

TSEA83 : Datorkonstruktion

Fö2

Mikroprogrammering 1

Fö2 : Agenda

- Att bygga en CPU
 - Mikroprogrammering vs Pipelining
- Mikroprogrammering
 - Grundläggande byggstenar
- Mikromaskinen
 - ”Olle Roos”-datorn
- Mikrokod
 - Ett mindre program
 - Ytterligare exempel för olika instruktioner och adresseringsmoder

Att bygga en CPU

Mikroprogrammering vs Pipelining

Att bygga en CPU

Mikroprogrammerad dator (fö2+3)

Björn Lindskog-datorn (lab1)

- Variabel exekveringstid
- Variabelt format
- Inget överlapp
- Central styrenhet, som är mikroprogrammerad
- Flera adresseringsmoder/instruktion
- 1 ackumulator
- Nästan alla instruktioner har operand i minnet:
LDA Adr ; $A=M(\text{Adr})$
ADDA Adr ; $A=A+M(\text{Adr})$

Pipelinaad dator (fö4,lab2)

- Alla instruktioner tar 5 CK
- Alla instruktioner har samma format
- Pipelining/överlapp ger 1 färdig instruktion/CK
- Flera avkodare (inget mprog)
- 1 a-mod/instruktion
- 32 register
- Endast LD/ST har operand i minnet:
LD Rd,(Ra) ; $Rd=M(\text{Ra})$
ADD Rd,Ra,Rb ; $Rd=Ra+Rb$

Att bygga en CPU

Mikroprogrammerad dator (fö2+3)

Björn Lindskog-datorn (lab1)

Pipelinaad dator (fö4,lab2)

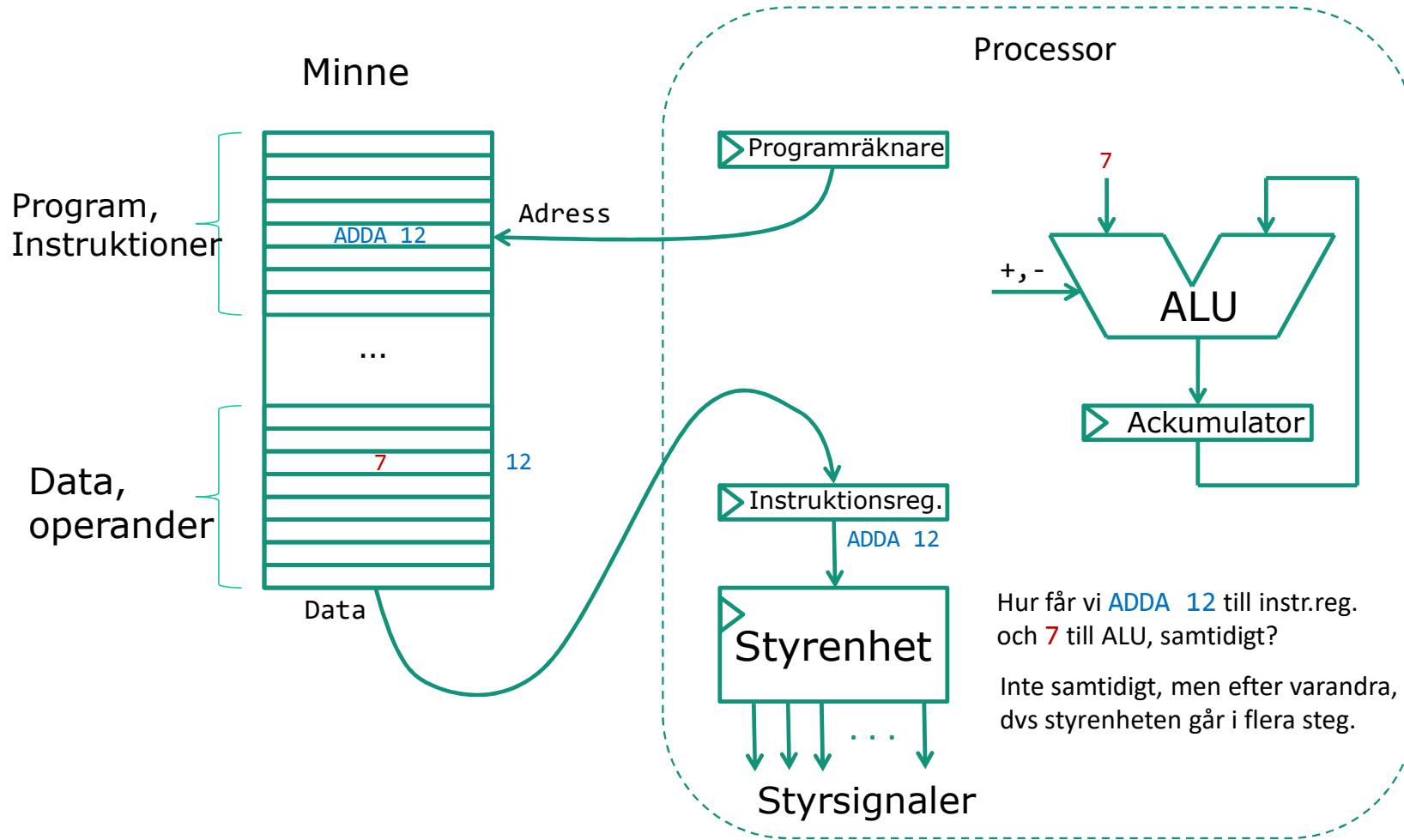
- Typisk CISC
- Programmering på 2 nivåer
asm och mikro
- Enkel controller:
 - garageportsöppnare
 - del av dator
- + Man kan göra avancerade instruktioner: **sortera**
- - Det blir många klockcykler / instr.

- Typisk RISC
- Programmering på 1 nivå:
asm
- Enkel CPU
 - enkel mobil
- Finns bara enkla instr.
- Snabb

Mikroprogrammering

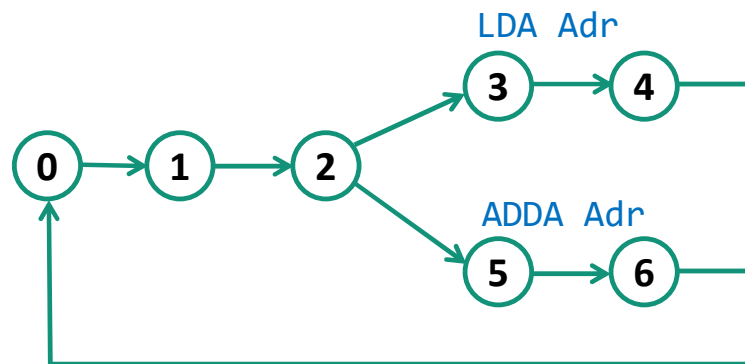
Grundläggande byggstenar

Ritning 1



Mikroprogrammering

- Vi ska bygga en liten dator med enkla komponenter
- Styrenheten (SekvensNät) visar sig vara svårast. Hur gör man för att konstruera ett SN med 100+ tillstånd?
- Mikroprogrammering är en vidareutveckling (och faktiskt begränsning) av tekniken att bygga sekvensnät med ROM
- Idé: byt tillståndsvipporna mot en universalräknare



Tänkbara händelser vid olika steg:

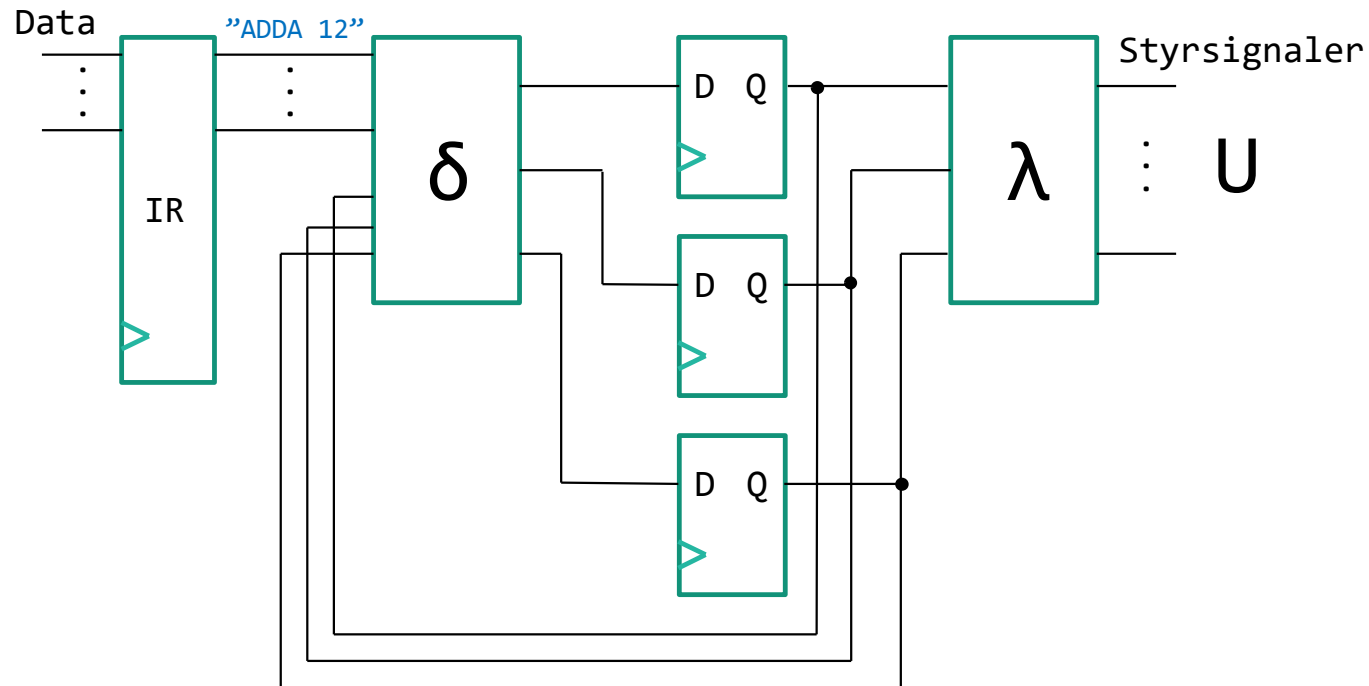
Steg 0-2: Hämta instruktion,
samt beräkna adress

