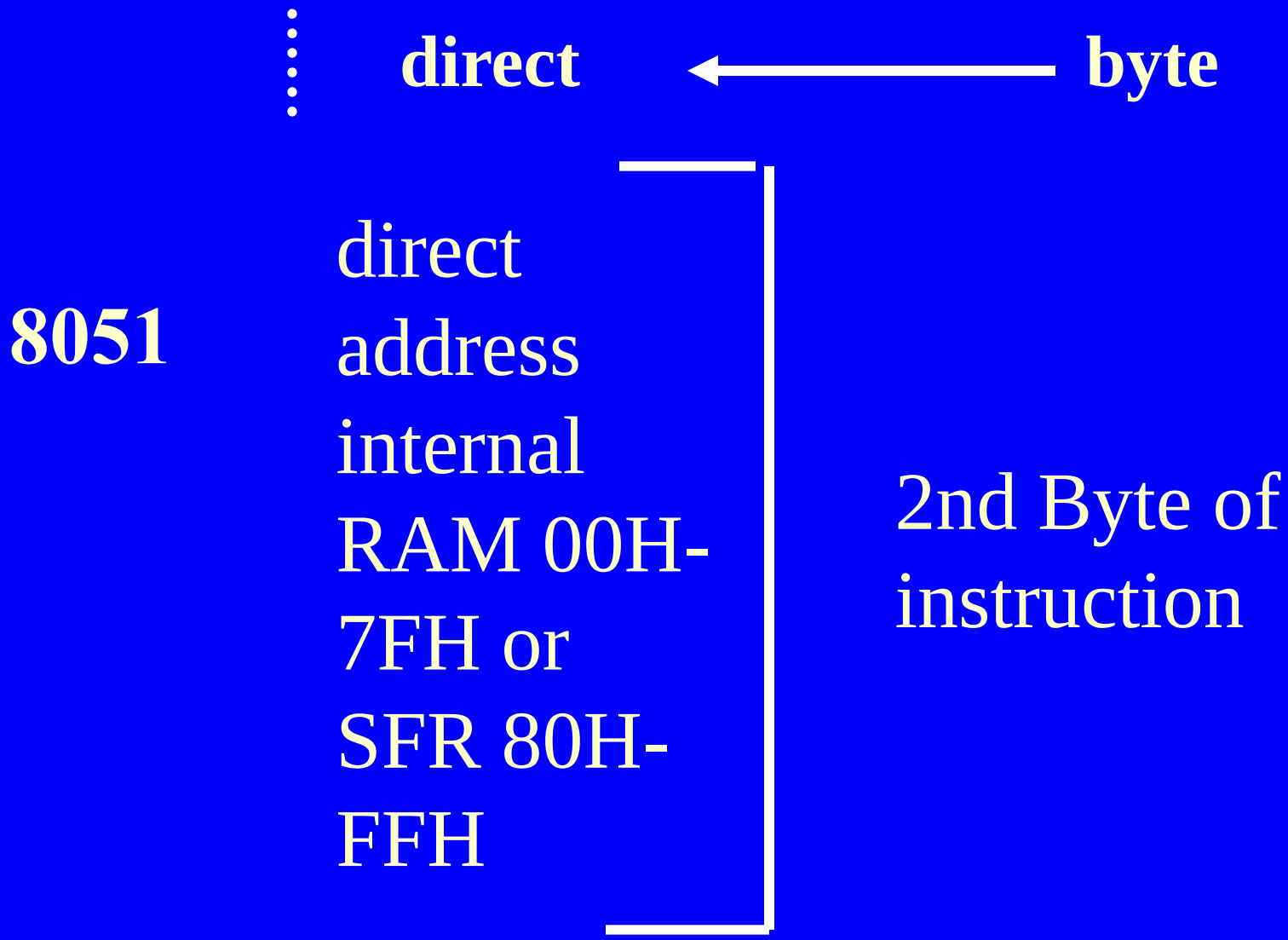
For moving 08H into Rn register

Fetch opcode and then byte for getting the operand for move

MOV Rn, #08H

Sign # specifies that 08H is the immediate succeeding byte is the operand

Codes in Memory- 78H-7FH, 08H


direct
byte
8051
direct address internal RAM 00H-7FH or SFR 80H-FFH
2nd Byte of instruction

