

(1) Logical Instructions

- **ANA R**
- **ANA M**

- **AND specified data in register or memory with accumulator.**
- **Store the result in accumulator (A).**

- **Example: ANA B, ANA M**

BEFORE EXECUTION

CY		AC	
----	--	----	--

A	AA		
B	0F	C	
D		E	
H		L	

1010 1010=AAH

0000 1111=0FH

0000 1010=0AH

ANA B

A=A and R

AFTER EXECUTION

CY	0	AC	1
----	---	----	---

A	0A		
B	0F	C	
D		E	
H		L	

BEFORE EXECUTION

CY		AC		
A	55			2050H
H	20	L	50	

0101 0101=55H

1011 0011=B3H

0001 0001=11H

ANA M

A=A and M

AFTER EXECUTION

CY	0	AC	1	
A	11			2050H
H	20	L	50	

(2) Logical Instructions

- **ANI 8-bit data**
- **AND 8-bit data with accumulator (A).**
- **Store the result in accumulator (A)**
- **Example: ANI 3FH**

BEFORE EXECUTION

AFTER EXECUTION

1011 0011=B3H

0011 1111=3FH

0011 0011=33H

CY		AC	
----	--	----	--

A	B3
---	----

ANI 3FH

A=A and DATA(8)

CY	0	AC	1
----	---	----	---

A	33
---	----

(3) Logical Instructions

- **XRA Register (8-bit)**
- **XOR specified register with accumulator.**
- **Store the result in accumulator.**
- **Example: XRA C**

BEFORE EXECUTION

1010 1010=AAH

0010 1101=2DH

1000 0111=87H

AFTER EXECUTION

CY		AC	
----	--	----	--

A	AA		
B		C	2D
D		E	
H		L	

CY	0	AC	0
----	---	----	---

A	87		
B		C	2D
D		E	
H		L	

XRA C
A=A xor R

(4) Logical Instructions

- **XRA M**
- **XOR data in memory (memory location pointed by H-L pair) with Accumulator.**
- **Store the result in Accumulator.**
- **Example: XRA M**

BEFORE EXECUTION

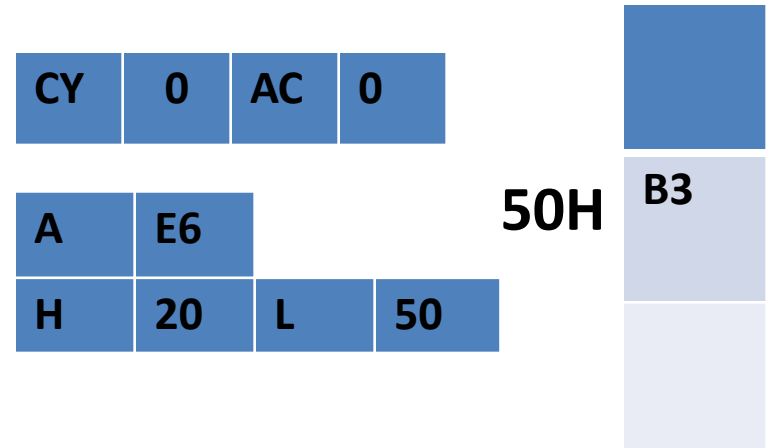
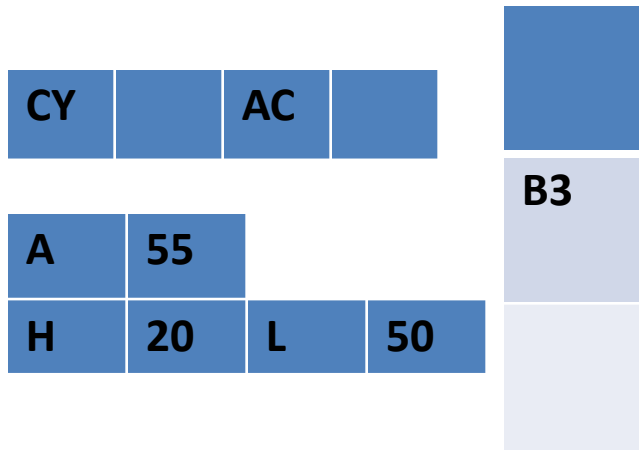
0101 0101=55H

1011 0011=B3H

1110 0110=E6H

AFTER EXECUTION

XRA M
A=A xor M



(5) Logical Instructions

- **XRI 8-bit data**
- **XOR 8-bit immediate data with accumulator (A).**
- **Store the result in accumulator.**
- **Example: XRI 39H**