Steg 3-4: Utför en instruktion

Steg 5-6: Utför en annan
instruktion

Mikroprogrammering

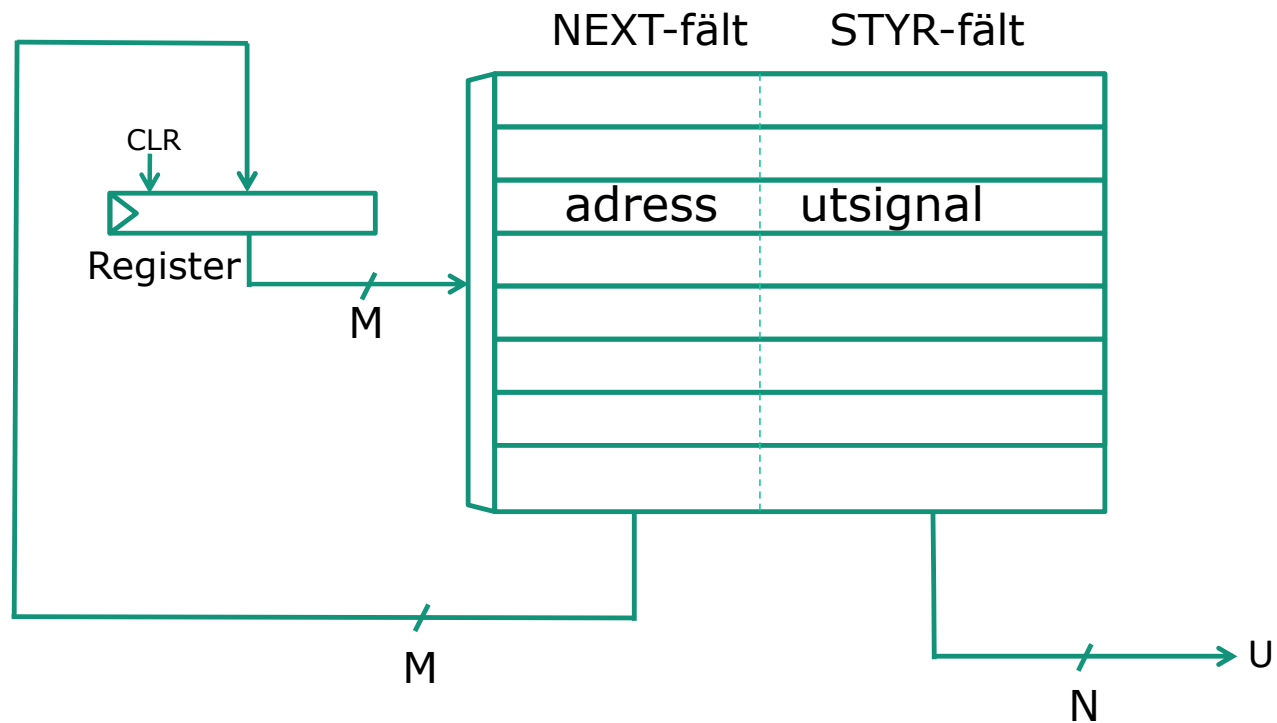
Vi skulle kunna bygga sekvensnätet på "vanligt" sätt:



Men det blir väldigt jobbigt för 100+ tillstånd,
dessutom omständligt att konfigurera om vid behov.