MOV direct,#data

2 clock cycles

For moving 08H into direct

Fetch opcode and then bytes for getting the operands

Time

MOV direct, #08H

Codes in Memory- 75H, 00H- FFH, 08H

Sign # specifies that 08H is the immediate succeeding byte is the operand

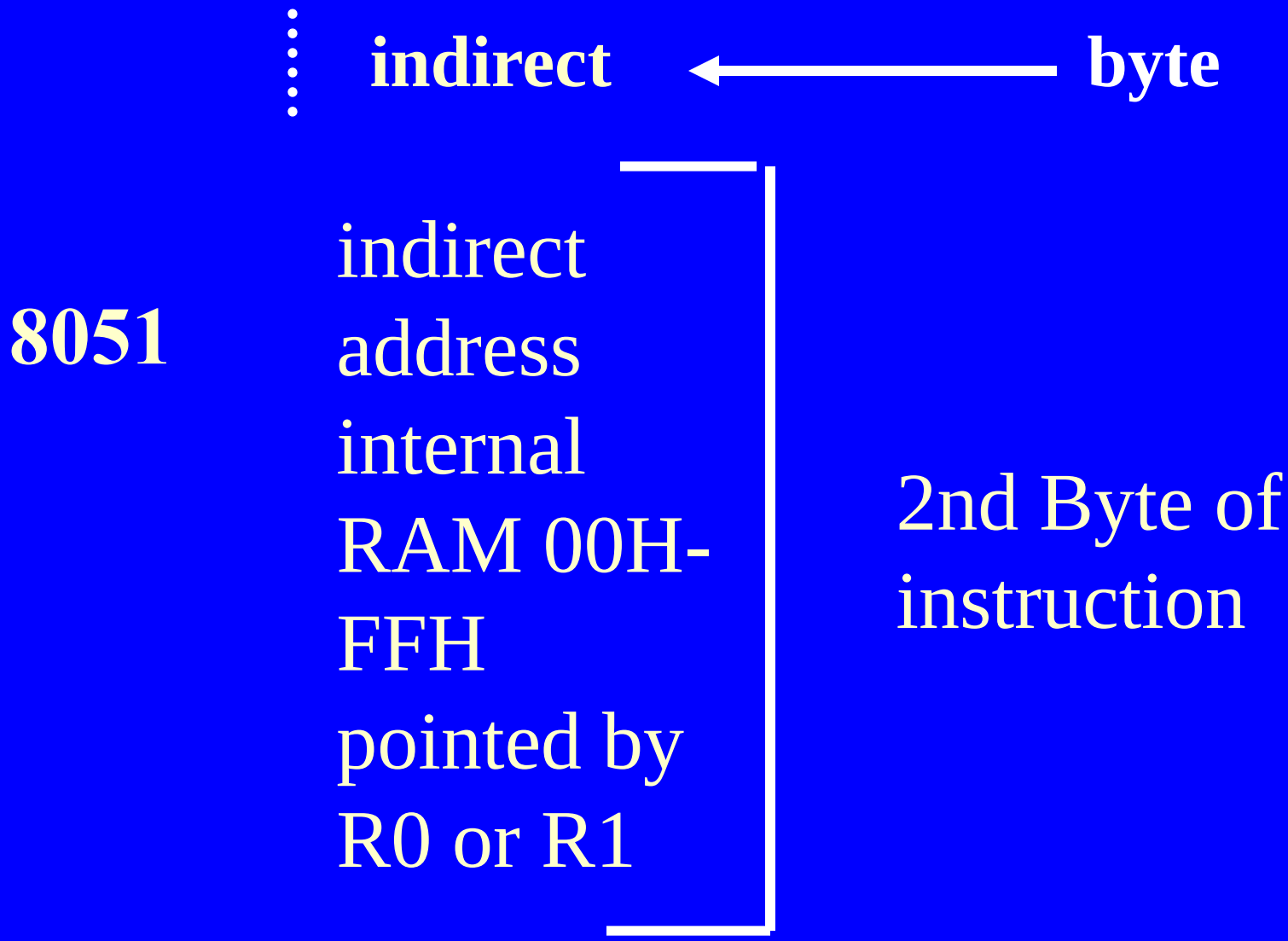
Destination Direct Addressing

STEP 2 → **Fetch byte(s) after the opcode**

clock cycle (s)

↓ **Time**

Addressing mode specifies that fetch byte(s) (next to the opcode of 8 bits) specify an address, destination operand is at that address.


indirect
byte
8051
indirect address internal RAM 00H-FFH pointed by R0 or R1
2nd Byte of instruction

MOV @Ri,#data

For moving 08H into indirect → **Fetch opcode and then bits for getting the operands**

1 clock cycle

Time

MOV @Ri, #08H

Codes in Memory-76-77H, 08H

Sign # specifies that 08H is the immediate succeeding byte is the operand

Destination indirect Addressing

STEP 2 ⟶ **Fetch bit 0 or 1 for R0 or R1 the opcode**

1 clock cycle

Time

Addressing mode specifies that fetch byte(s) specify by an address pointed by R0 or R1 at a bank

Immediate Addressing Four type of instructions

- A
- Rn
- direct for Internal RAM in between 00H to 7FH or direct for SFR 80H-FFH
- Indirect using R0 or R1

Immediate addressing mode for source two-bytes

- Lower byte is second byte of the instruction and is for DPL
- Higher byte is third byte of the instruction and is for DPH

MOV DPTR, #data16

2 clock cycles

For moving 08C1H into DPTR → **Fetch opcode and then 16 bits for getting the operands**

Time

MOV DPTR, #08C1H

Codes in Memory- 90H, C1H, 08H

Sign # specifies that 08C1H is two immediate succeeding bytes as the operand

MOVC and MOVX indirect

1. Only indirect addressing in 8051 instruction for a byte in external data memory

2. Only indirect addressing in 8051 instruction for a code in internal/external program memory

Always Indirect Addressing mode

Internal/External
Program Memory
(max. 64kB)

External Data Memory
and Ports (max 64kB)

MOVC A, @A + PC

For transferring addressed code byte into A

Fetch opcode bits

C means code memory is used

2 clock cycles

Time

STEPS 2-3

MOVC A, @A + PC

Code bits in Memory-83H

indirectly specify, PC16-bits adds with 8-bits at A register and then new 16-bits point to the code at that address

