Chapter 4

8051 Family Microcontrollers Instruction Set

Lesson 4

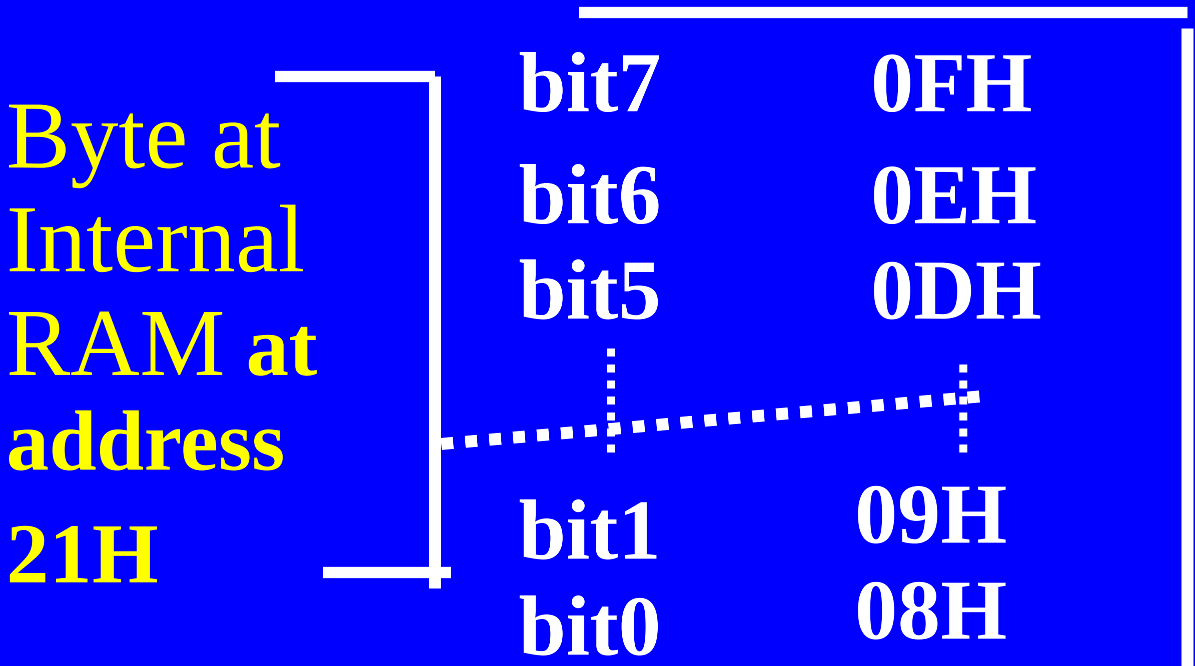
Boolean, Arithmetic and Logic Processing Instructions

Addresses of the Bit at Port

- Bit 5 port P0 bit address = 85H
- Bit 3 of A register address = E3H
- Bit 7 PSW (C bit) address = D7H

Bit-Address of Bits at Internal RAM

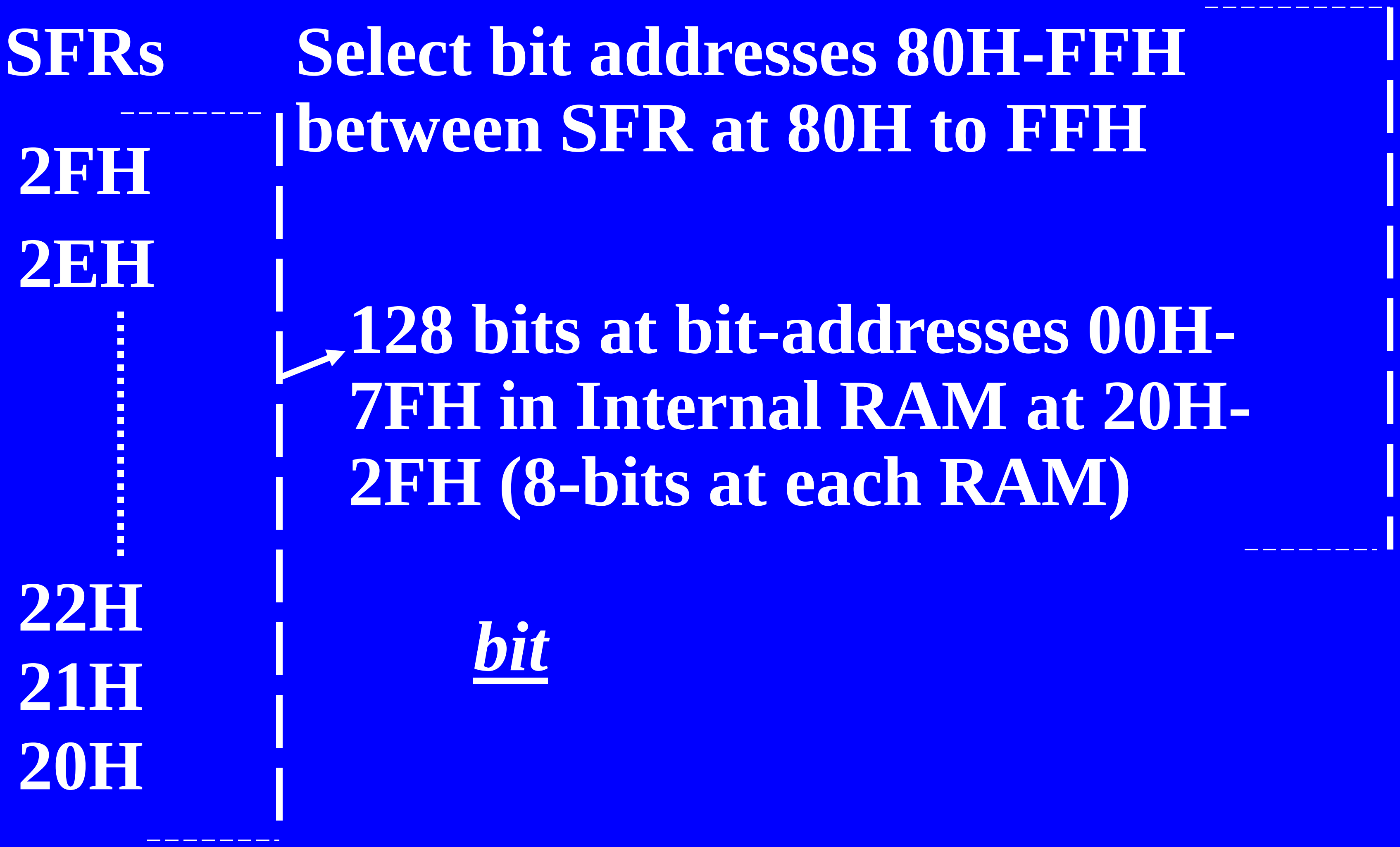
Bit-Address of Bit 5 at Internal RAM at 21H = 0DH



Byte at Internal RAM at **address 21H**

bit7	**0FH**
bit6	**0EH**
bit5	**0DH**
bit1	**09H**
bit0	**08H**

Source or Destination of a bit



SFRs

Select bit addresses 80H-FFH between SFR at 80H to FFH

2FH

2EH

22H

21H

20H

128 bits at bit-addresses 00H-7FH in Internal RAM at 20H-2FH (8-bits at each RAM)

bit

Instruction Execution

STEP 1 **Fetch opcode-bits**

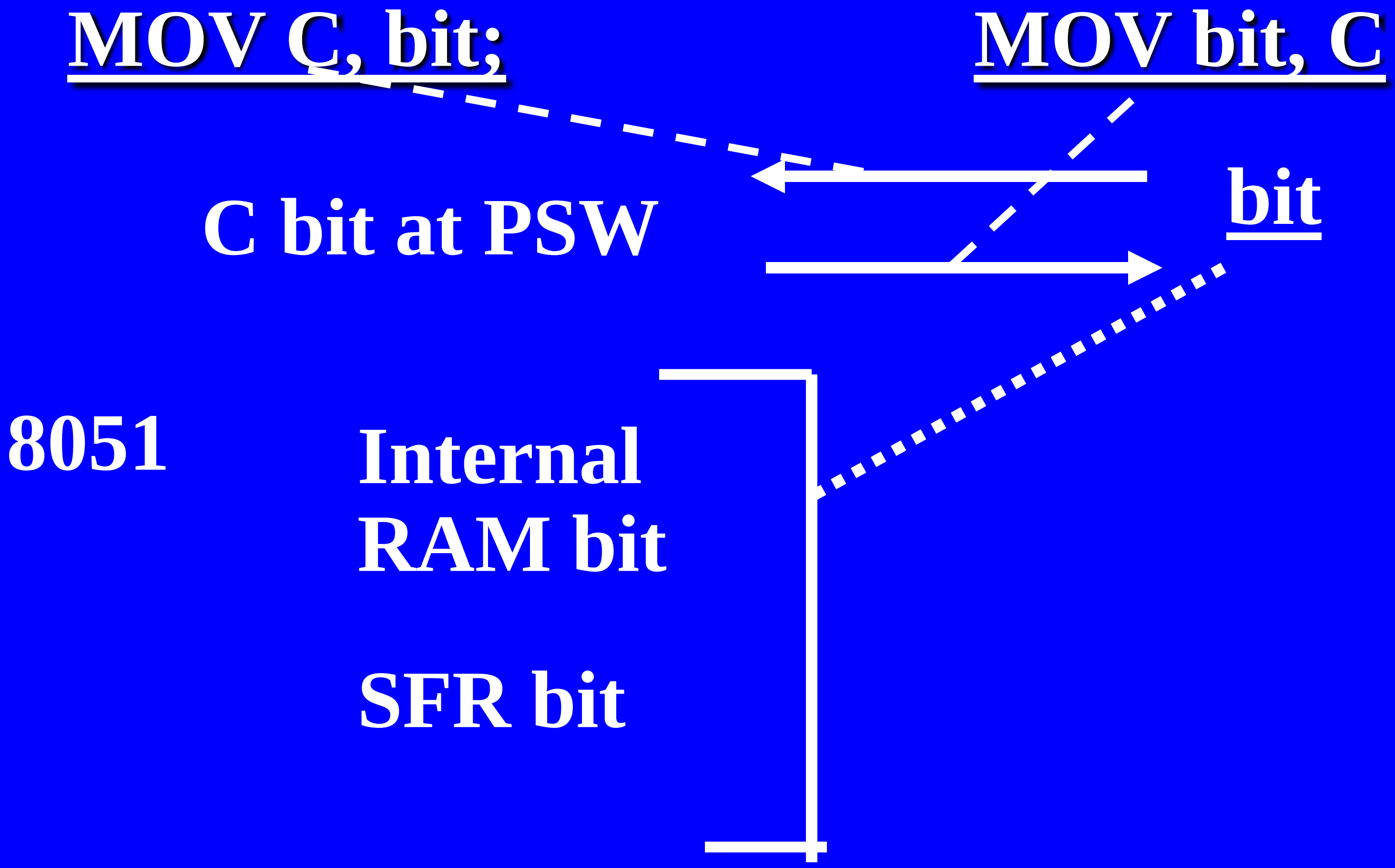
STEP 2 **Fetch byte for *<u>bit</u>* address**