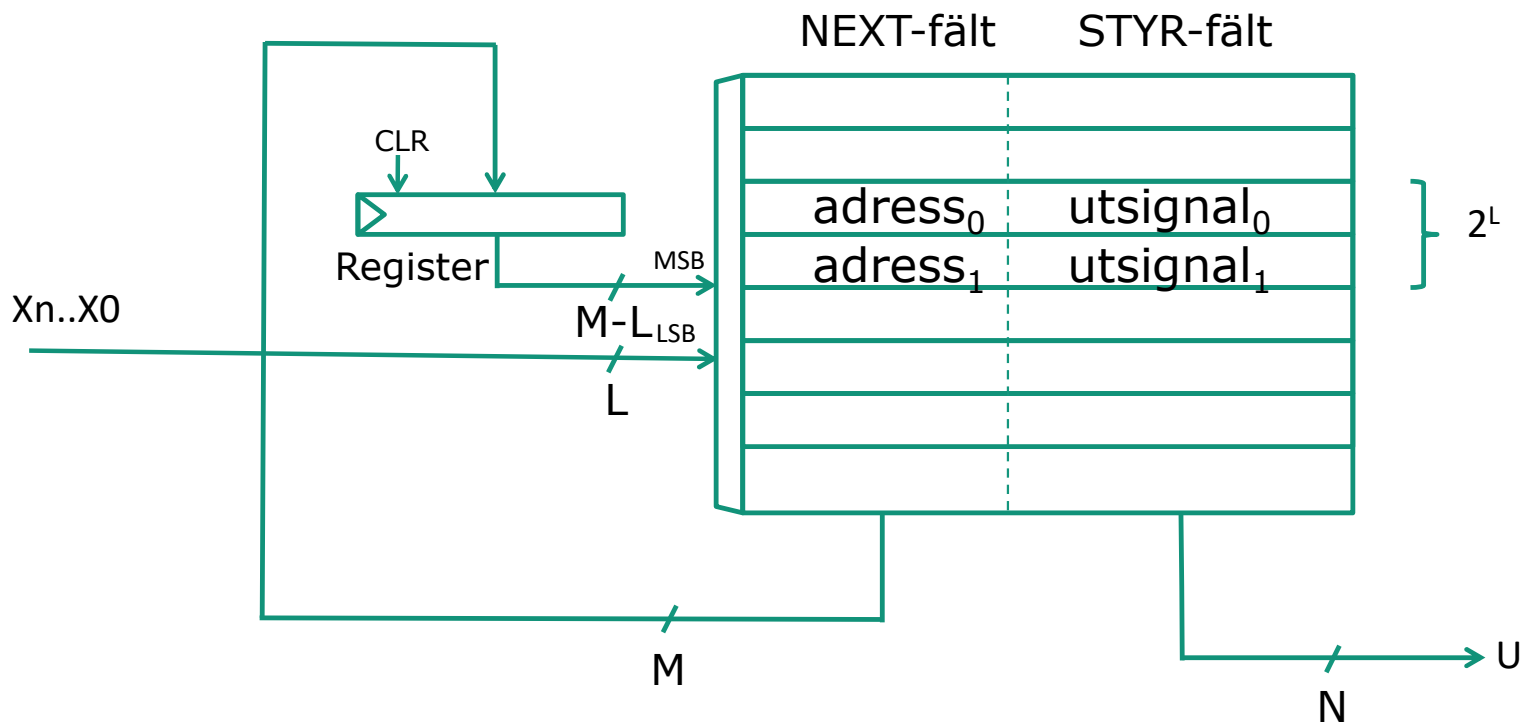
Ritning 2

Autonom styrenhet med ROM/Register



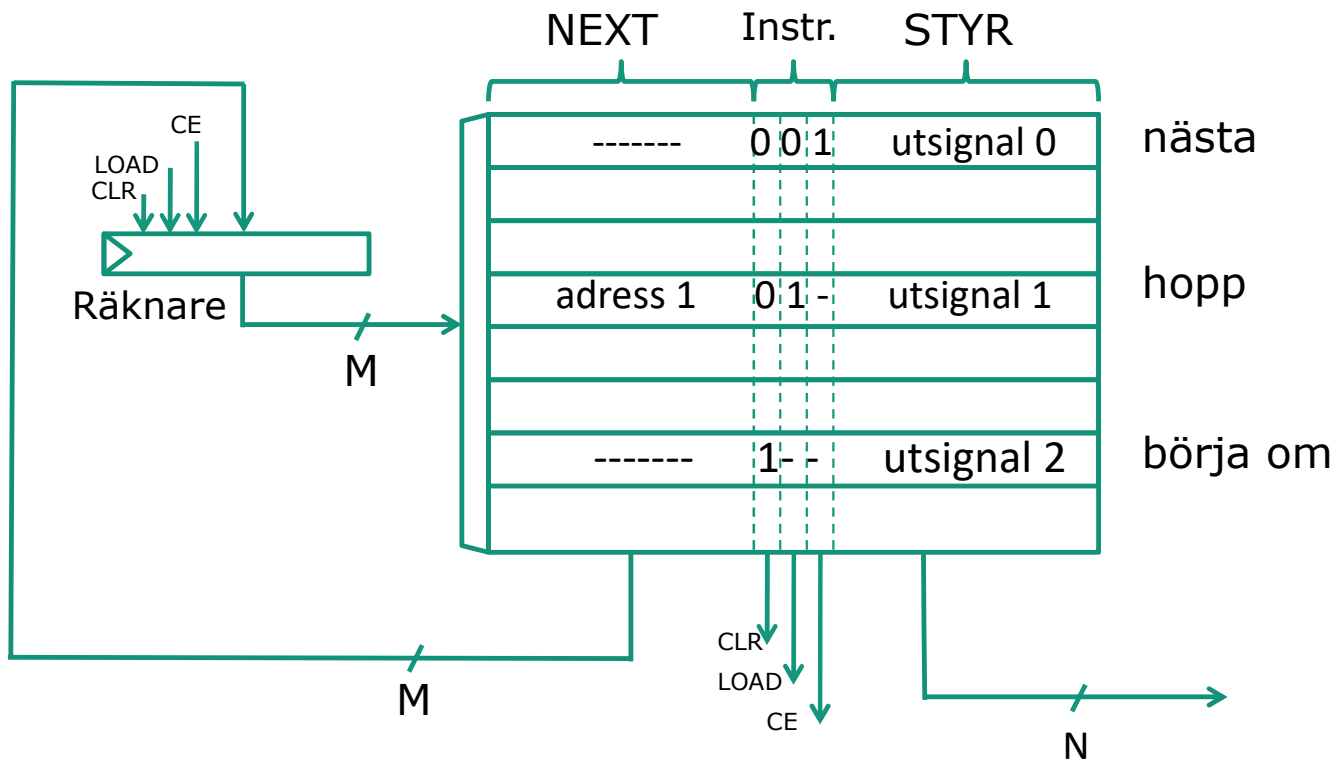
Variant 2

Autonom styrenhet med ROM/Register



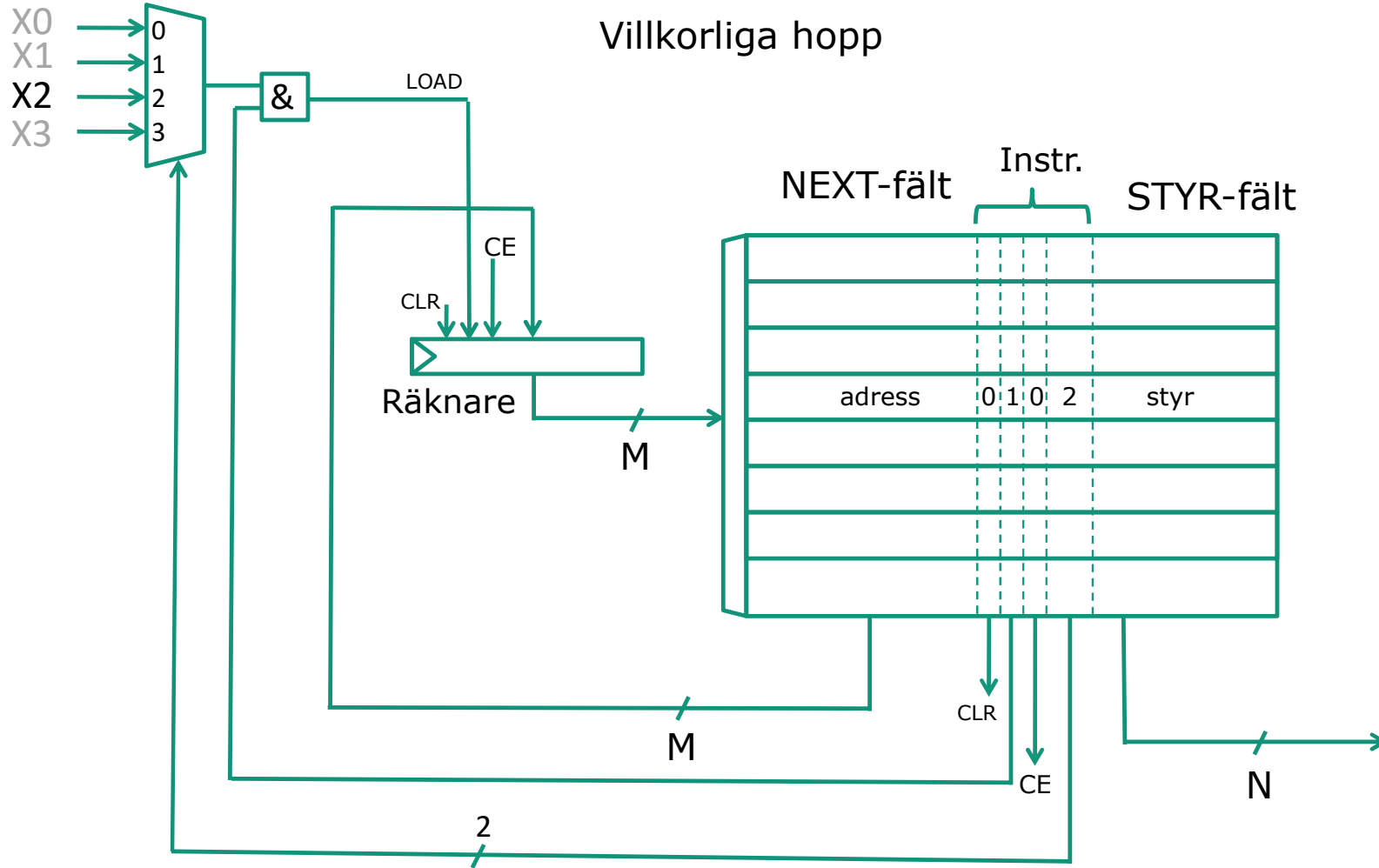
Ritning 3

Autonom styrenhet med ROM/Räknare



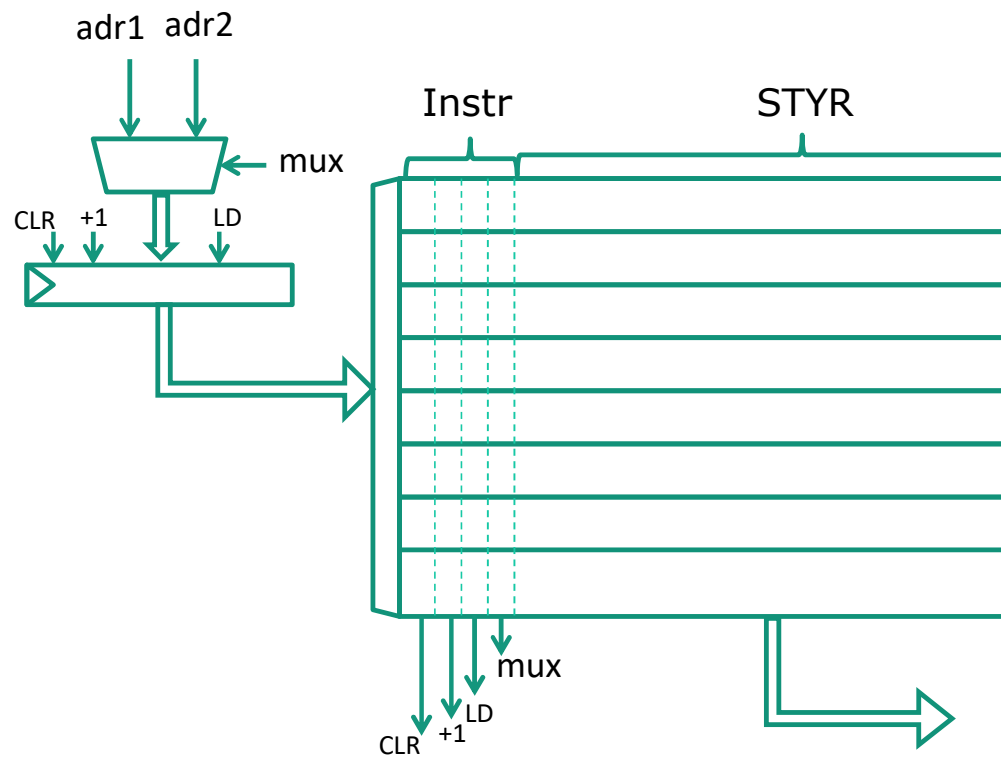
Ritning 4

Villkorliga hopp

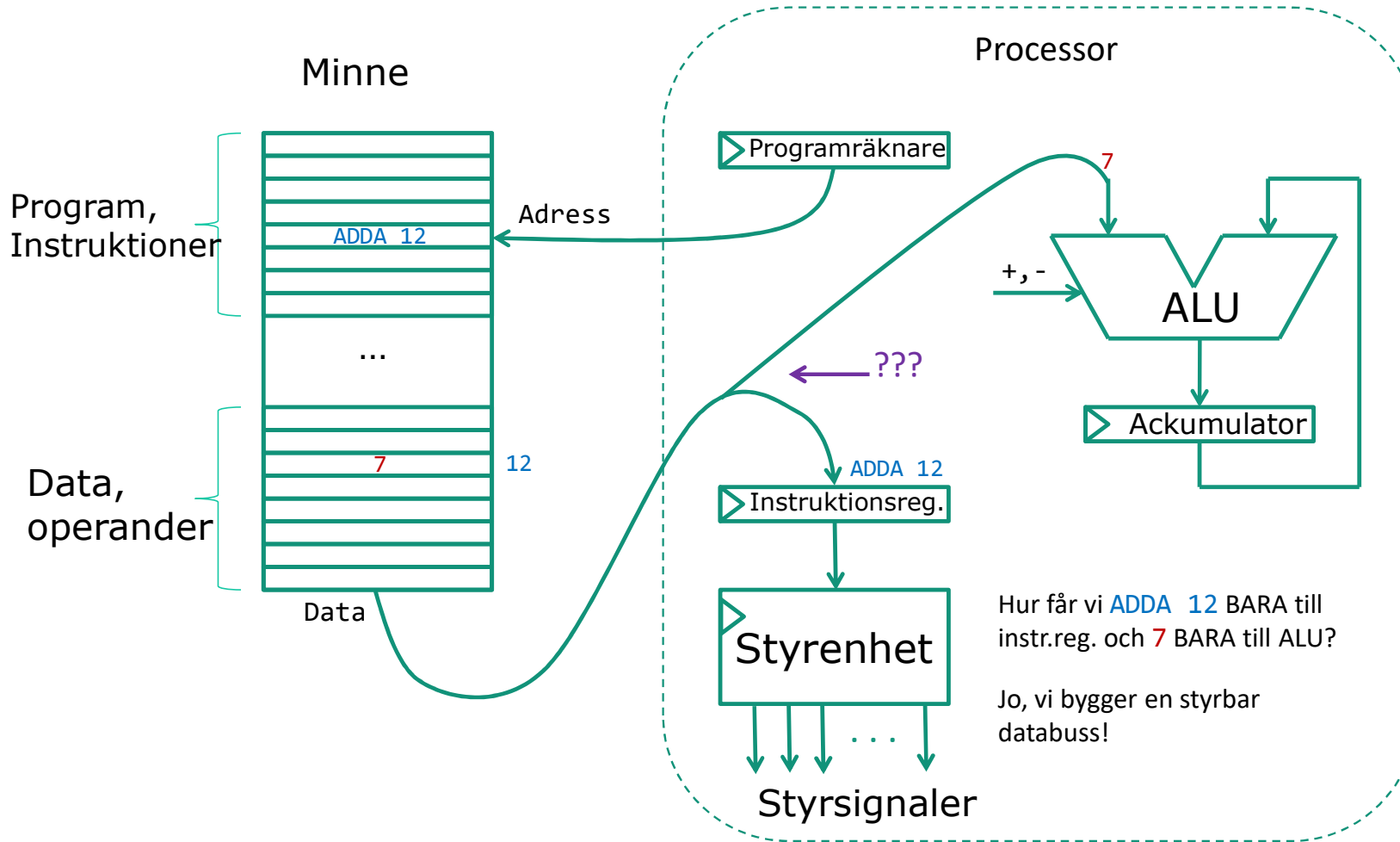


En variant

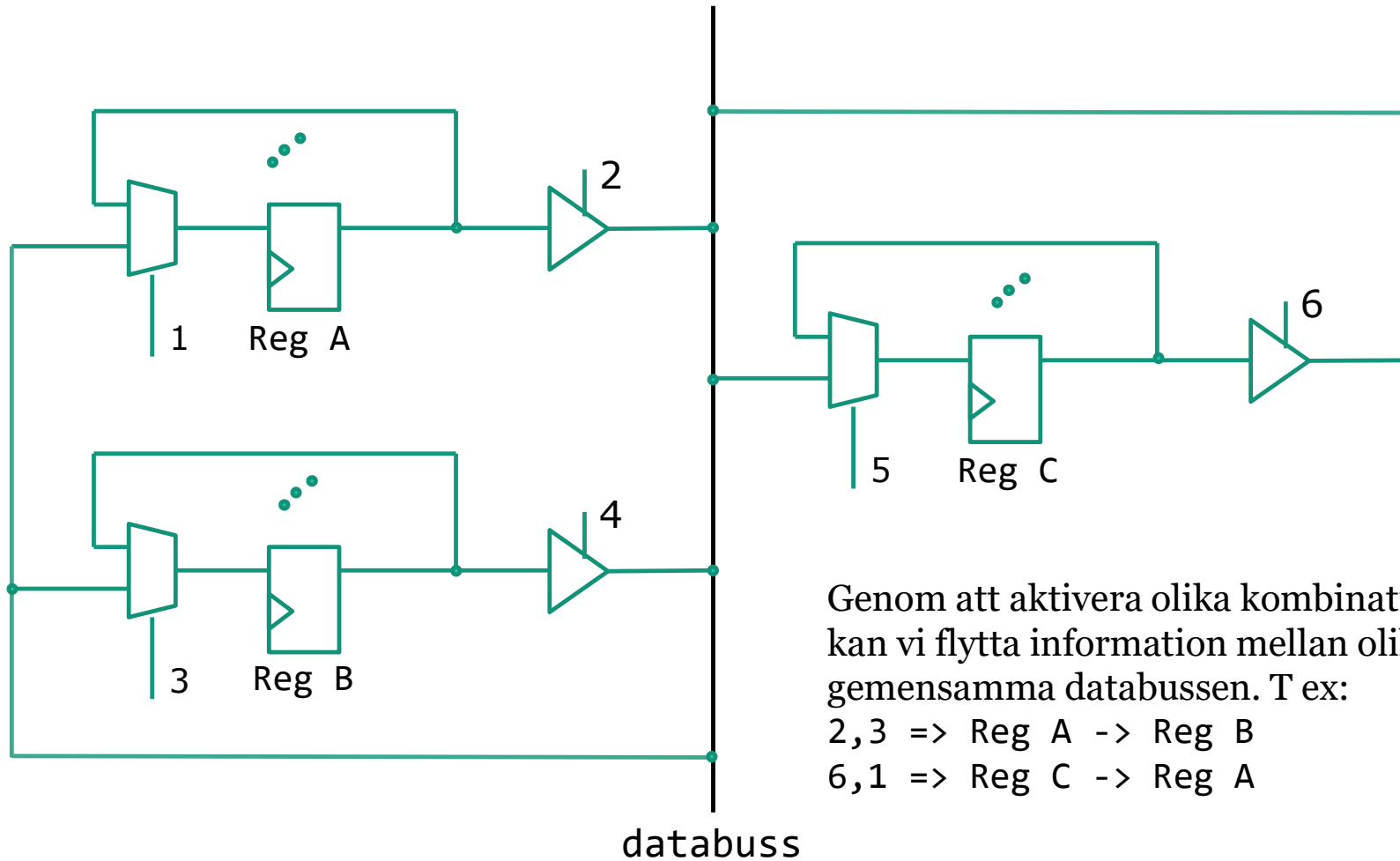
Autonom styrenhet utan adressfält



Databussen då?



Datorkonstruktion Ritning 5



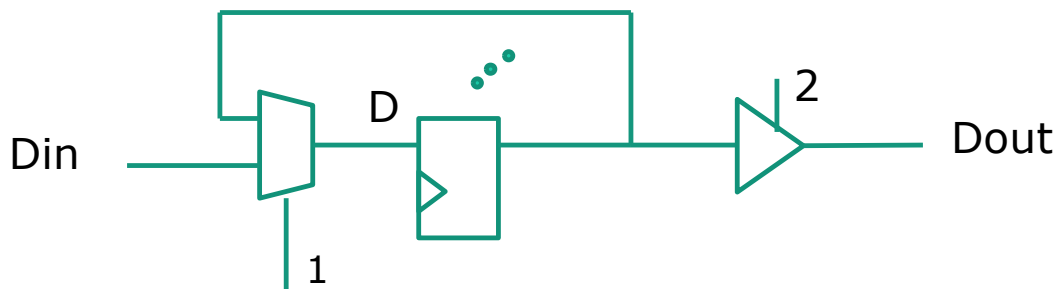
Genom att aktivera olika kombinationer av styrsignaler kan vi flytta information mellan olika register via den gemensamma databussen. T ex:

2,3 => Reg A -> Reg B

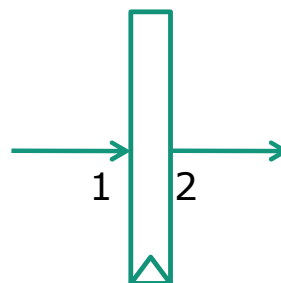
6,1 => Reg C -> Reg A

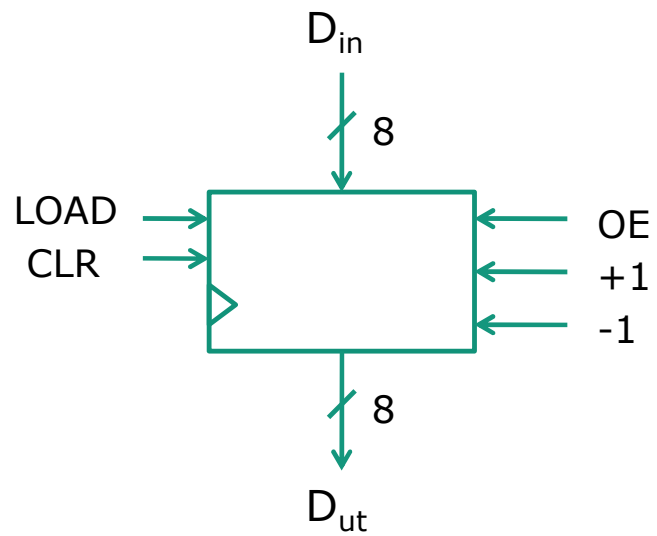