MOVC A, @A + DPTR

For transferring addressed code byte into A

Fetch opcode bits

2 clock cycles

Time

STEPS 2-3

MOV<u>C</u> A, @A + DPTR

indirectly specify, DPTR16-bits adds in 8-bits at A register and then 16-bits point to code at that address

Code bits in Memory-93H

<u>C</u> means code memory is used

MOVX A, @DPTR

For transferring an external byte into A

Fetch opcode

Sign @ means DPTR is a pointer

MOVX A, @DPTR

X specifies external memory, byte at DPTR points indirectly to the address for the operand

2 clock cycles

Time

STEP 2

Code bits in Memory-E0H

MOVX @DPTR, A

For transferring A to a byte at external pointed address

Fetch opcode

Sign @ means DPTR is a pointer

2 clock cycles

Time

STEP 2

Code bits in Memory-F0H

MOVX @DPTR, A

X specifies external memory, byte at DPTR points indirectly to the address for the operand

MOVX A, @Ri

For transferring an external byte into A

Fetch opcode

Sign @ means Ri is a pointer

2 clock cycles

Time

STEP 2

Code bits in Memory- E2-E3H

MOVX A, @Ri

X specifies external memory, byte at Ri points indirectly to an external memory address for the operand

MOVX @Ri, A

For transferring A to a byte at external pointed address

Fetch opcode

Sign @ means Ri is a pointer

2 clock cycles

Time

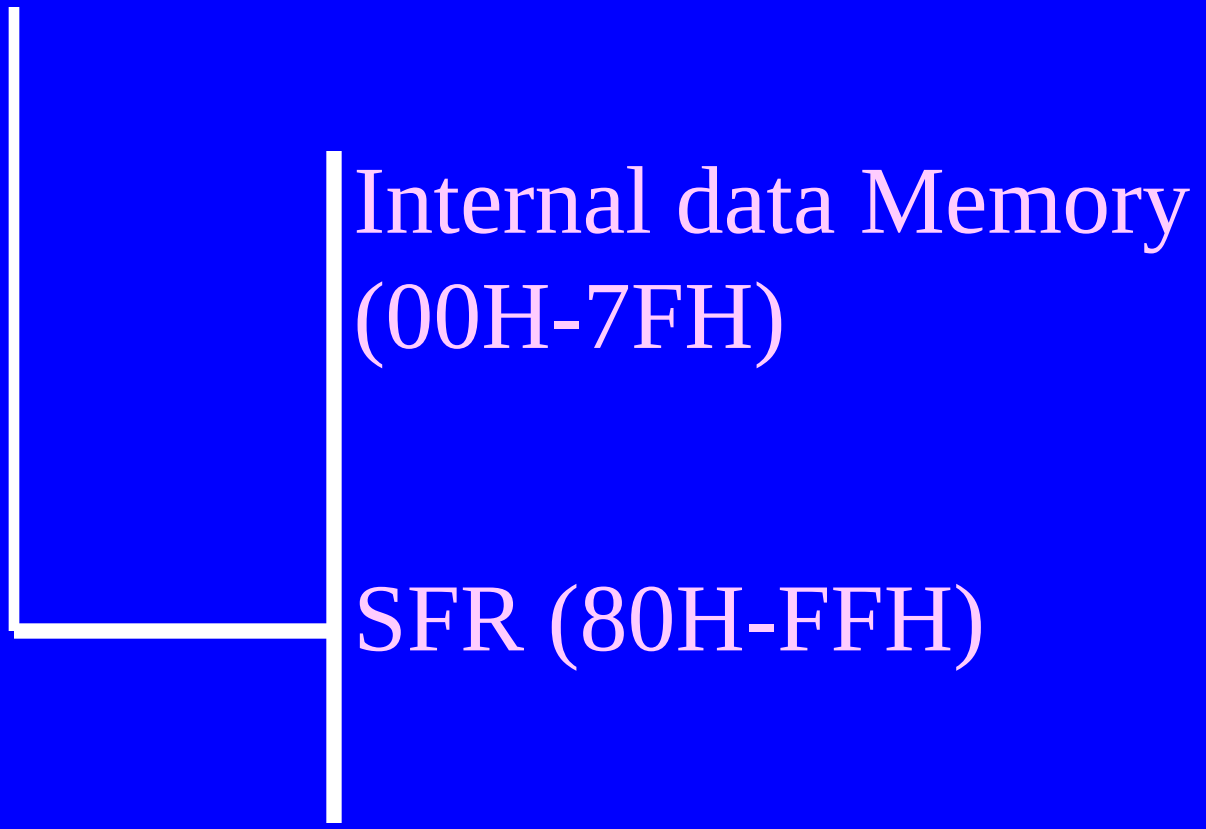
MOVX @Ri, A

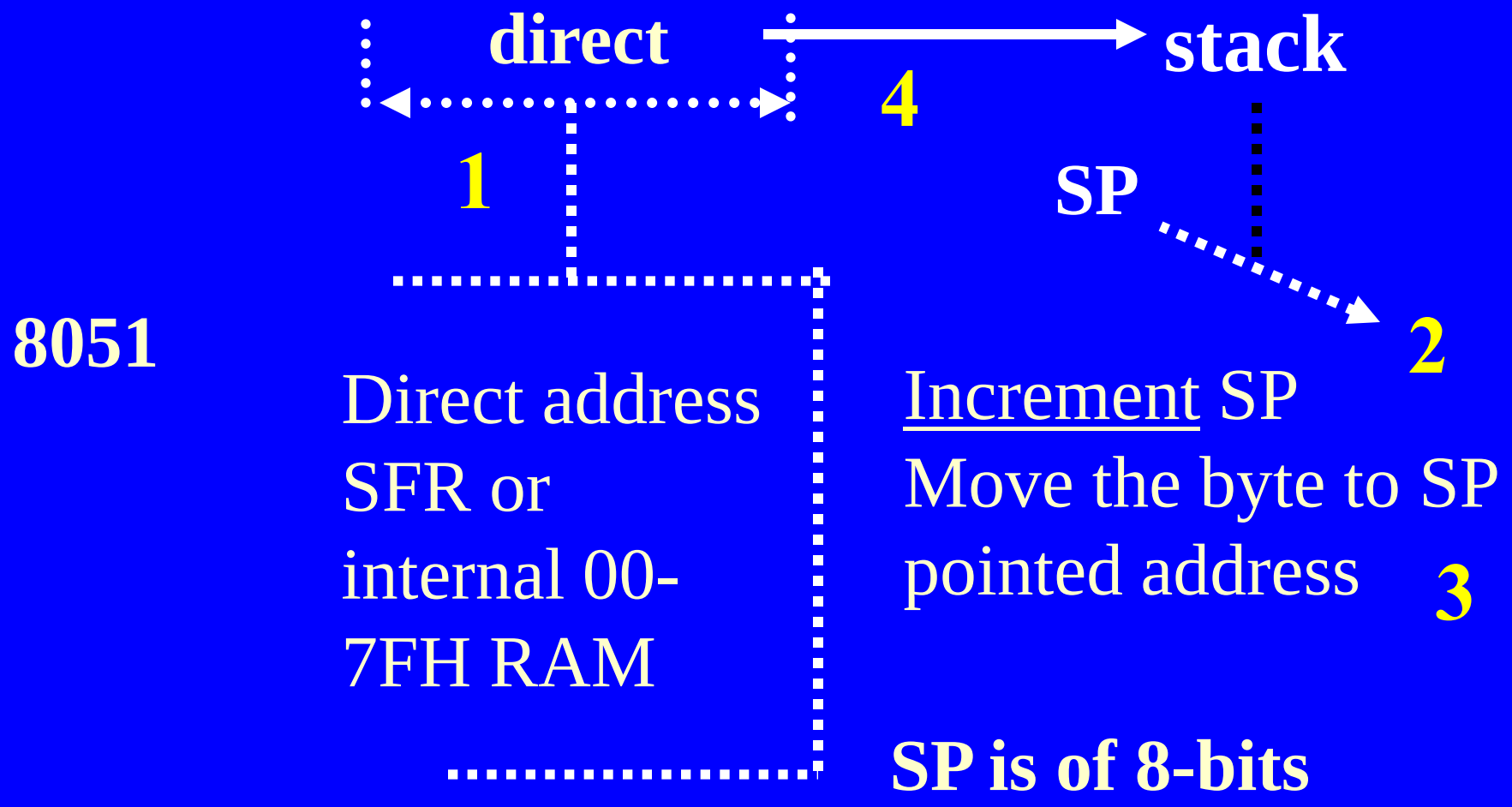
X specifies external memory, byte at Ri points indirectly to the address for the operand