bit addressing Mode:

<u>clock cycle (s)</u>

- 1 cycle for MOV C, bit
- CLR bit
- SETB bit and
- CPL bit else 2 cycles

Time



MOV C, bit;

MOV bit, C

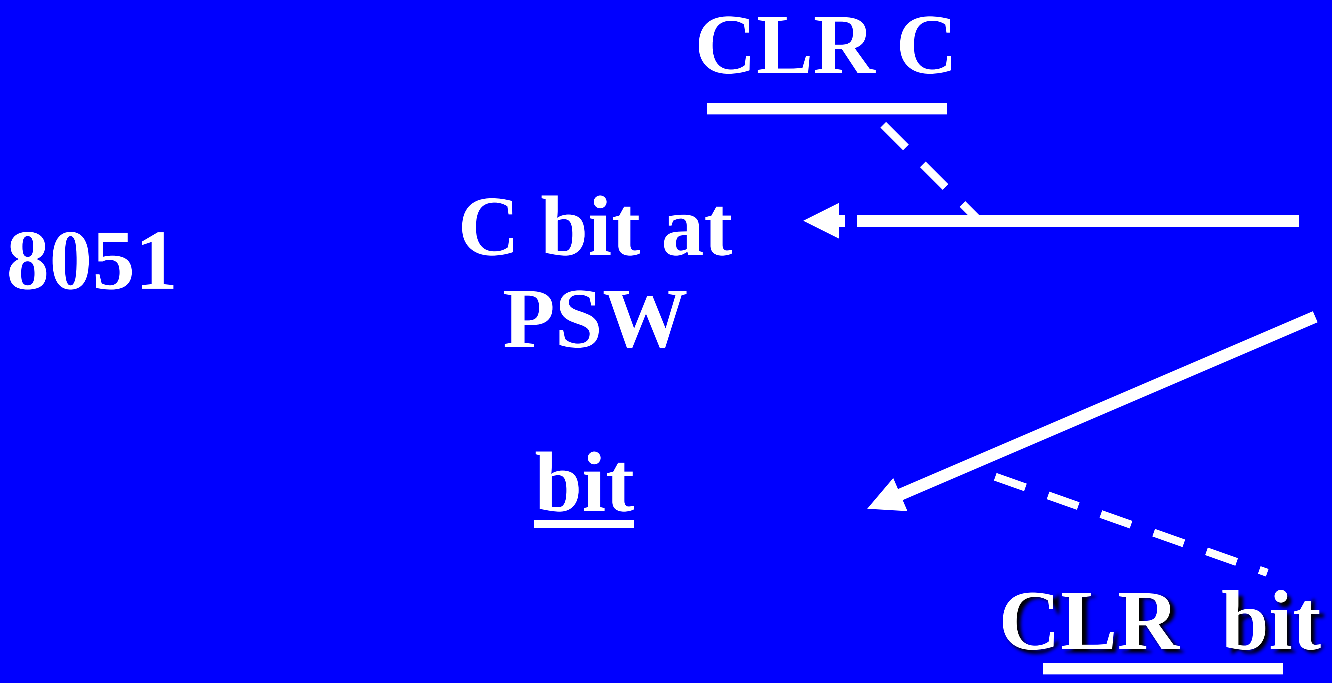
C bit at PSW

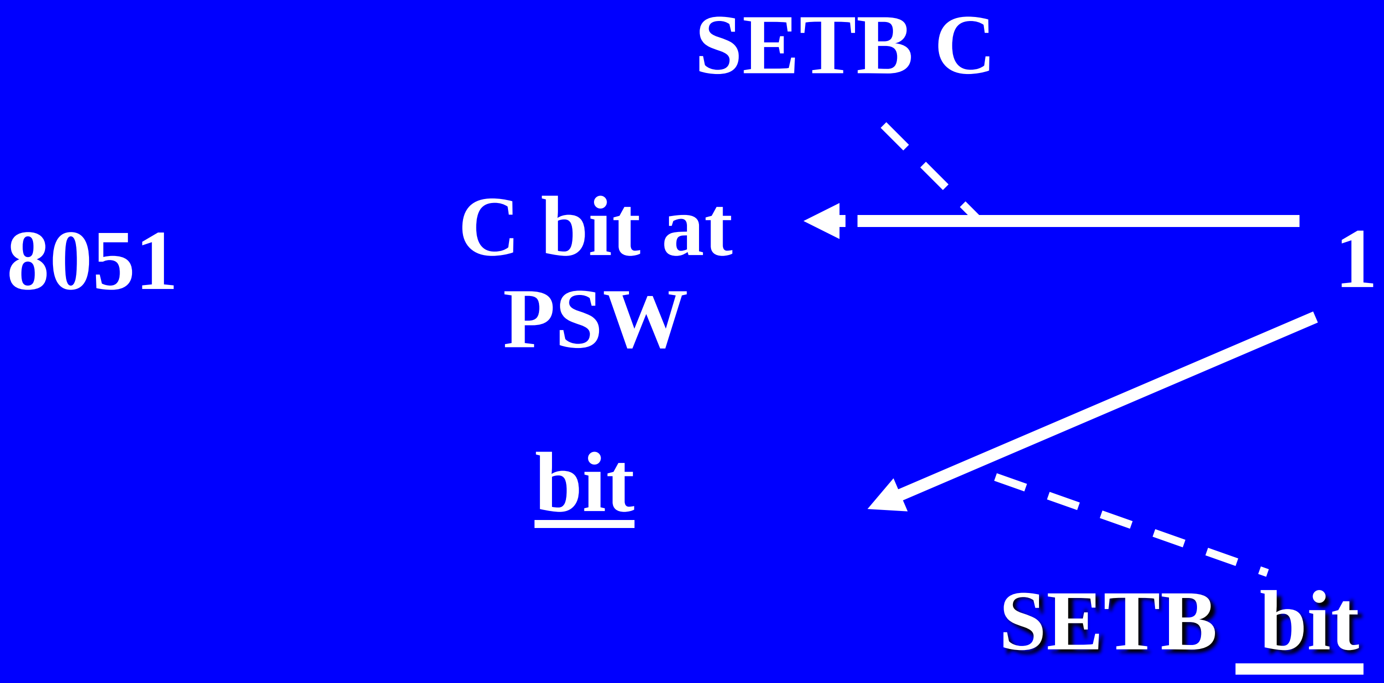
bit

8051

Internal RAM bit

SFR bit


CLR C
8051
C bit at PSW
0
bit
CLR bit



SETB C

8051

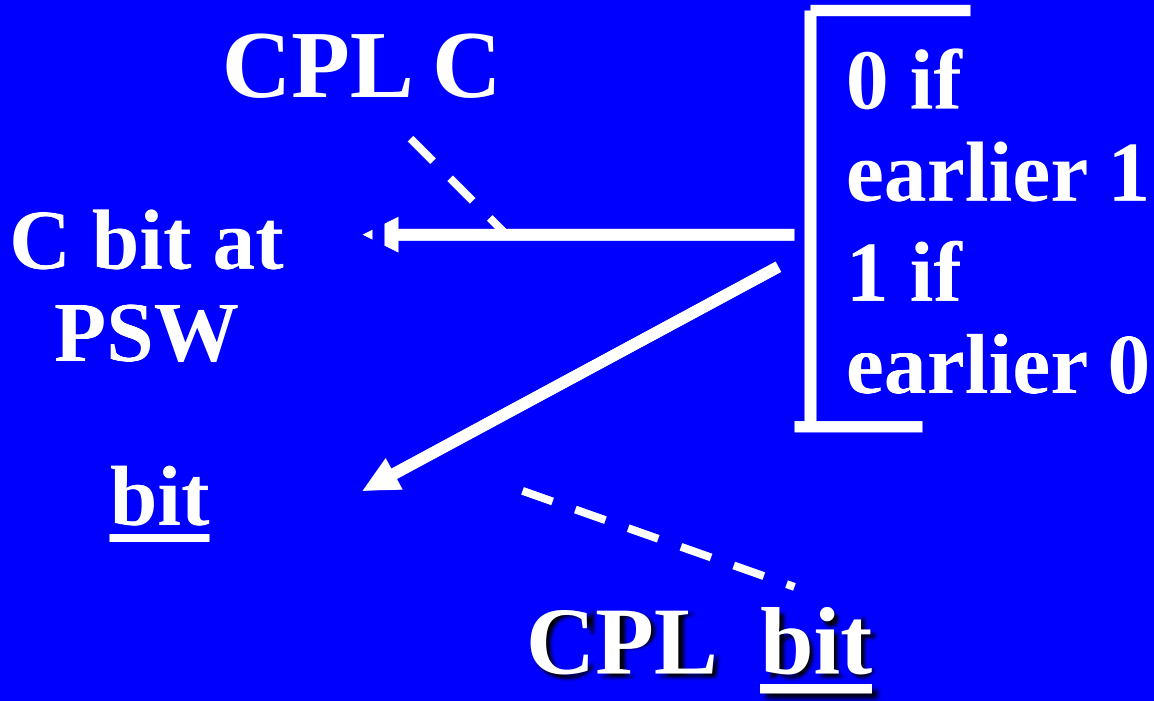
C bit at PSW

1

bit

SETB bit

8051



CPL C

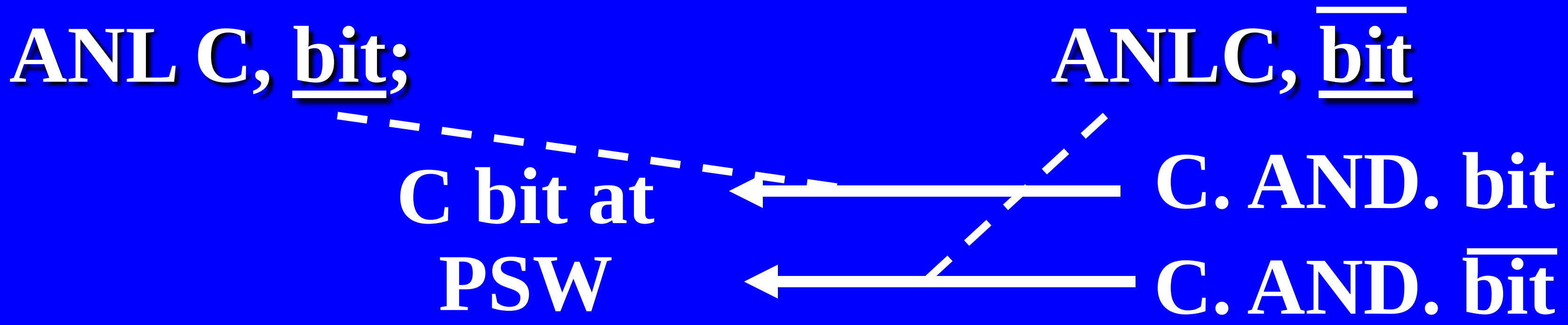
C bit at PSW

bit

0 if earlier 1

1 if earlier 0

CPL bit



ANL C, <u>bit</u>;

ANLC, $\overline{\text{bit}}$