Register

Det som egentligen ser ut så här =>

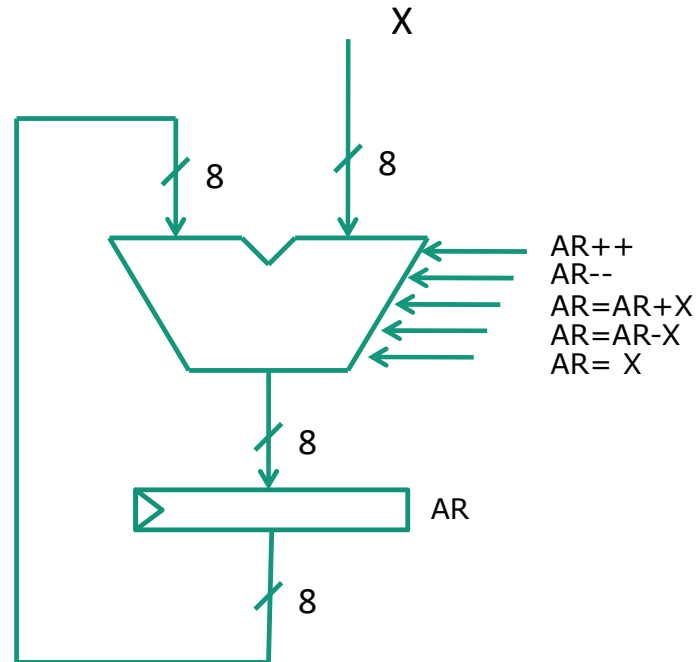


Förenklar vi genom att rita så här =>



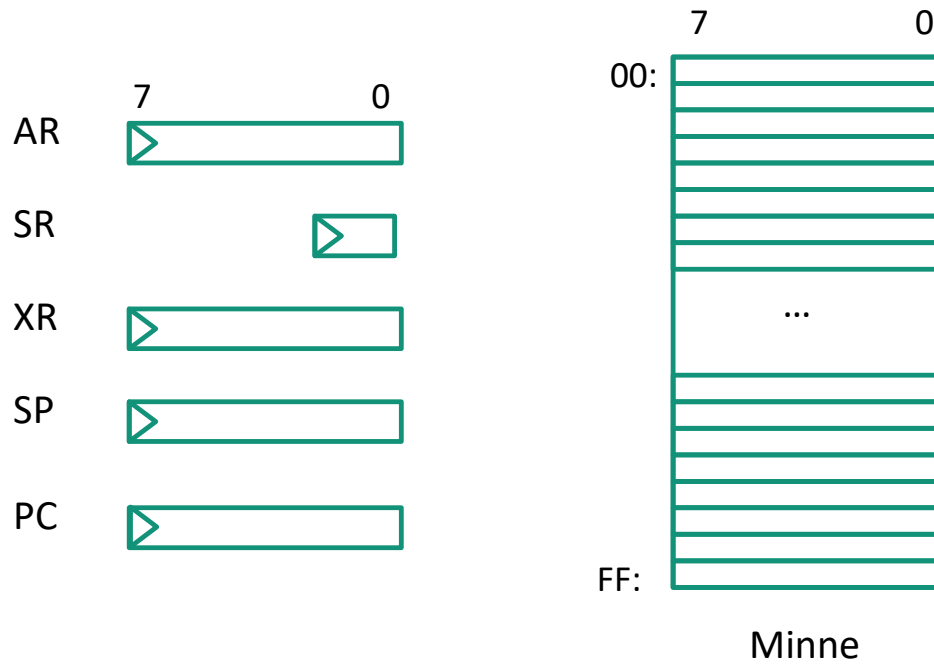


universalsräknare



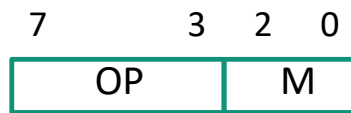
ALU/ackumulator

Ritning 8 - Programmerarmodell



Endast 2 flaggor: Z,N

Ritning 9 - Instruktionsformat

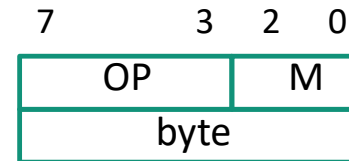


Opkod

32 instruktioner

Märkfält

8 adresseringsmoder



Adress/data

Adresseringsmoder

M	Mod	Exempel	EA	
000	Absolut	LDA <i>addr</i>	<i>addr</i>	M(addr) ->AR
001	Indirekt	LDA (<i>addr</i>)	M(<i>addr</i>)	M(M(addr)) ->AR
010	Indexerad	LDA <i>disp</i> , (XR)	XR + <i>disp</i>	M(XR+disp) ->AR
011	Relativ	JMP <i>disp</i>	PC + 2 + <i>disp</i>	PC+2+disp->PC
100	Omedelbar	LDA # <i>n</i>	PC + 1	n->AR
101	Underförstådd	INCA/INC	—	

Exempelvis:

LDA 3

0:	LDA	000
1:	3	
2:		
3:	5	

Absolut
EA=3 (Effektiv Adress)
Operanden = 5

Instruktion	Verkan	Status			
		N	Z	C	V
LDA <i>addr</i>	$AR := M(addr)$	*	*	-	0
STA <i>addr</i>	$M(addr) := AR$	-	-	-	0
ADD <i>addr</i>	$AR := AR + M(addr)$	*	*	*	*
SUB <i>addr</i>	$AR := AR - M(addr)$	*	*	*	*
INCA	$AR := AR + 1$	*	*	*	*
DEC	$AR := AR - 1$	*	*	*	*
CMP <i>addr</i>	$AR - M(addr)$	*	*	*	*
CLRA	$AR := 0$	0	1	0	0
ASRA	$AR := AR/2$	*	*	*	-
ASLA	$AR := AR \cdot 2$	*	*	*	*
LSRA	logiskt högerskift av AR	0	*	*	-
AND <i>addr</i>	$AR := AR \text{ and } M(addr)$	*	*	-	0
OR <i>addr</i>	$AR := AR \text{ or } M(addr)$	*	*	-	0
JMP <i>addr</i>	$PC := addr$	-	-	-	-
JMPN <i>addr</i>	$PC := addr$ om $N = 1$	-	-	-	-
JMPZ <i>addr</i>	$PC := addr$ om $Z = 1$	-	-	-	-
JMPC <i>addr</i>	$PC := addr$ om $C = 1$	-	-	-	-
JMPV <i>addr</i>	$PC := addr$ om $V = 1$	-	-	-	-
IN	$AR := IN$	*	*	-	0
OUT	$UT := AR$	-	-	-	0

Mikromaskinen

”Olle-roos”-datorn

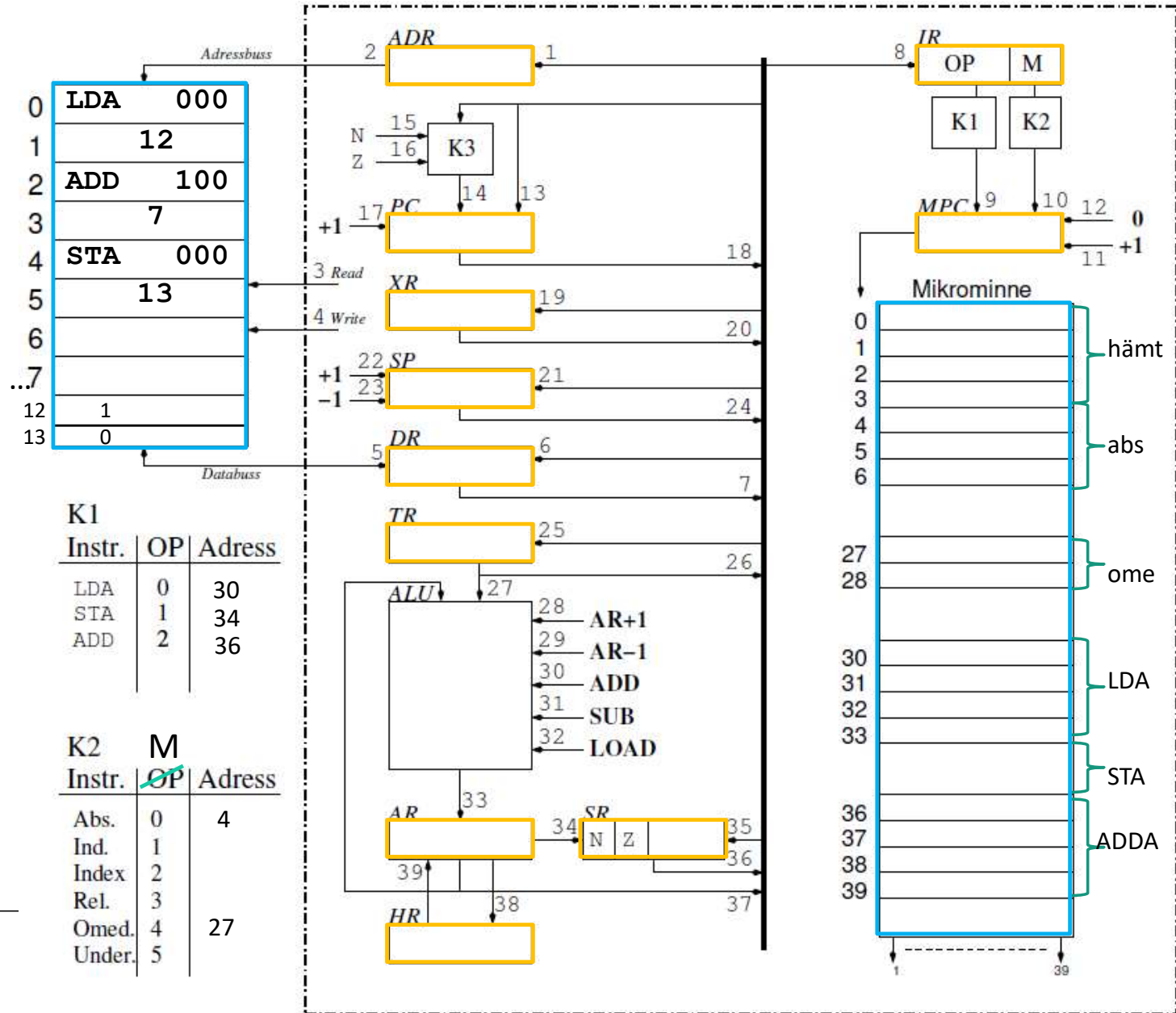
Mikromaskinen

”Olle Roos – datorn”

- = register
- = minne
- = kombinatorik

Tabellerna K1, K2 och K3 kan även implementeras som minnen.

RESET ●



Normal arbetsgång - översikt

För varje instruktion {

1. Hämtfas => Samma för alla instruktioner

1. Hämta instruktionen till IR
2. PC++
3. Hoppa till rätt ...

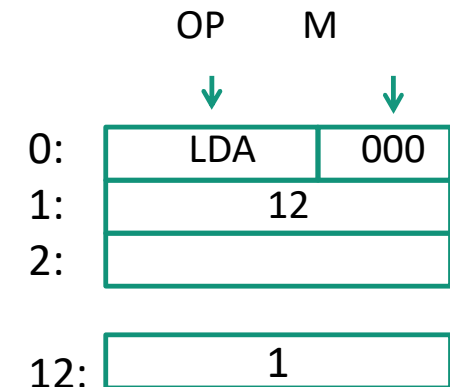
2. Adresseringsmodsfas Beroende på M sker olika saker

1. Vanligen: Hämta byten, PC++
2. EA till ADR
3. Hoppa till rätt ...

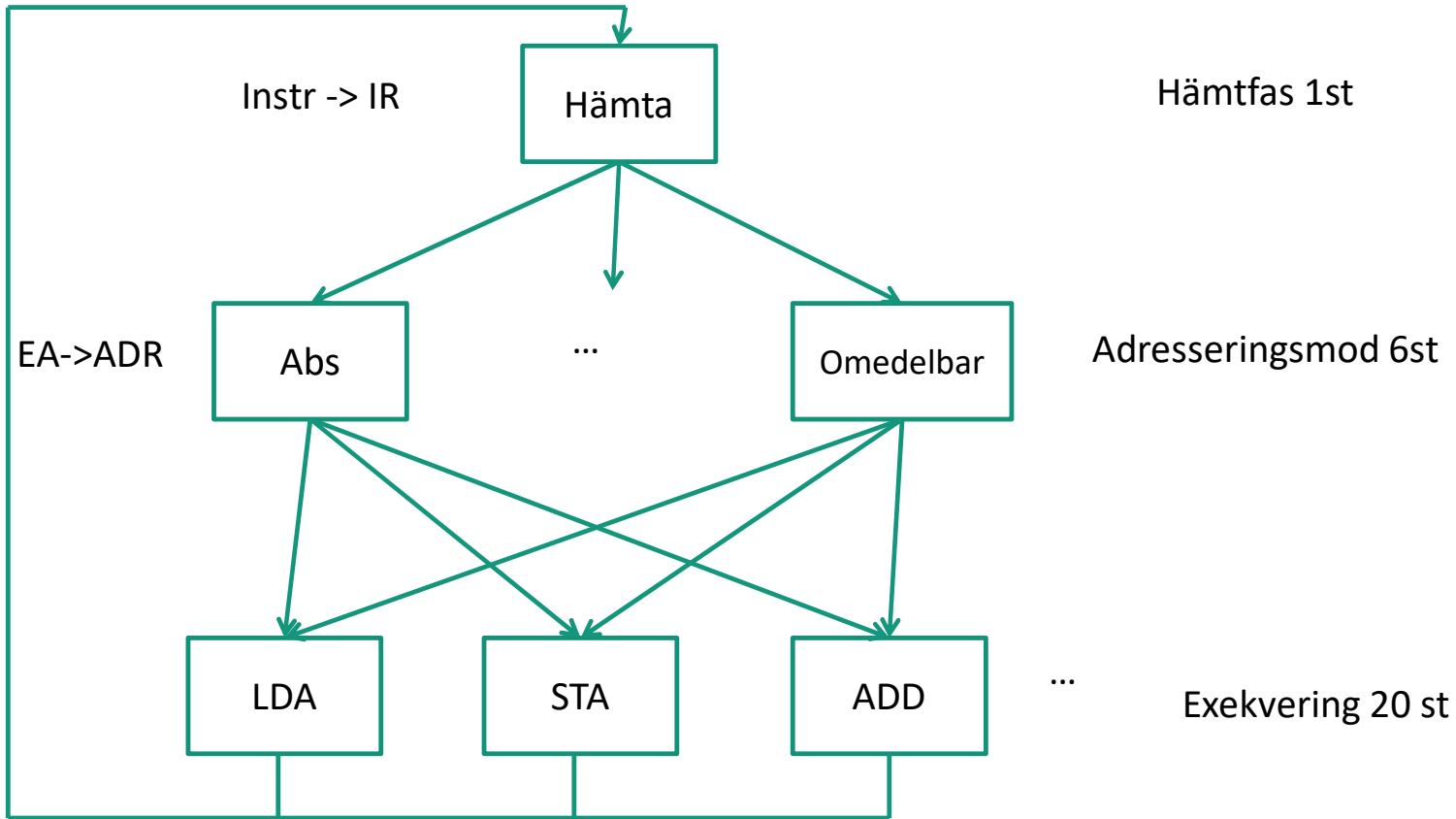
3. Exekveringsfas => Beroende på OP sker olika saker

1. Vanligen: Hämta operanden
2. Resultatet till AR och uppdatera SR
3. Hoppa till Hämtfas

}



Organisation av mikroprogram

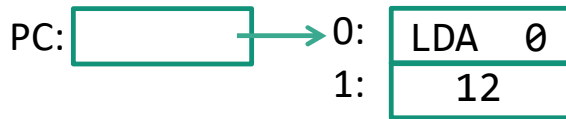


Mikrokod

- Ett mindre program
- Ytterligare exempel för olika instruktioner och adresseringsmoder