1011 0011=B3H

0011 1001=39H

BEFORE EXECUTION

1000 1010=8AH

AFTER EXECUTION



XRI 39H

A=A xor DATA(8)



(6) Logical Instructions

- **ORA Register**

- **OR specified register with accumulator (A).**
- **Store the result in accumulator.**

- **Example: ORA B**

BEFORE EXECUTION

1010 1010=AAH

0001 0010=12H

1011 1010=BAH

AFTER EXECUTION

CY		AC	
----	--	----	--

CY	0	AC	0
----	---	----	---

ORA B
A=A or R

A	AA		
B	12	C	
D		E	
H		L	

A	BA		
B	12	C	
D		E	
H		L	

(7) Logical Instructions

- **ORA M**

- **OR specified register with accumulator (A).**
- **Store the result in accumulator.**

- **Example: ORA M**

BEFORE EXECUTION

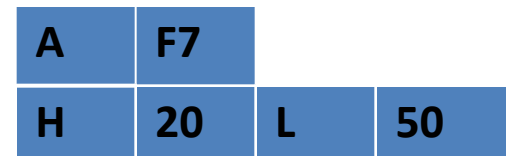
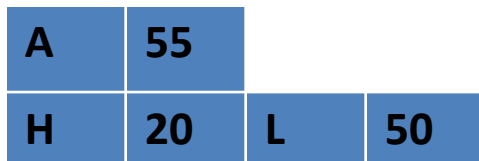
0101 0101=55H

1011 0011=B3H

1111 0111=F7H

AFTER EXECUTION

ORA M
A=A or M



(8) Logical Instructions

- **ORI 8-bit data**

- **OR 8-bit data with accumulator (A).**
- **Store the result in accumulator.**

- **Example: ORI 08H**

1011 0011=B3H

0000 1000=08H

BEFORE EXECUTION

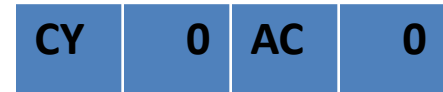
1011 1011=BBH

AFTER EXECUTION



ORI 08H

A=A or DATA(8)



(9) Logical Instructions

- **CMP Register**
- **CMP M**

- Compare specified data in register or memory with accumulator (A).
- Example: CMP D or CMP M
- The result of comparison is shown by setting the flags as follows:
- If $(A) < \text{reg/memory}$: carry flag =1.
- If $(A) > \text{reg/memory}$: carry flag and zero flag are reset.
- If $(A) = \text{reg/memory}$: zero flag is set.
- No. of byte: 1-byte

BEFORE EXECUTION

CY		Z	
----	--	---	--

A	B8		
B		C	
D	B9	E	
H		L	

A>R: CY=0,Z=0
 A=R: CY=0,Z=1
 A<R: CY=1,Z=0

CMP D
A-R

AFTER EXECUTION

CY	1	Z	0
----	---	---	---

A	B8		
B		C	
D	B9	E	
H		L	

BEFORE EXECUTION

CY		Z		
A	B8		2050H	B8
H	20	L	50	

A>M: CY=0,Z=0
 A=M: CY=0,Z=1
 A<M: CY=1,Z=0

CMP M
A-M

AFTER EXECUTION

CY	0	Z	1	
A	B8		2050H	B8
H	20	L	50	

(10) Logical Instructions

- **CPI 8-bit data**
- **Compare 8-bit immediate data with accumulator (A).**
- **Example: CPI 30H**