C bit at PSW

C. AND. bit

C. AND. $\overline{\text{bit}}$

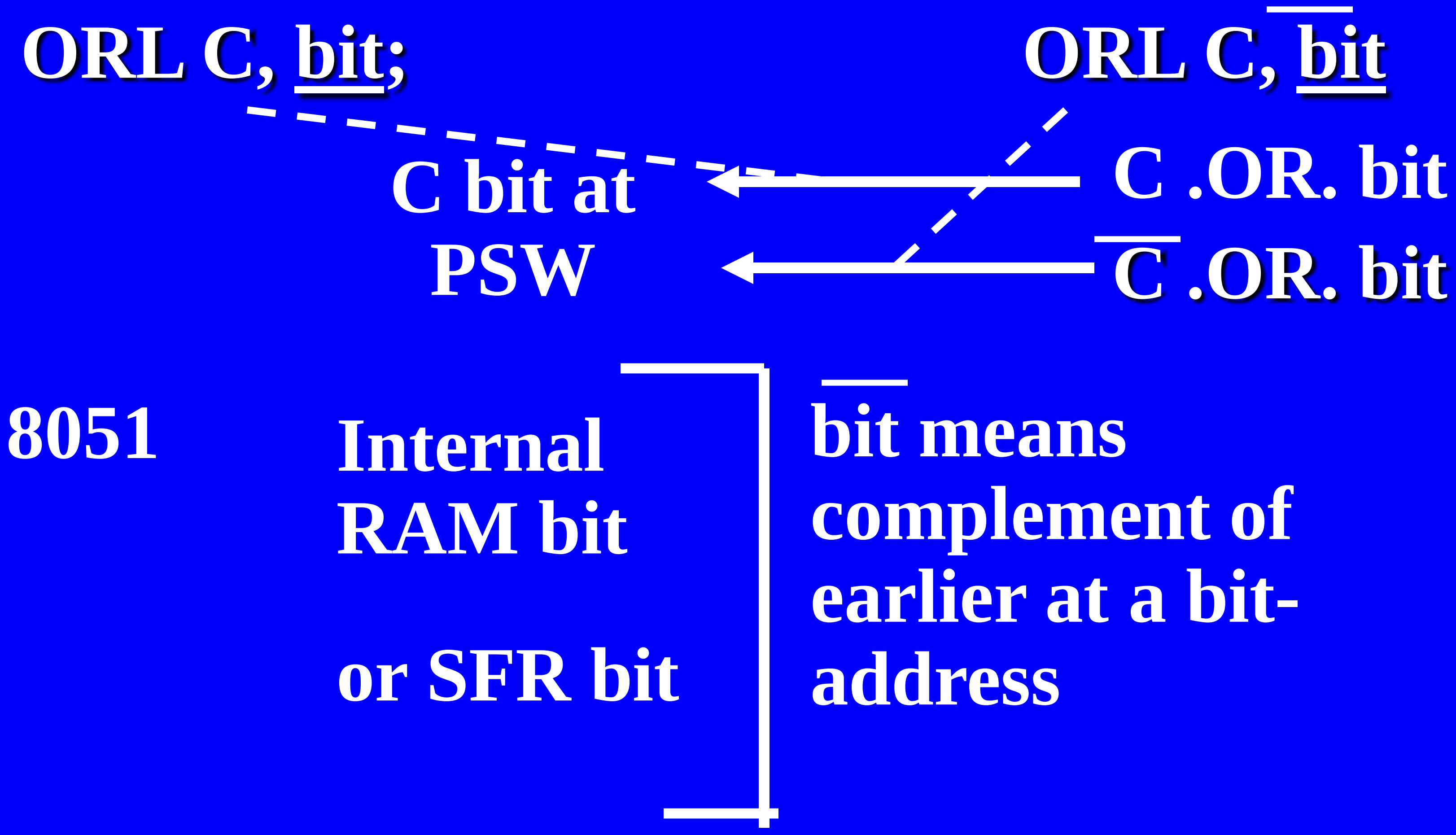
8051

Internal RAM bit

or SFR bit

$\overline{\text{bit}}$ means complement of earlier at a bit-address

<u>ANL means logical AND between two bits</u>



ORL C, bit;

ORL C, $\overline{\text{bit}}$

C bit at PSW

C .OR. bit

$\overline{\text{C}}$ .OR. bit

8051

Internal RAM bit

or SFR bit

$\overline{\text{bit}}$ means complement of earlier at a bit-address

ORL means logical OR between two bits

CLR A;

CPL A;

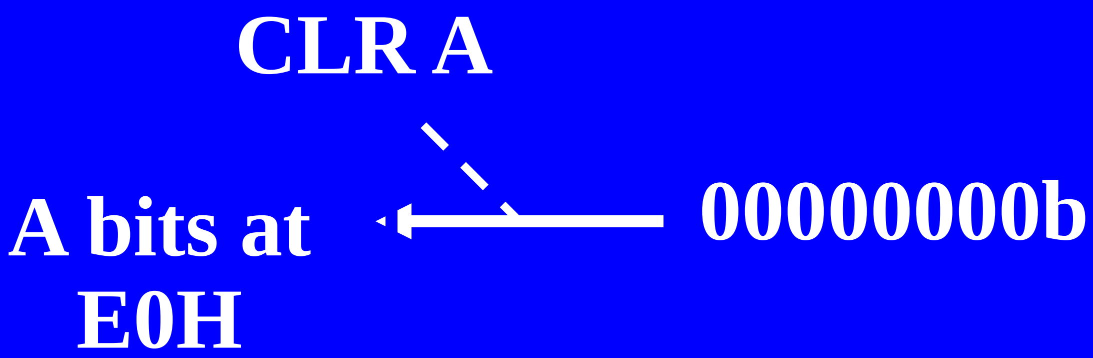
RR A;

RL A;

RLC A;

RRC.A;