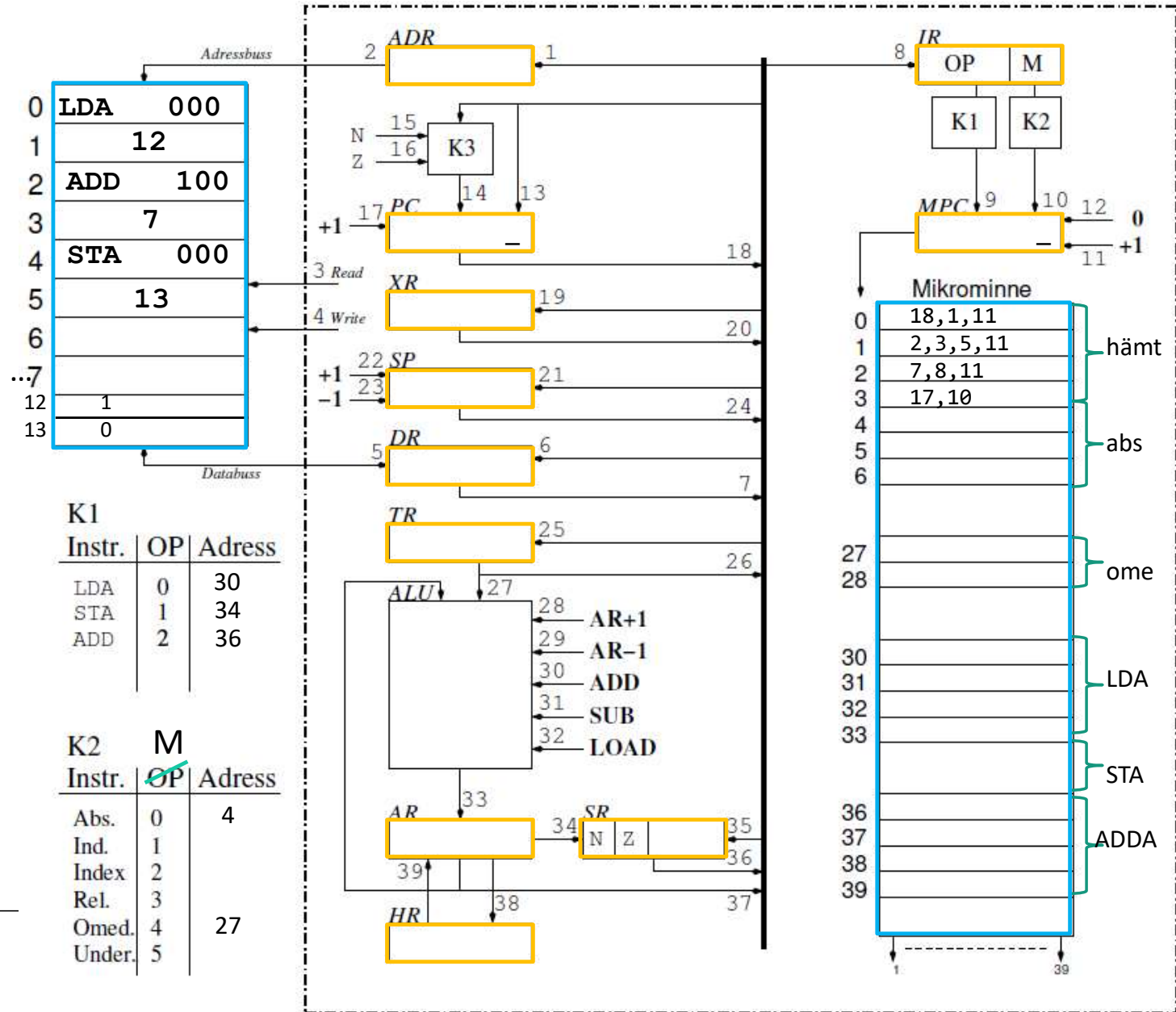
Steg 1 : Hämtfas

M(PC) -> IR



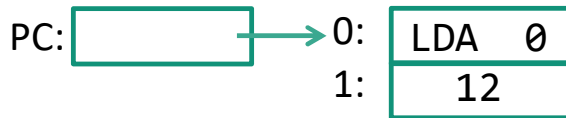
- 0: pc->adr, mpc++ 18, 1, 11
- 1: adr->minne, data->dr, mpc++ 2, 3, 5, 11
- 2: dr->ir, mpc++ 7, 8, 11
- 3: PC++, K2->mpc 17, 10

Vid reset nollställs PC och MPC



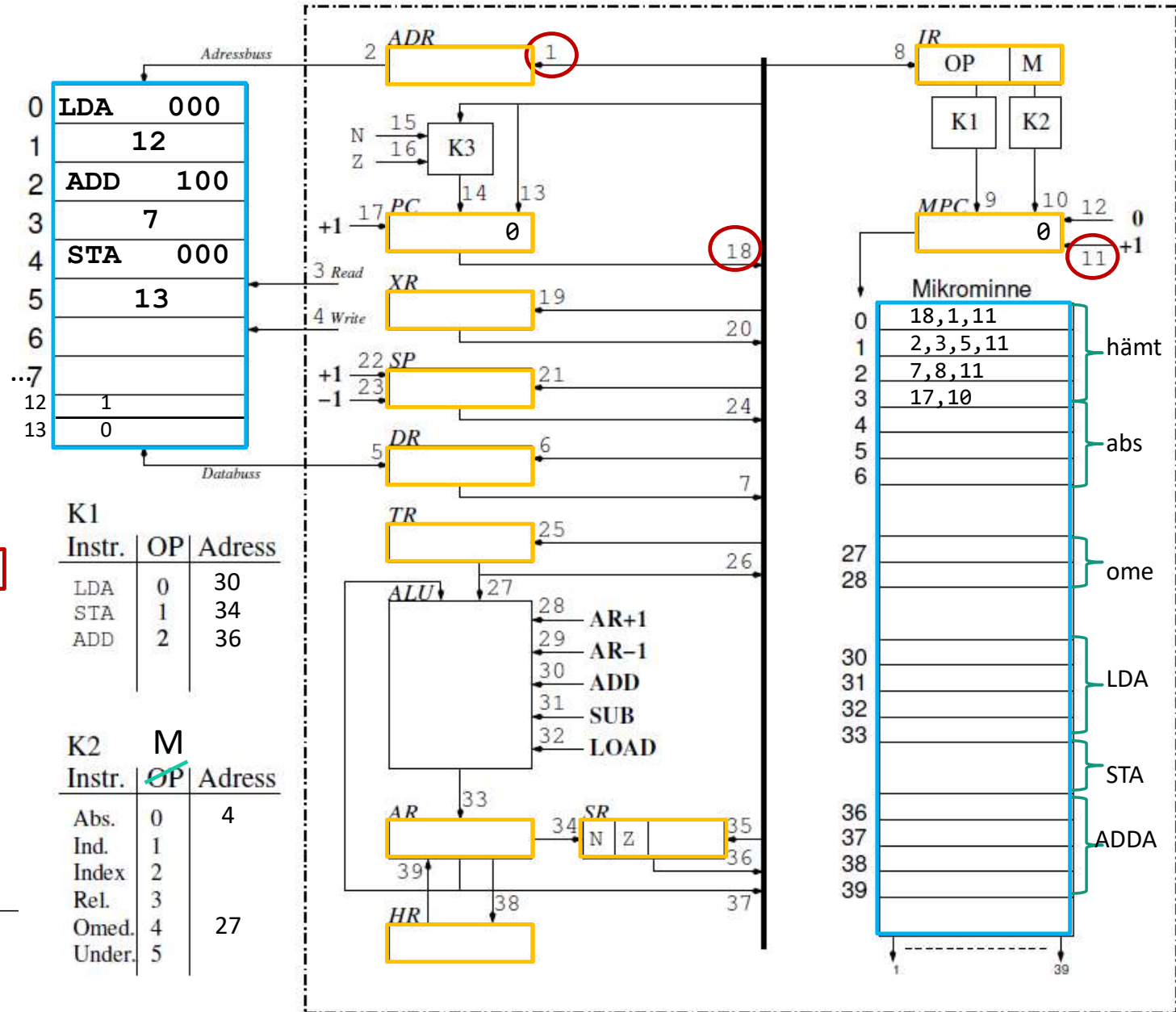
Steg 1 : Hämtfas

M(PC) -> IR



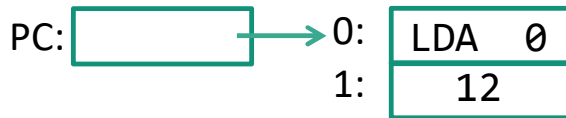
- 0: pc->adr, mpc++ 18, 1, 11
- 1: adr->minne, data->dr, mpc++ 2, 3, 5, 11
- 2: dr->ir, mpc++ 7, 8, 11
- 3: PC++, K2->mpc 17, 10

Rad 0 i Mikrominnet adresseras.
 Effekterna av aktiverade styrsignaler sker vid nästkommande klockflank.