Code bits in Memory- F2-F3H

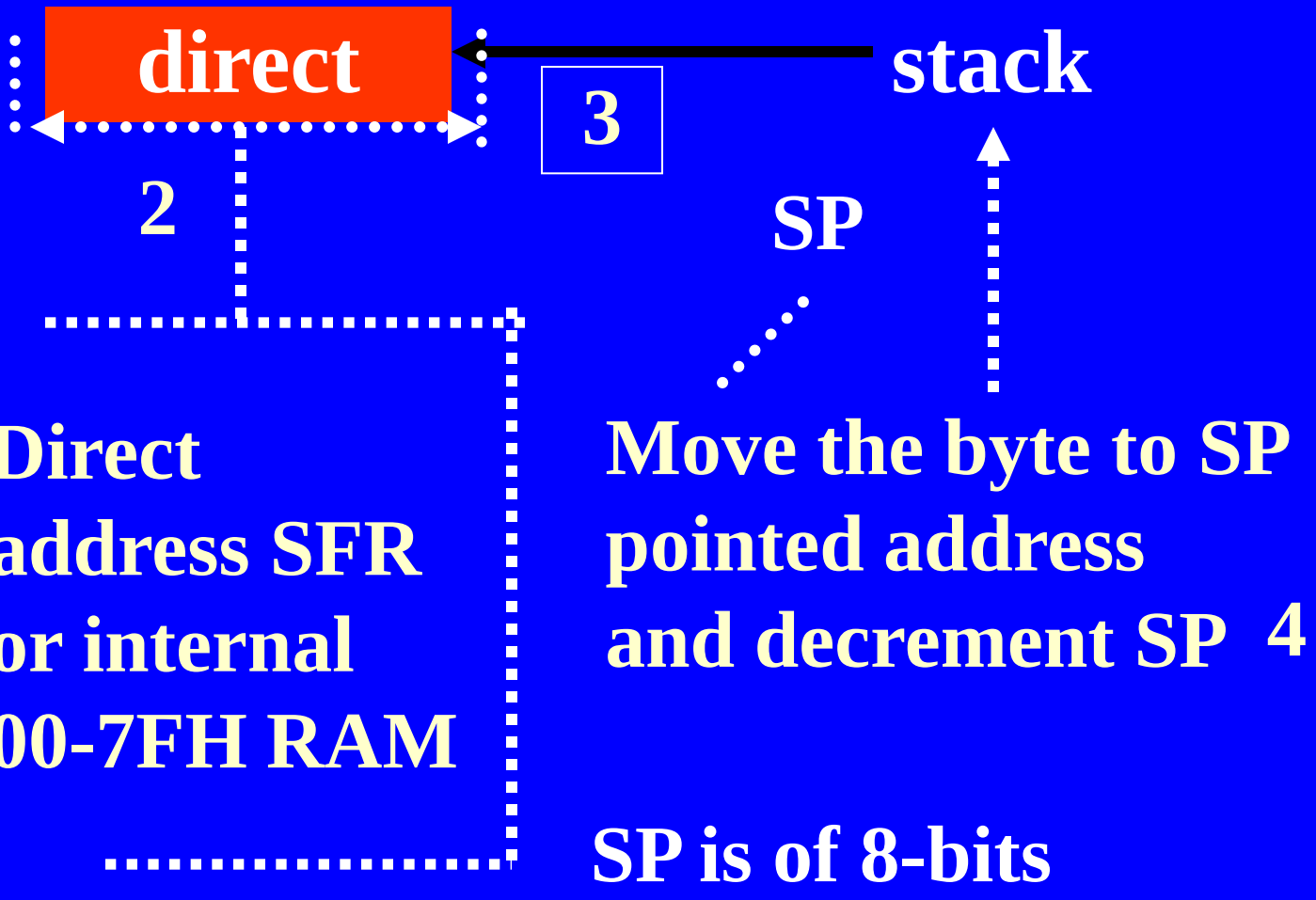
Push or Pop *direct* or Exchange or Swap

Always direct Addressing mode for Push or Pop




direct
stack
4
1
SP
2
8051
Direct address SFR or internal 00-7FH RAM
Increment SP
Move the byte to SP pointed address
3
SP is of 8-bits

Push *direct*


direct
stack
3
2
SP
8051
Direct address SFR or internal 00-7FH RAM
Move the byte to SP pointed address and decrement SP
4
SP is of 8-bits