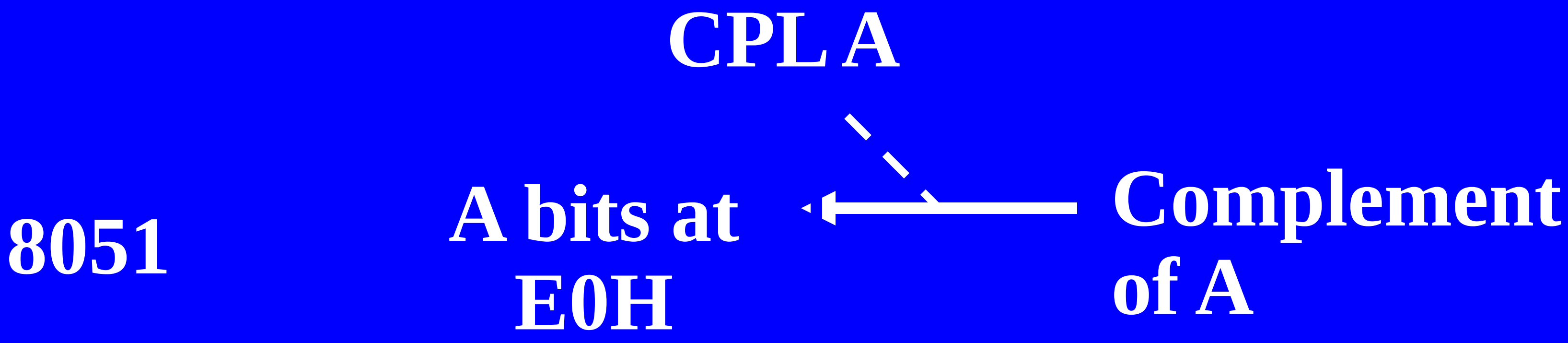
8051



CLR A

A bits at E0H

00000000b



CPL A

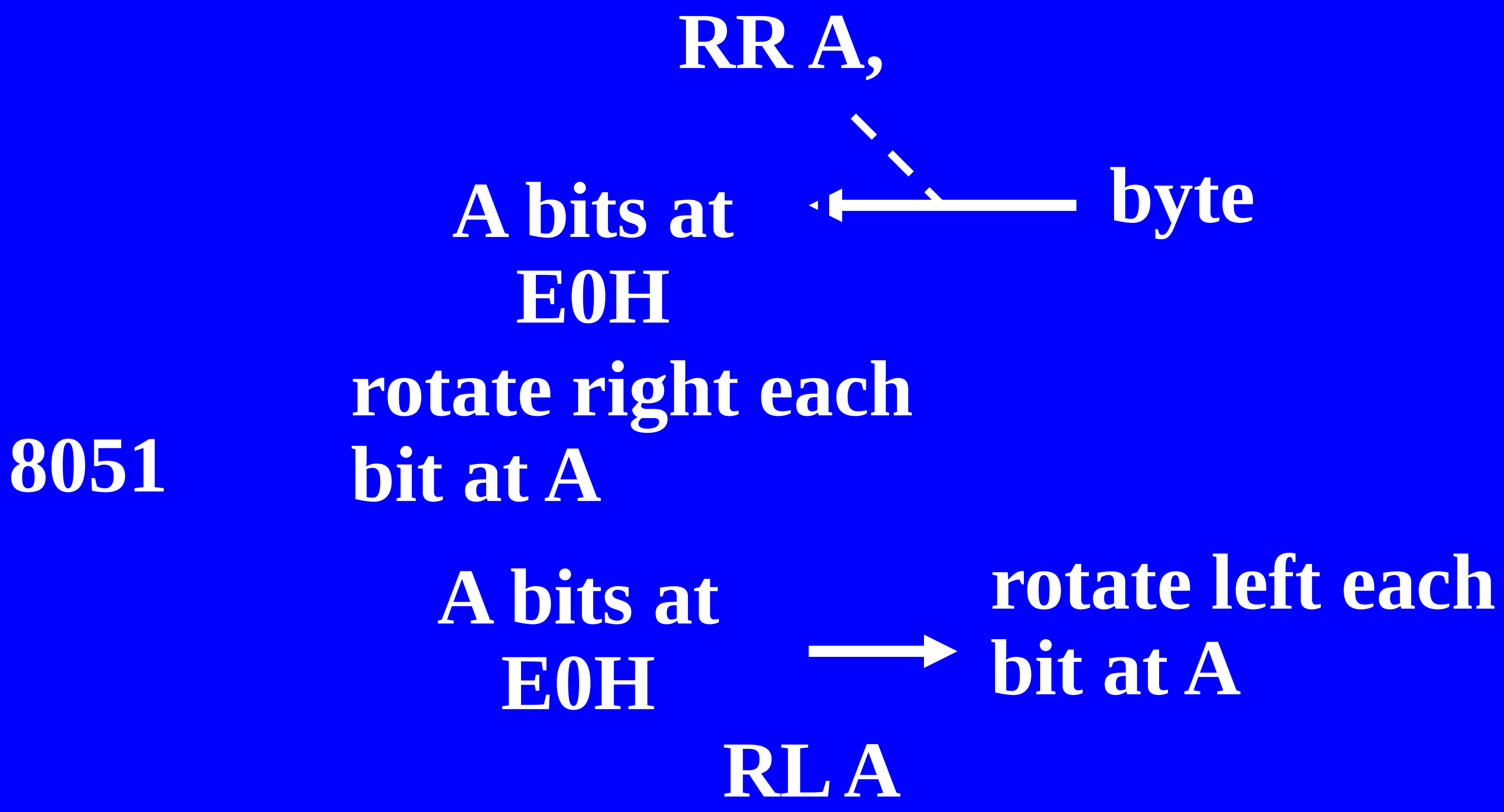
8051

A bits at E0H

Complement of A

Complement each bit at A;

0 → 1 and 1 → 0



RR A,

A bits at E0H ← **byte**

rotate right each bit at A

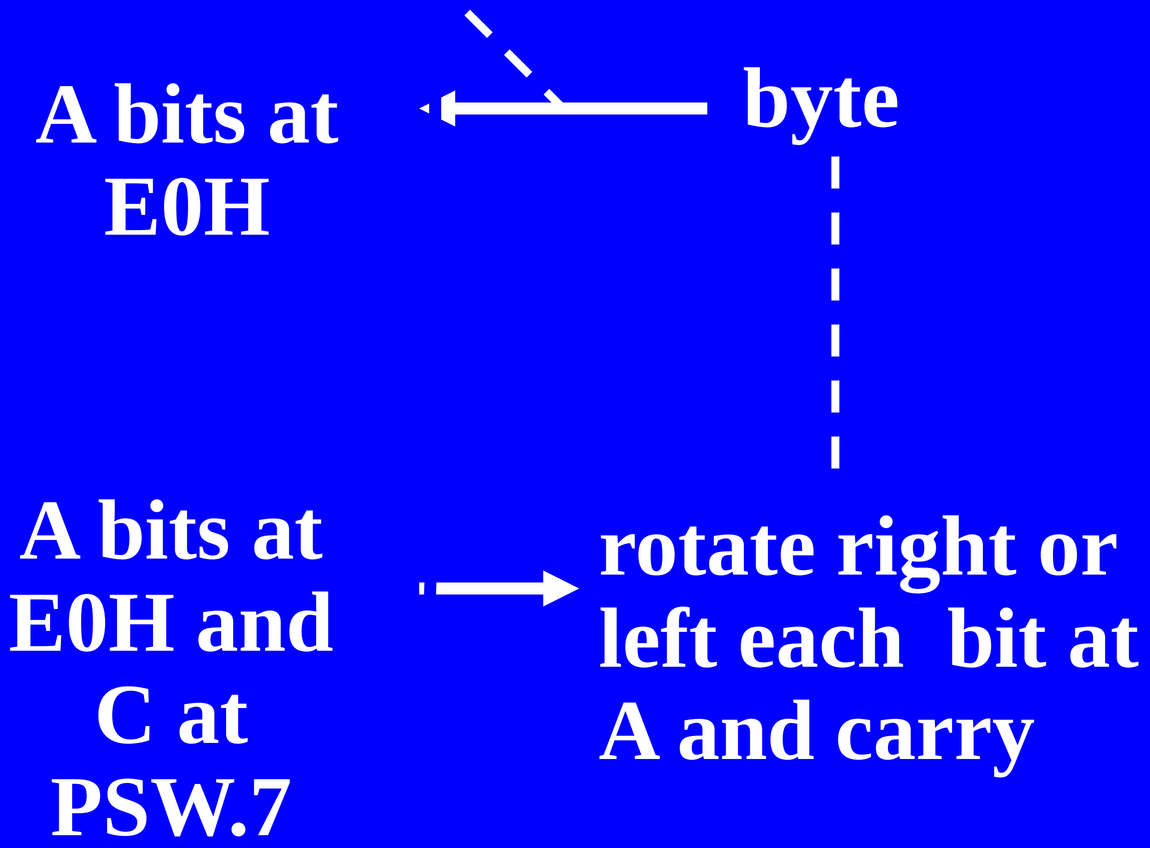
8051

A bits at E0H → **rotate left each bit at A**

RL A

RRC A, RLC A

8051



A bits at E0H

byte

A bits at E0H and C at PSW.7

rotate right or left each bit at A and carry

Arithmetic Instructions

- ADD, ADDC, SBB, MUL and DIV
- A is accumulator and is source-1 cum destination operand

```
ADD A, @Ri;
ADD A, Rn;
ADD A, direct;
ADD A, #data
```

C, AC and OV Flags at PSW also affect

A ← A + byte

8051

ADD without carry

From *Rn* or *direct* or *indirect* or *immediate*

ADDC A, @Ri;