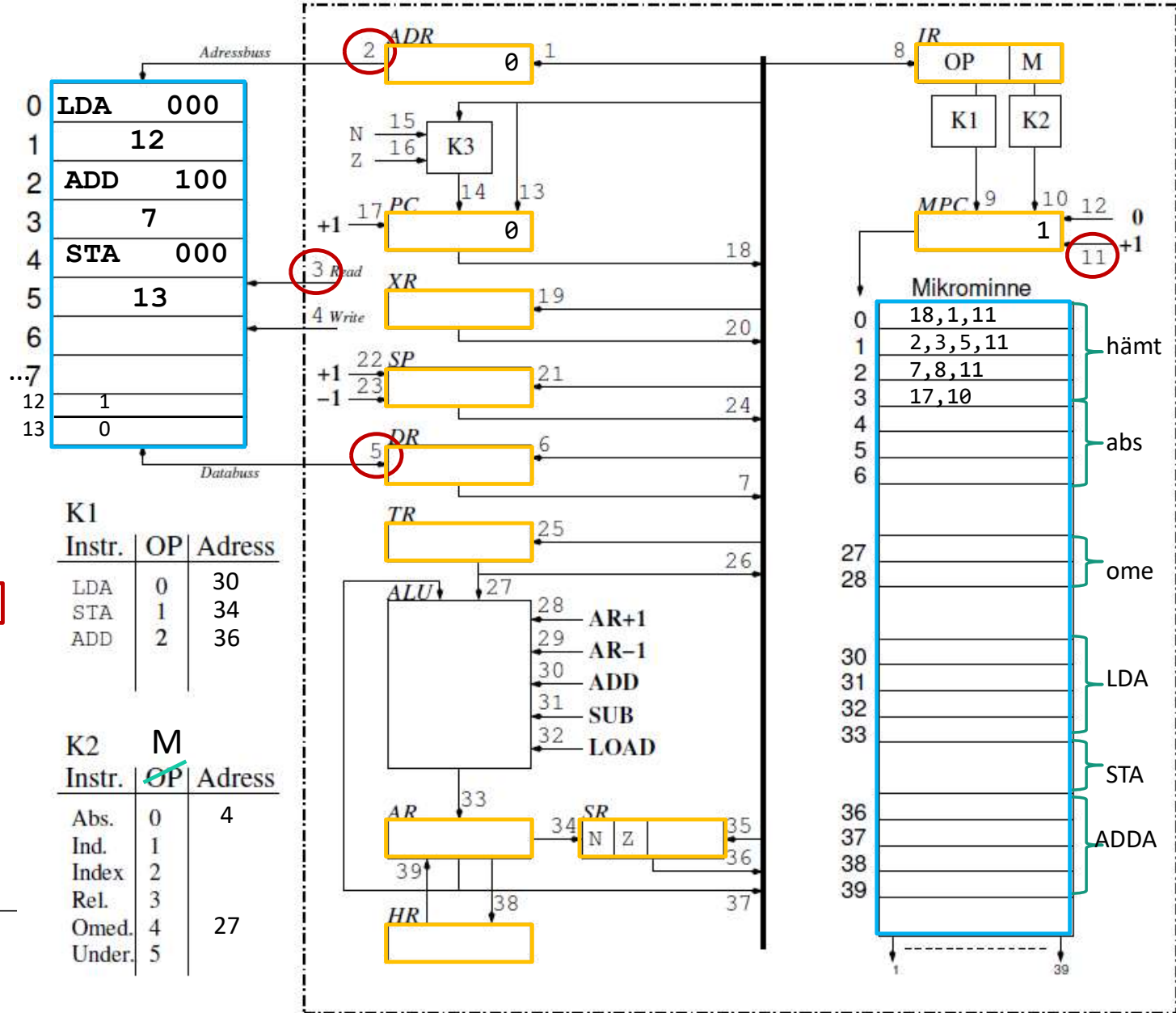


Steg 1 : Hämtfas

M(PC) -> IR



- 0: pc->adr, mpc++ 18, 1, 11
- 1: adr->minne, data->dr, mpc++ 2, 3, 5, 11
- 2: dr->ir, mpc++ 7, 8, 11
- 3: PC++, K2->mpc 17, 10

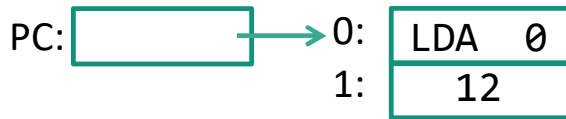


K1		
Instr.	OP	Address
LDA	0	30
STA	1	34
ADD	2	36

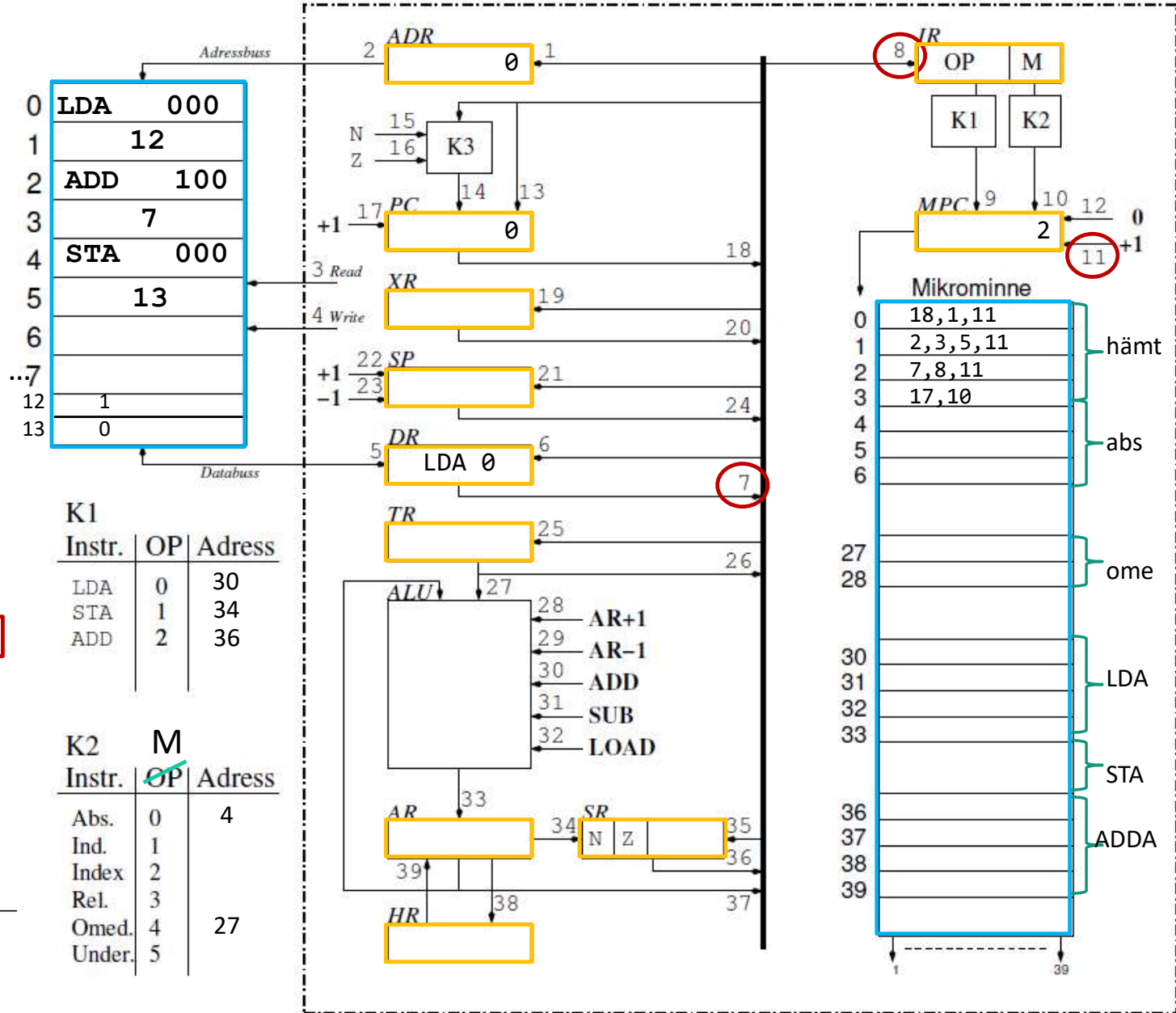
K2		
Instr.	OP	Address
Abs.	0	4
Ind.	1	
Index	2	
Rel.	3	
Omed.	4	27
Under.	5	

Steg 1 : Hämtfas

M(PC) -> IR

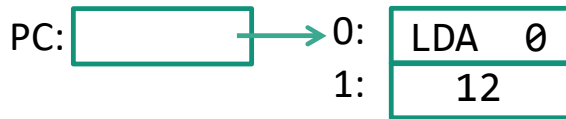


- 0: pc->adr, mpc++ 18, 1, 11
- 1: adr->minne, data->dr, mpc++ 2, 3, 5, 11
- 2: dr->ir, mpc++ 7, 8, 11
- 3: PC++, K2->mpc 17, 10



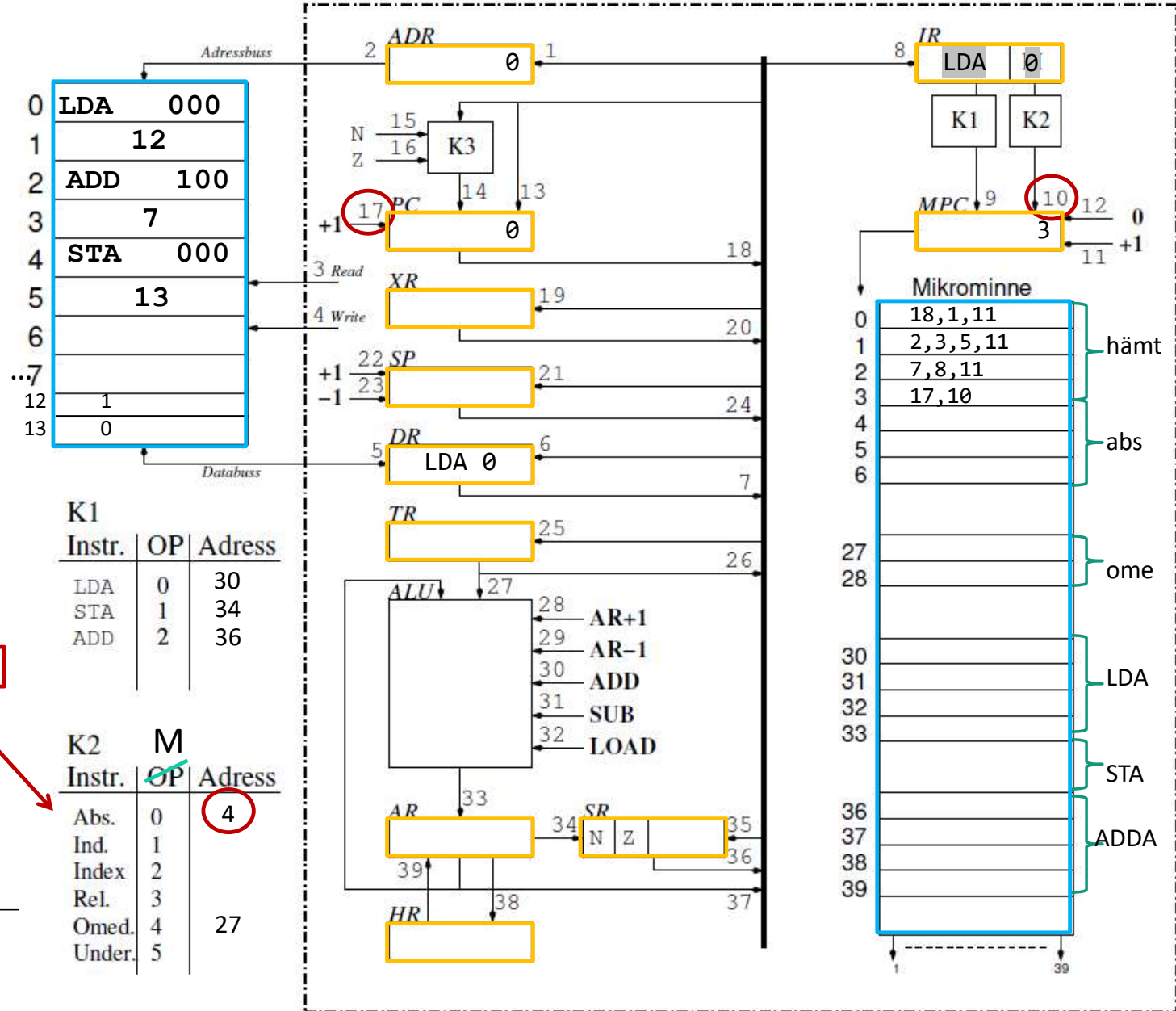
Steg 1 : Hämtfas

M(PC) -> IR



- 0: pc->adr, mpc++ 18, 1, 11
- 1: adr->minne, data->dr, mpc++ 2, 3, 5, 11
- 2: dr->ir, mpc++ 7, 8, 11
- 3: PC++, K2->mpc 17, 10