BEFORE EXECUTION

A>DATA: CY=0,Z=0

A=DATA: CY=0,Z=1

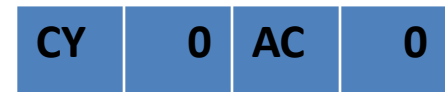
A<DATA: CY=1,Z=0

AFTER EXECUTION



CPI 30H

A-DATA



1011 1010=BAH

(11) Logical Instructions

- **STC**
- **It sets the carry flag to 1.**
- **Example: STC**

BEFORE EXECUTION

CY	0
----	---

STC
CY=1

AFTER EXECUTION

CY	1
----	---

(12) Logical Instructions

- **CMC**

- **It complements the carry flag.**
- **Example: CMC**

BEFORE EXECUTION

AFTER EXECUTION

CY	1
----	---

CMC

CY	0
----	---

(13) Logical Instructions

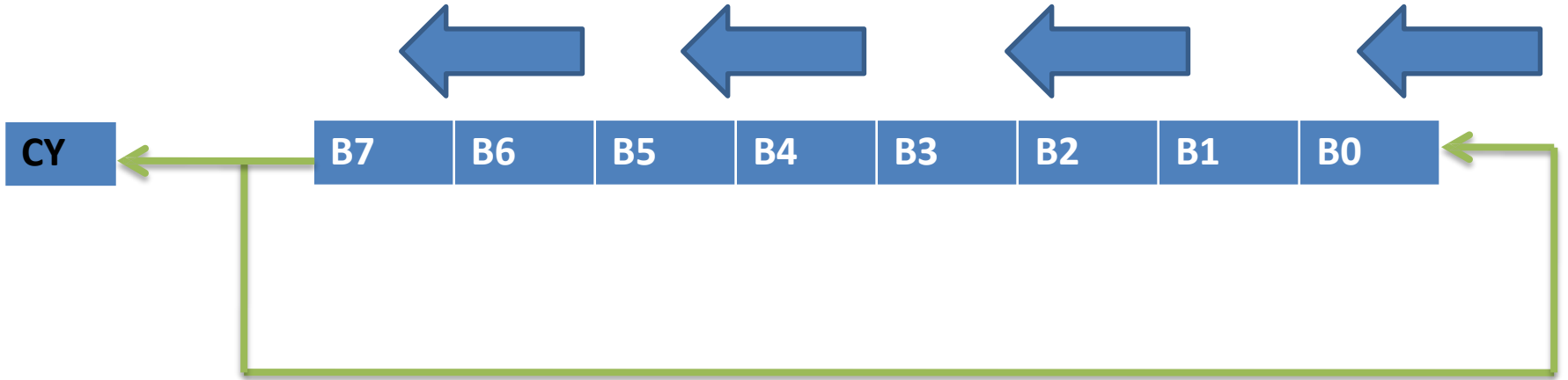
- **CMA**

- **It complements each bit of the accumulator.**
- **Example: CMA**

(14) Logical Instructions

- **RLC**
- **Rotate accumulator left**
- **Each binary bit of the accumulator is rotated left by one position.**
- **Bit D7 is placed in the position of D0 as well as in the Carry flag.**
- **CY is modified according to bit D7.**
- **Example: RLC.**

BEFORE EXECUTION



AFTER EXECUTION



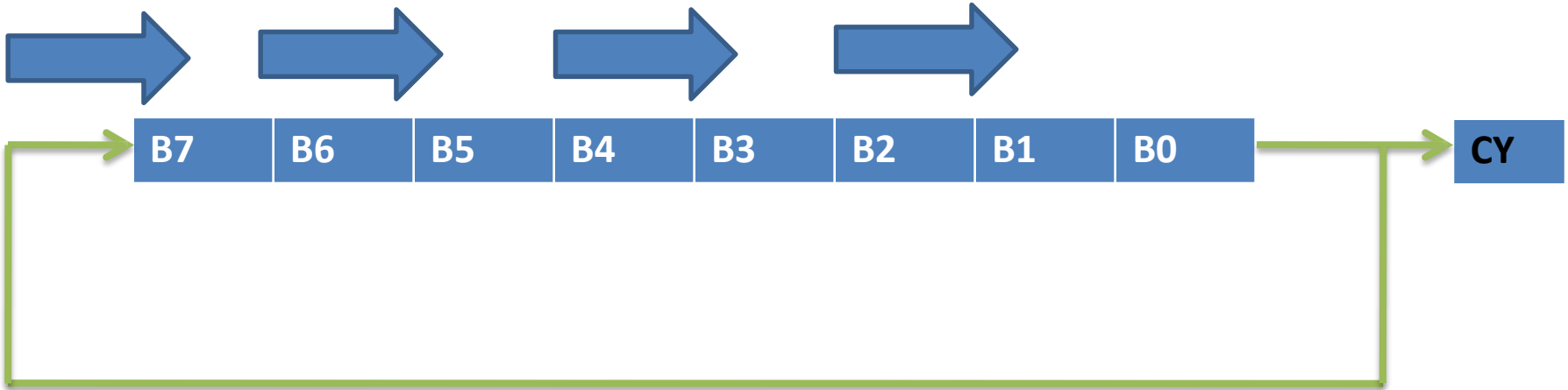
(15) Logical Instructions

- **RRC**

- Rotate accumulator right
- Each binary bit of the accumulator is rotated right by one
- position.
- Bit D0 is placed in the position of D7 as well as in the Carry flag.
- CY is modified according to bit D0.

- Example: RRC.