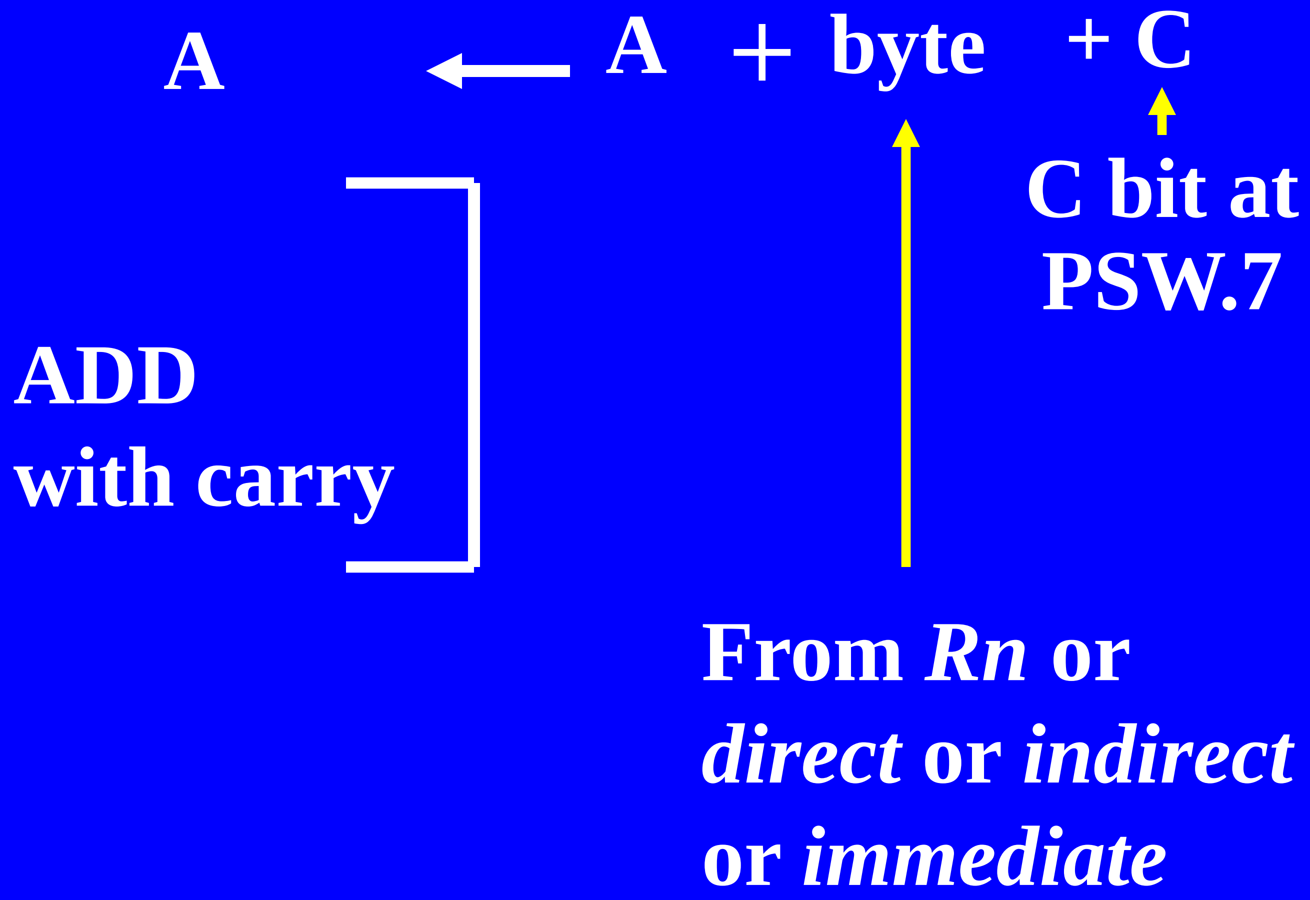
ADDC A, Rn;

ADDC A, direct;