Pop *direct*

Instruction Execution

2 clock cycles

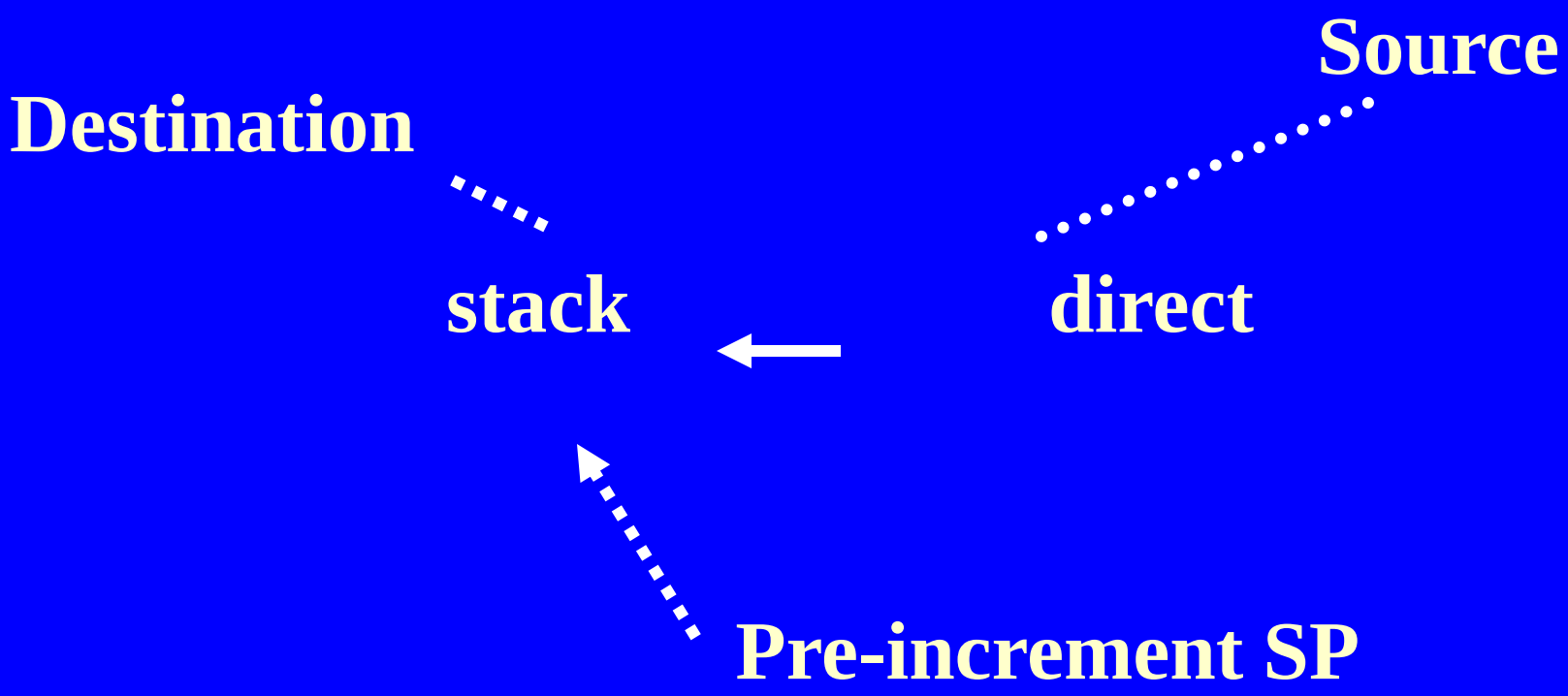
STEP 1 — **Fetch opcode-bits**

STEP 2 — **Fetch byte direct**

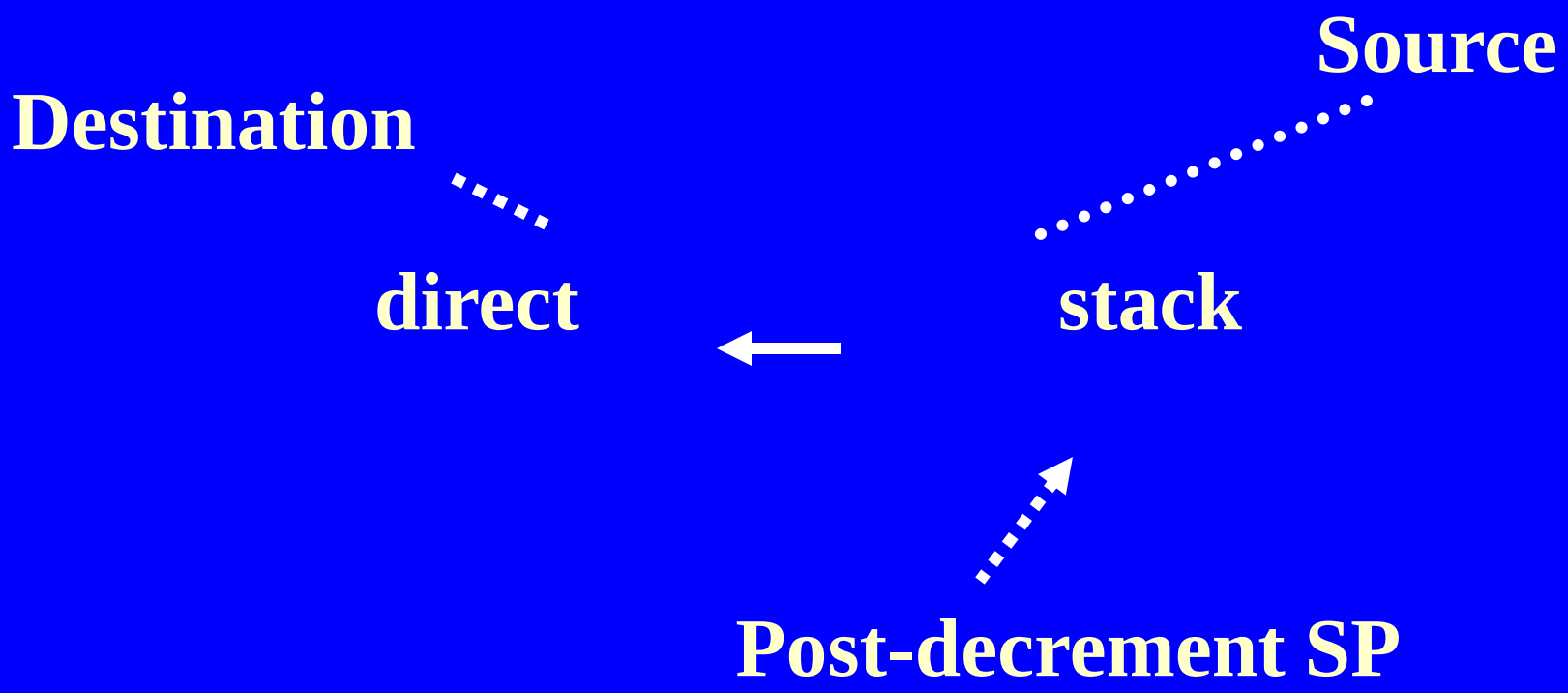
Time

direct addressing mode for source operand in push and for destination operand for pop