BEFORE EXECUTION



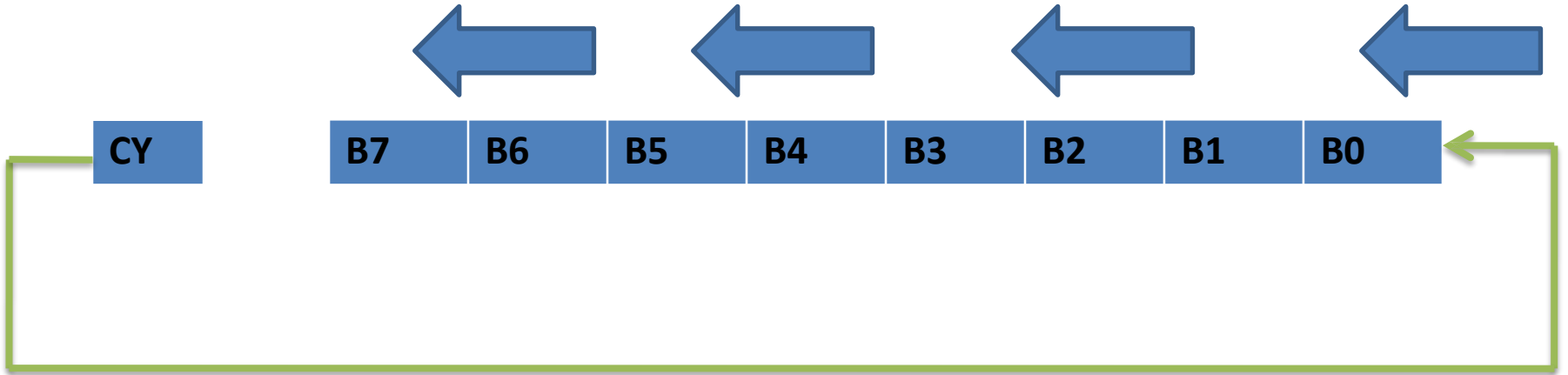
AFTER EXECUTION



(16) Logical Instructions

- **RAL**
- **Rotate accumulator left through carry**
- **Each binary bit of the accumulator is rotated left by one position through the Carry flag.**
- **Bit D7 is placed in the Carry flag, and the Carry flag is placed in the least significant position D0.**
- **CY is modified according to bit D7.**
- **Example: RAL.**

BEFORE EXECUTION



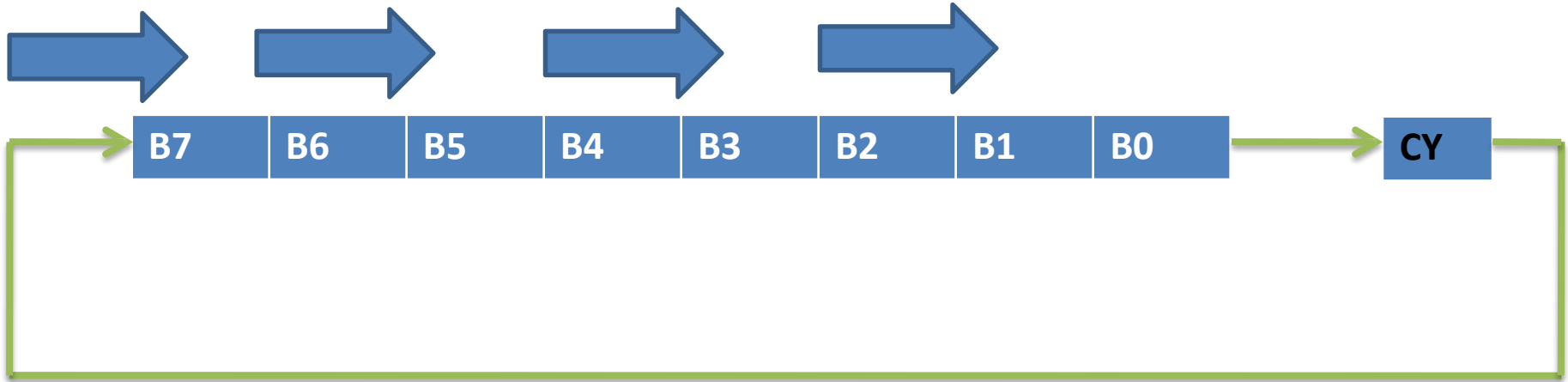
AFTER EXECUTION



(17) Logical Instructions

- **RAR**
- **Rotate accumulator right through carry**
- **Each binary bit of the accumulator is rotated left by one position through the Carry flag.**
- **Bit D7 is placed in the Carry flag, and the Carry flag is placed in the least significant position D0.**
- **CY is modified according to bit D7.**
- **Example: RAR**

BEFORE EXECUTION



AFTER EXECUTION