ADDC A, #data

C, AC and OV Flags at PSW also affect

8051



A ← A + byte + C

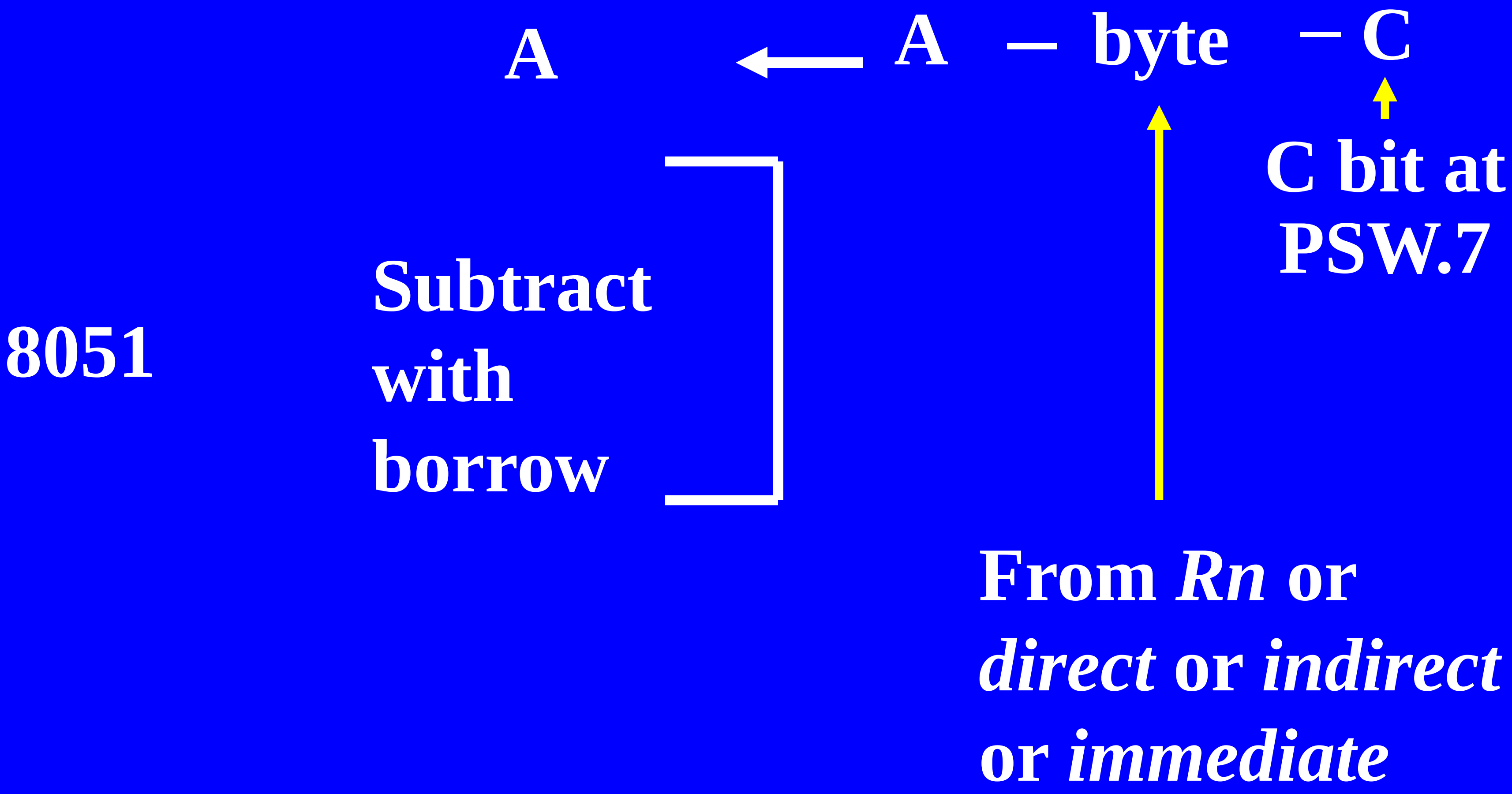
ADD with carry

C bit at PSW.7

From *Rn* or *direct* or *indirect* or *immediate*

SBB A, @Ri;
SBB A, Rn;
SBB A, direct;
SBB A, #data

C, AC and OV Flags at PSW also affect



A ← A − byte − C

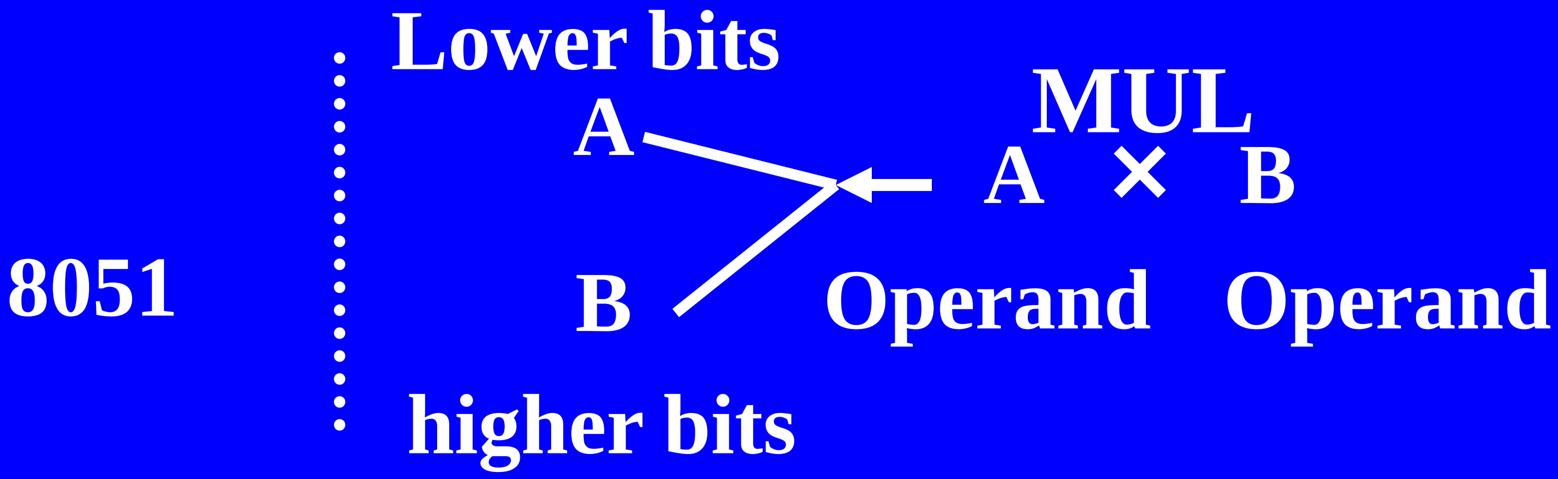
C bit at PSW.7

8051

Subtract with borrow

From *Rn* or *direct* or *indirect* or *immediate*

MUL AB



Lower bits

MUL

A

A × B

8051

B

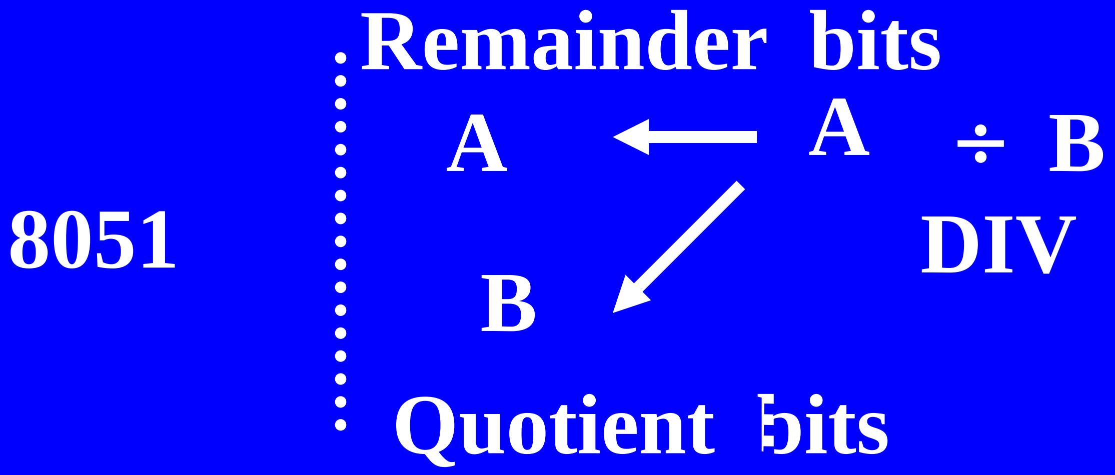
Operand Operand

higher bits

C ← 0 and

OV at PSW also affect

DIV AB



8051

Remainder bits

A ← A ÷ B

DIV

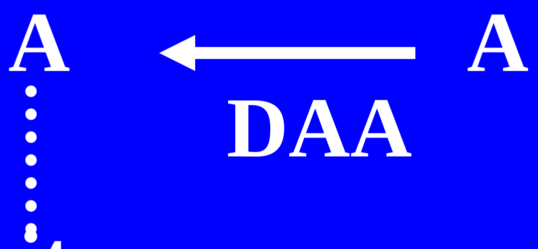
B

Quotient bits

C ← 0

and OV at PSW also affect

DAA AB



A

A

DAA

8051

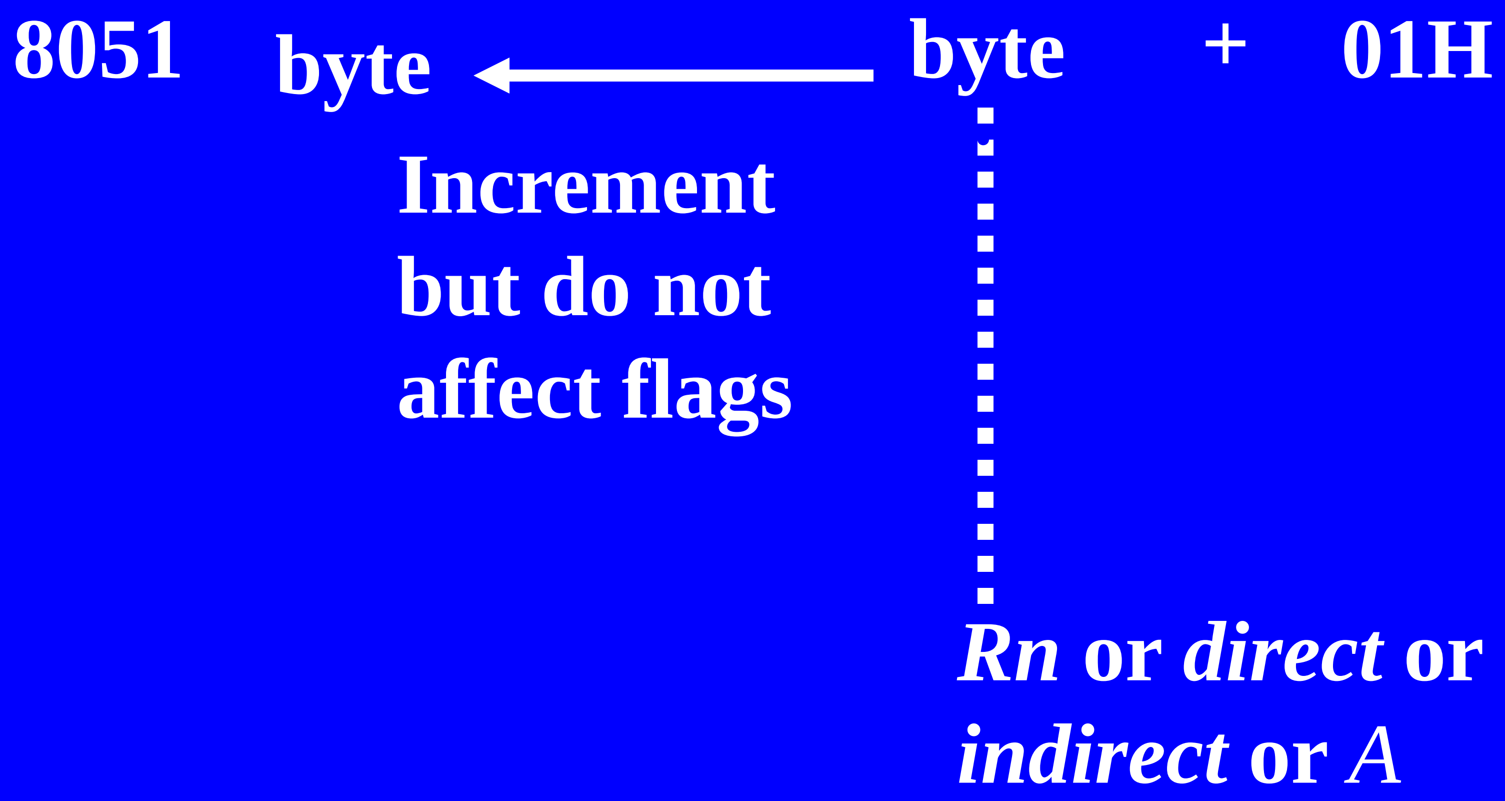
Adjust bits assuming previous add as BCD add

INC A;

INC @Ri;

INC direct;