Push an SFR or Internal RAM between 00H-7FH

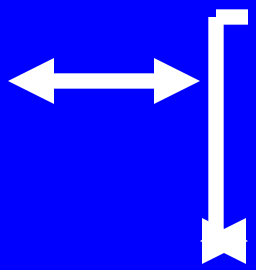


Pop an SFR or Internal RAM between 00H-7FH



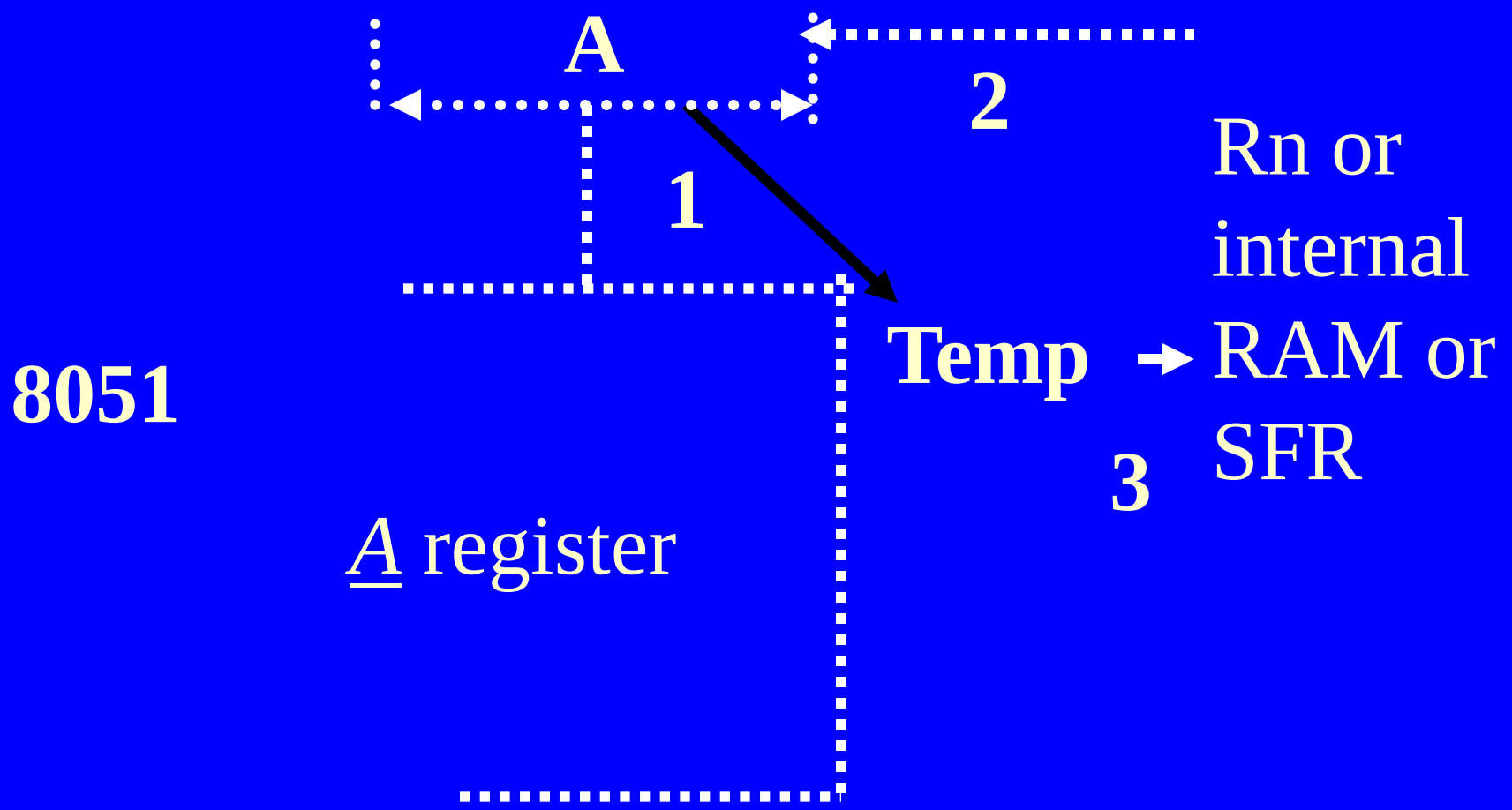
Exchange of A register byte with other

A register



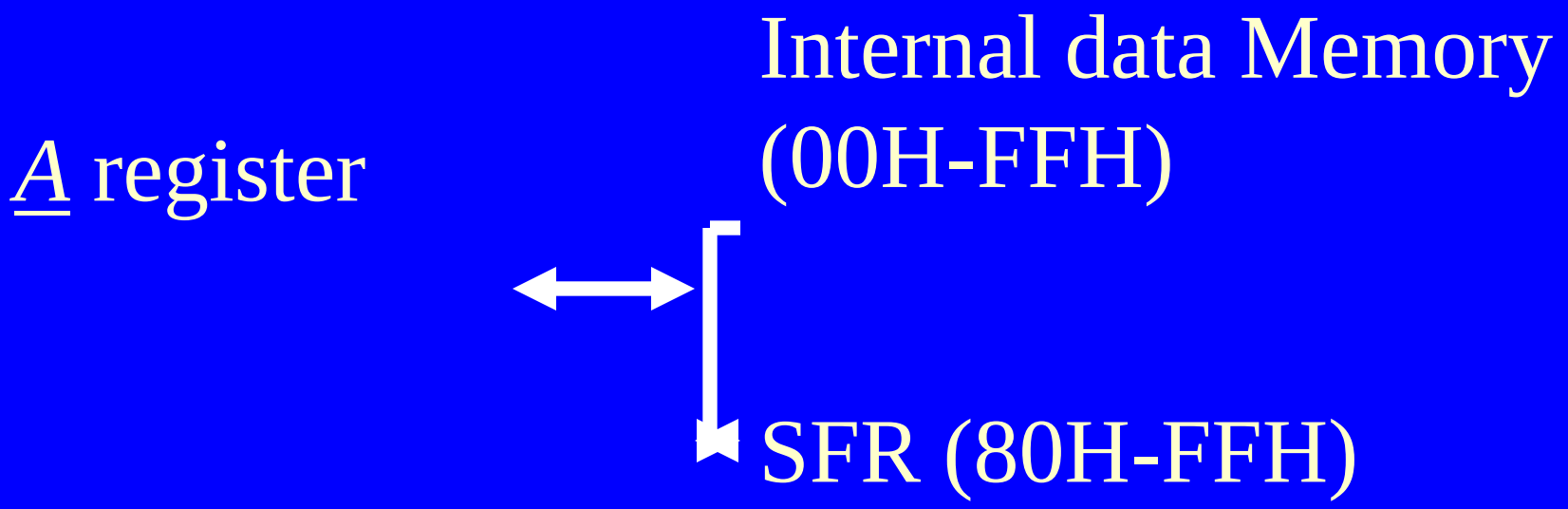
Internal data Memory (00H-FFH)

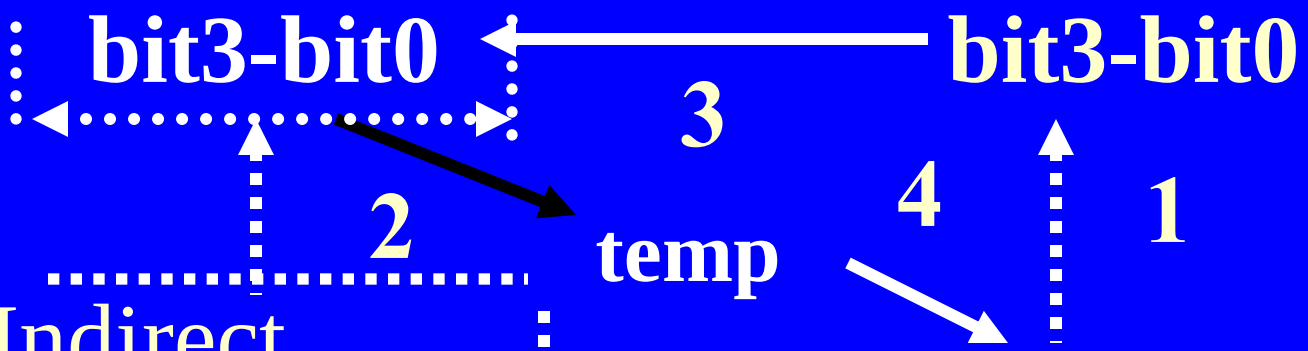
SFR (80H-FFH)



XCH A, Rn, XCH A, *direct*,
XCH A, @Ri; exchange bytes

Exchange of A register lower digit with other





8051

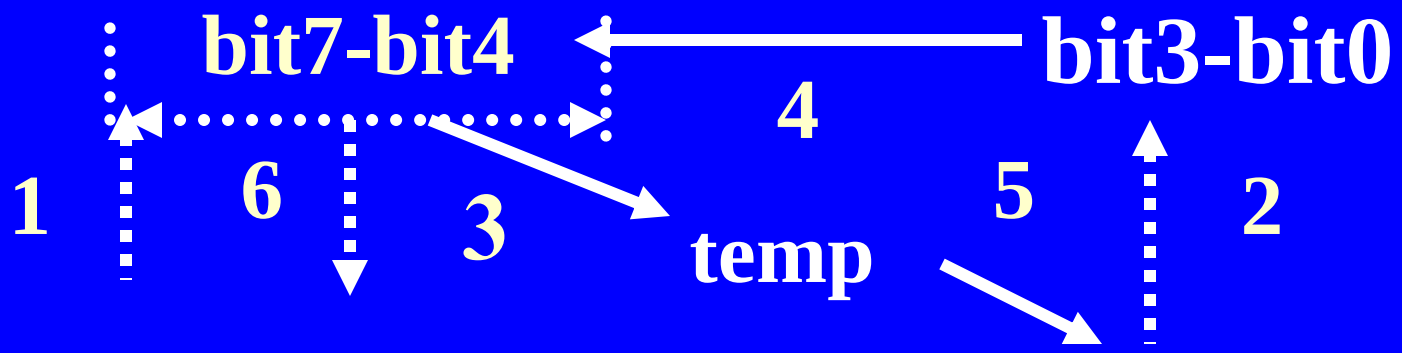
Indirect address internal 00-FFH RAM pointed by R0 or R1

A register lower 4 bits

XCHD A, @Ri; exchange lower digits (4-bits)

Exchange of *A* register lower digit with higher digit

A register lower bits  *A* register higher bits



8051

A register lower 4 bits

A register higher 4 bits

SWAP A; exchange of lower digit (4-bits) with higher at *A*

Summary

We learnt

8051 family data transfer instructions

- move (copy)
- push
- pop
- exchange
- exchange lower 4-bits