INC Rn



8051 byte ← byte + 01H

Increment but do not affect flags

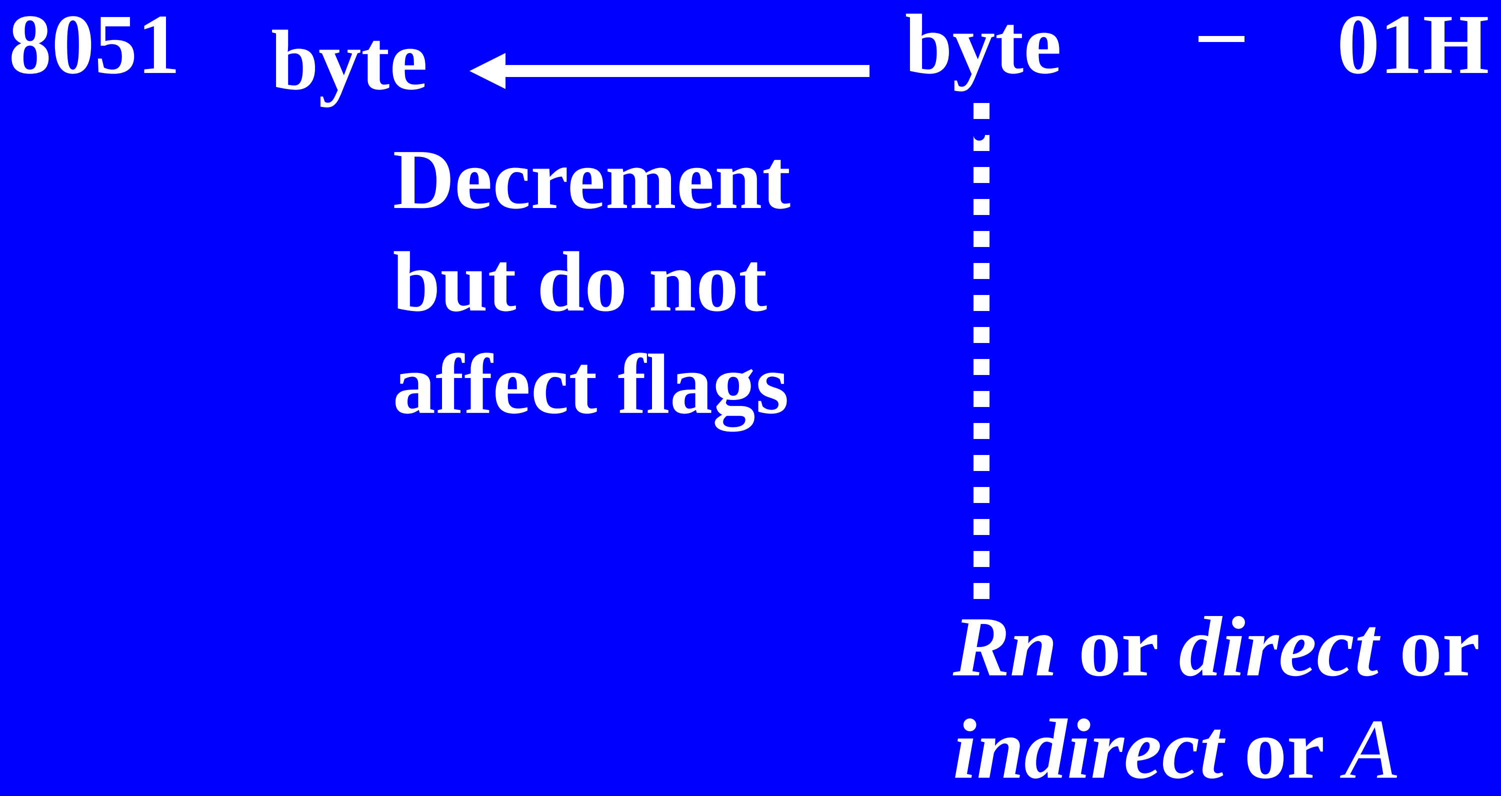
Rn* or *direct* or *indirect* or *A

DEC A;

DEC @Ri;

DEC direct;

DEC Rn



8051 byte ← byte − 01H

Decrement but do not affect flags

Rn* or *direct* or *indirect* or *A

- ANL, ORL, XRL
- A or direct source-1 cum destination operand
- No flag affects in 8051 logic operations

```
ANL A, @Ri;
ANL A, Rn;
 ANL A, direct;
ANL A, #data;
```

No Flags at PSW affect

A ← A . AND. byte

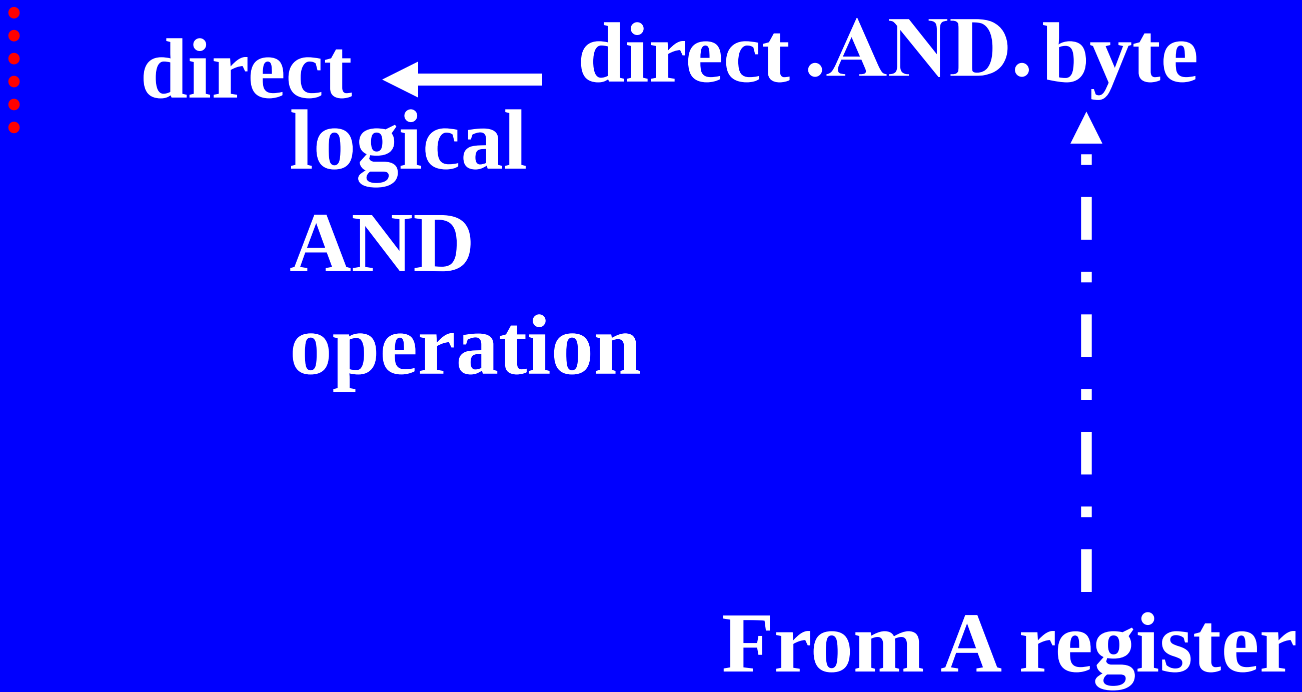
logical AND operation

8051

From *Rn* or *direct* or *indirect* or *immediate*

ANL direct, A;

No Flags at PSW affect



direct ← direct .AND. byte

logical AND operation

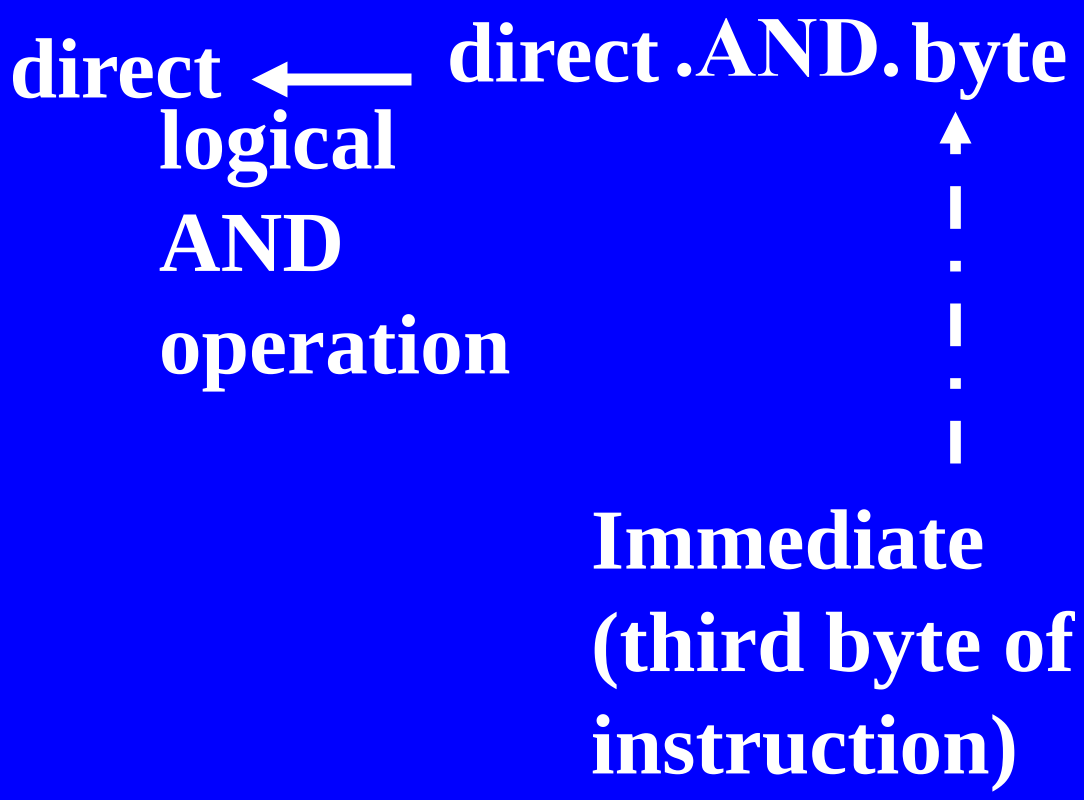
From A register

8051

ANL direct, #data;

No Flags at PSW affect

8051



direct ← direct .AND. byte

logical AND operation

Immediate (third byte of instruction)

```
ORL A, @Ri;
ORL A, Rn;
ORL A, direct;
ORL A, #data;
```

No Flags at PSW affect

A ← A . OR. byte

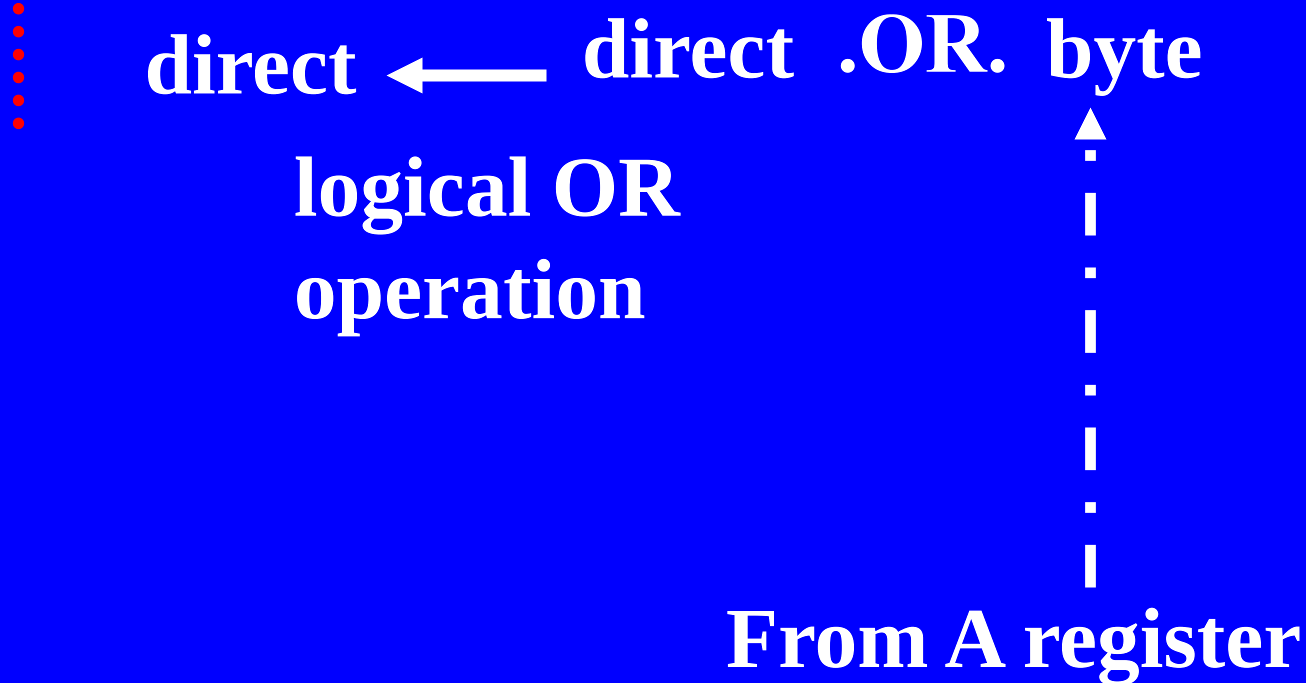
logical OR operation

8051

From *Rn* or *direct* or *indirect* or *immediate*

ORL direct, A;

No Flags at PSW affect



direct ← direct .OR. byte

logical OR operation

From A register

8051

ORL direct, #data;

No Flags at PSW affect

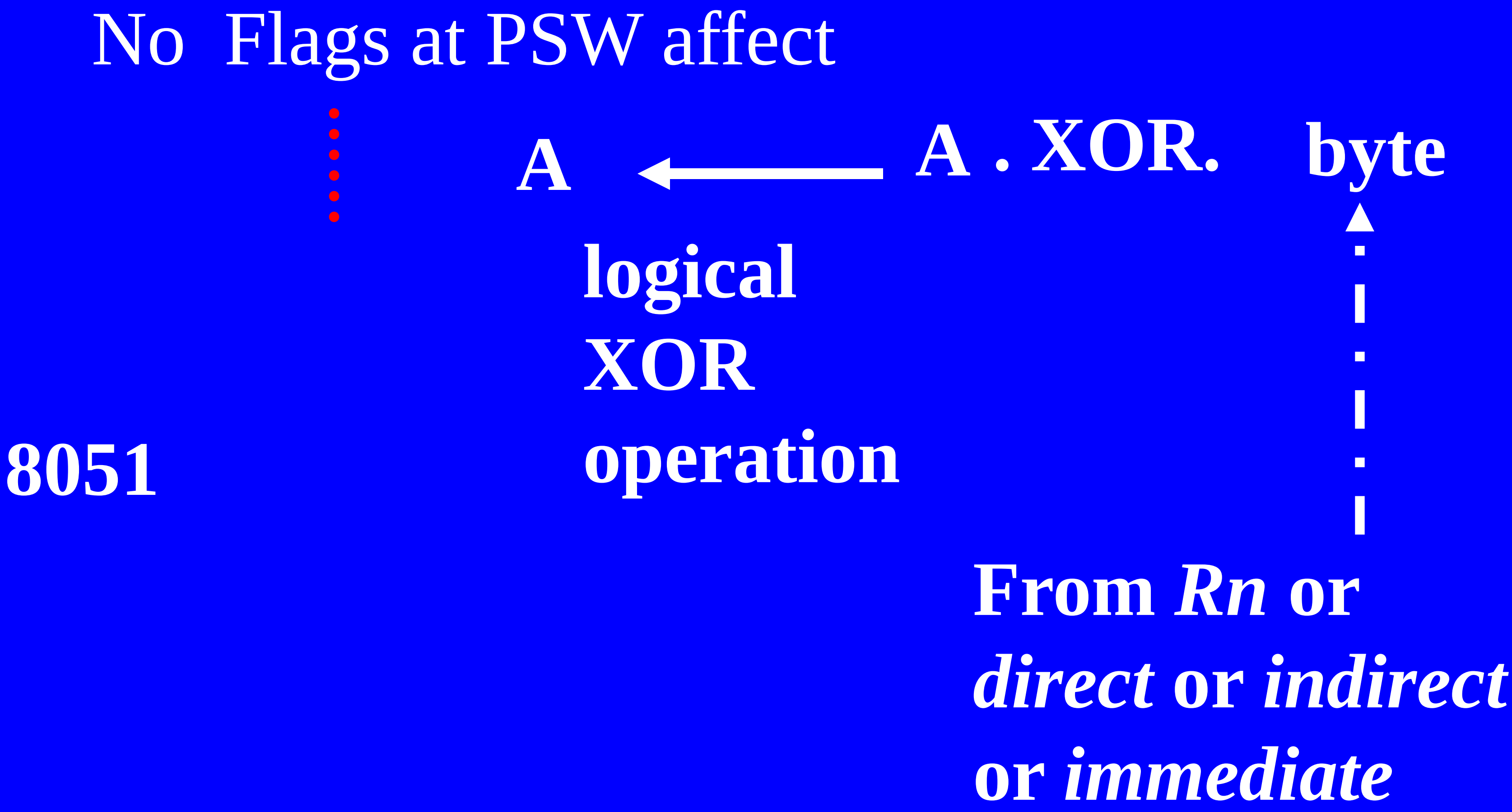
direct ← direct .OR. byte

logical OR operation

8051

Immediate (third byte of instruction)

XRL A, @Ri;
XRL A, Rn;
XRL A, direct;
XRL A, #data;



No Flags at PSW affect

A ← A . XOR. byte

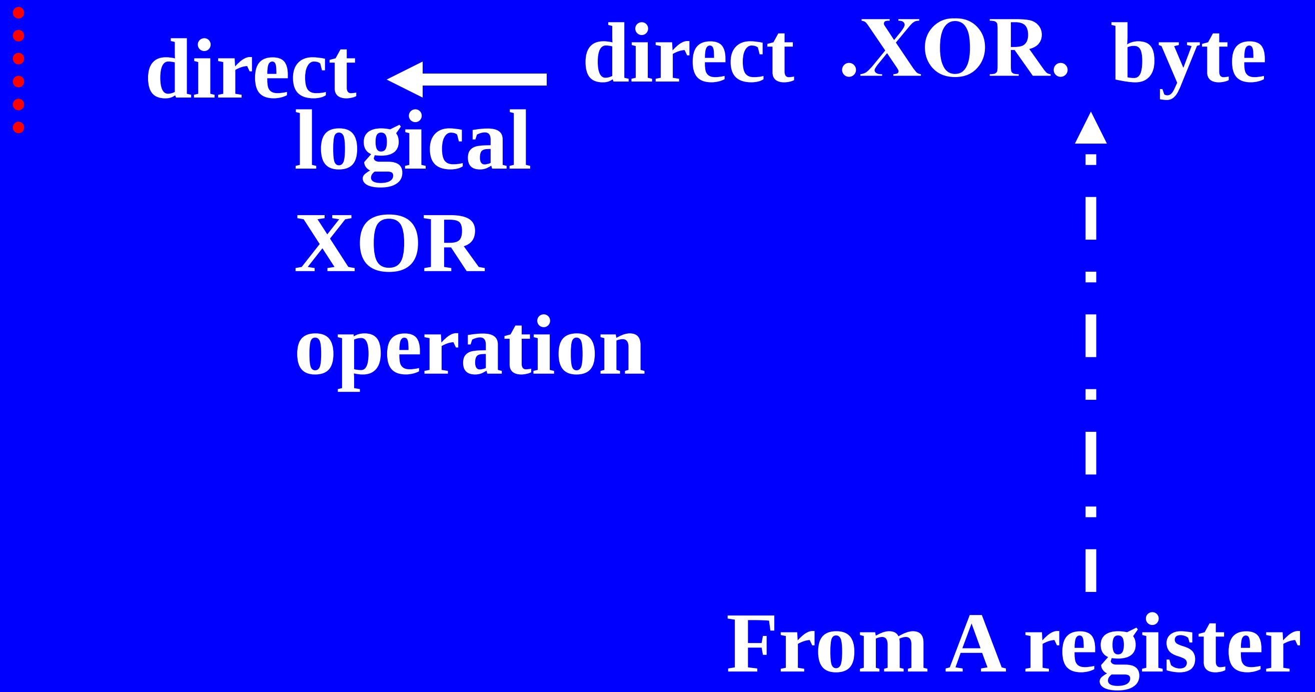
logical XOR operation

8051

From *Rn* or *direct* or *indirect* or *immediate*

XRL direct, A;

No Flags at PSW affect



direct ← direct .XOR. byte

logical XOR operation

From A register

8051

XRL direct, #data;

No Flags at PSW affect

direct ← direct .XOR. byte

logical XOR operation

8051

Immediate (third byte of instruction)

Summary

We learnt 8051 family processing instructions

- clear, complement, set bit
- clear and complement A
- Arithmetic and logical instructions
- Flags do not affect in logic operations and increment-decrement
- No instruction for subtract without borrow