Signal 10 aktiverar hopp till adresseringsmodsfas (Abs)

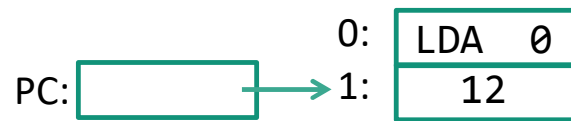


K1		
Instr.	OP	Address
LDA	0	30
STA	1	34
ADD	2	36

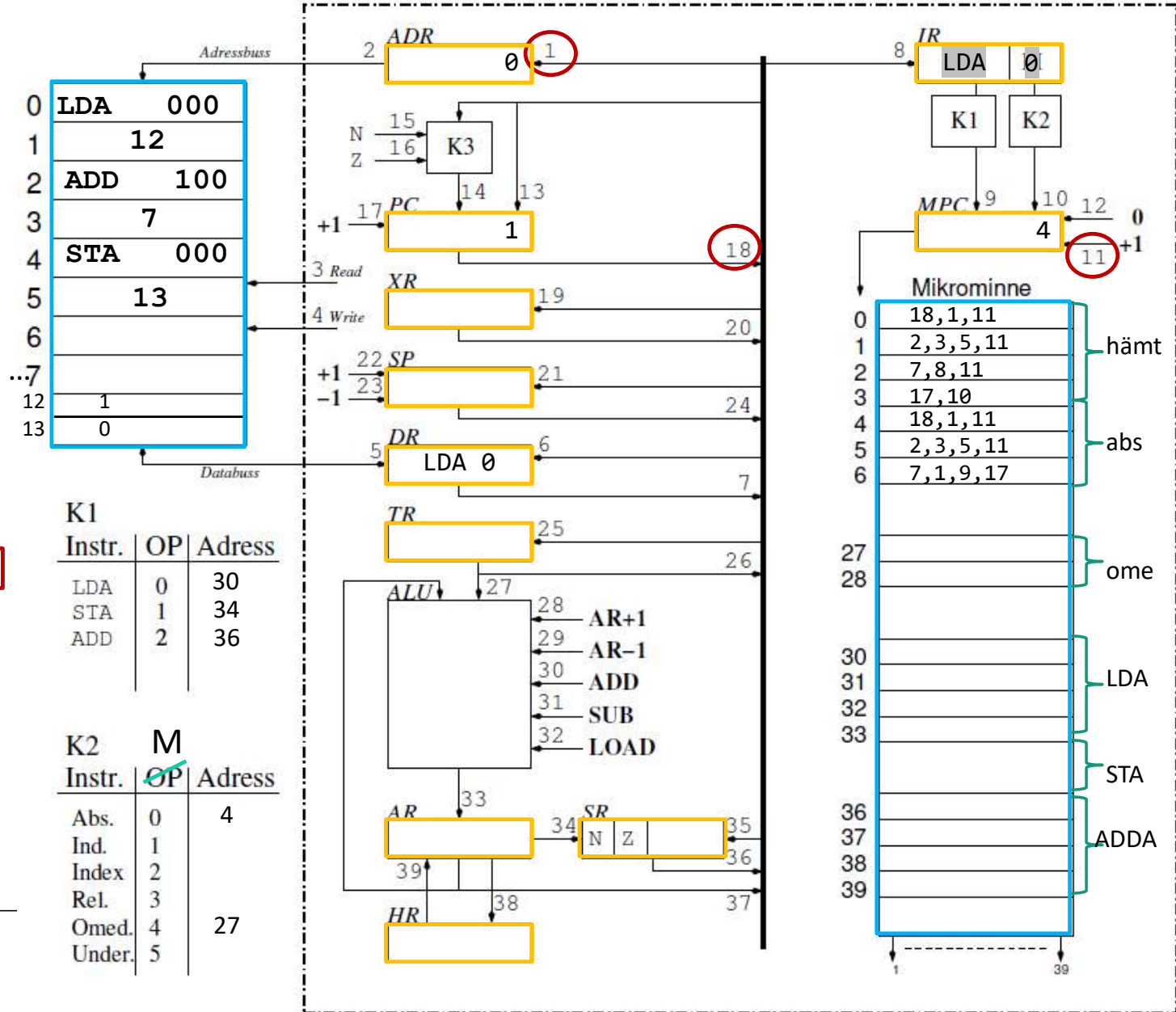
K2			M
Instr.	OP	Address	
Abs.	0		4
Ind.	1		
Index	2		
Rel.	3		
Omed.	4	27	
Under.	5		

Steg 2 : A-modfas (Abs)

M(PC) -> ADR



- 4: pc->adr, mpc++ 18,1,11
- 5: adr->minne, data->dr, mpc++ 2,3,5,11
- 6: dr->adr, K1->mpc, PC++ 7,1,9,17

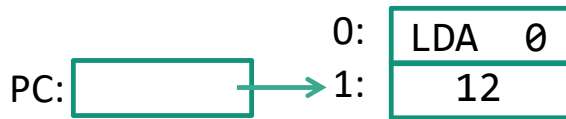


K1		
Instr.	OP	Address
LDA	0	30
STA	1	34
ADD	2	36

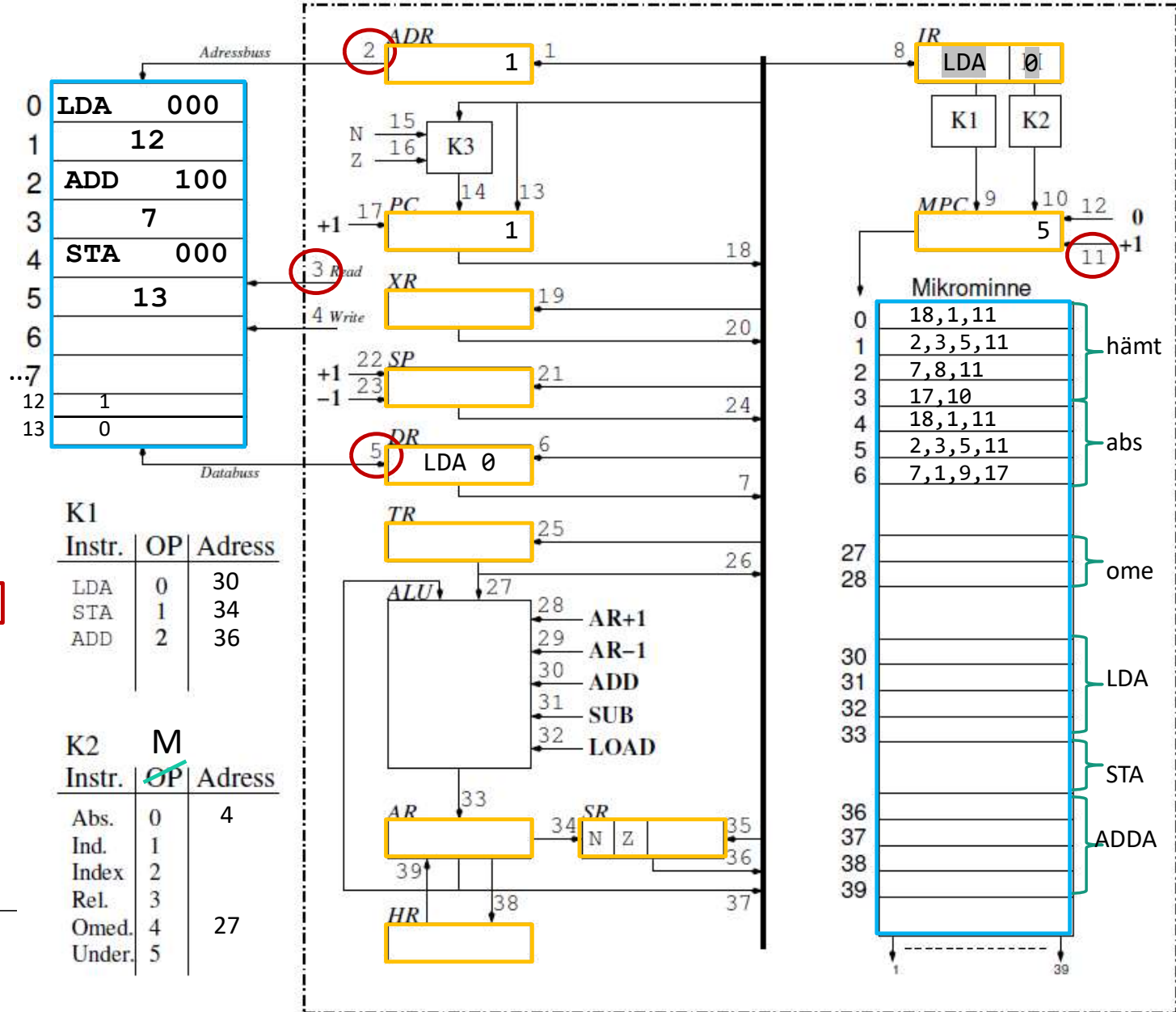
K2 M		
Instr.	OP	Address
Abs.	0	4
Ind.	1	
Index	2	
Rel.	3	
Omed.	4	27
Under.	5	

Steg 2 : A-modfas (Abs)

M(PC) -> ADR

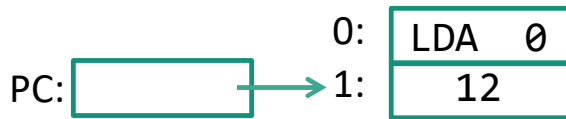


- 4: pc->adr, mpc++ 18, 1, 11
- 5: adr->minne, data->dr, mpc++ 2, 3, 5, 11
- 6: dr->adr, K1->mpc, PC++ 7, 1, 9, 17



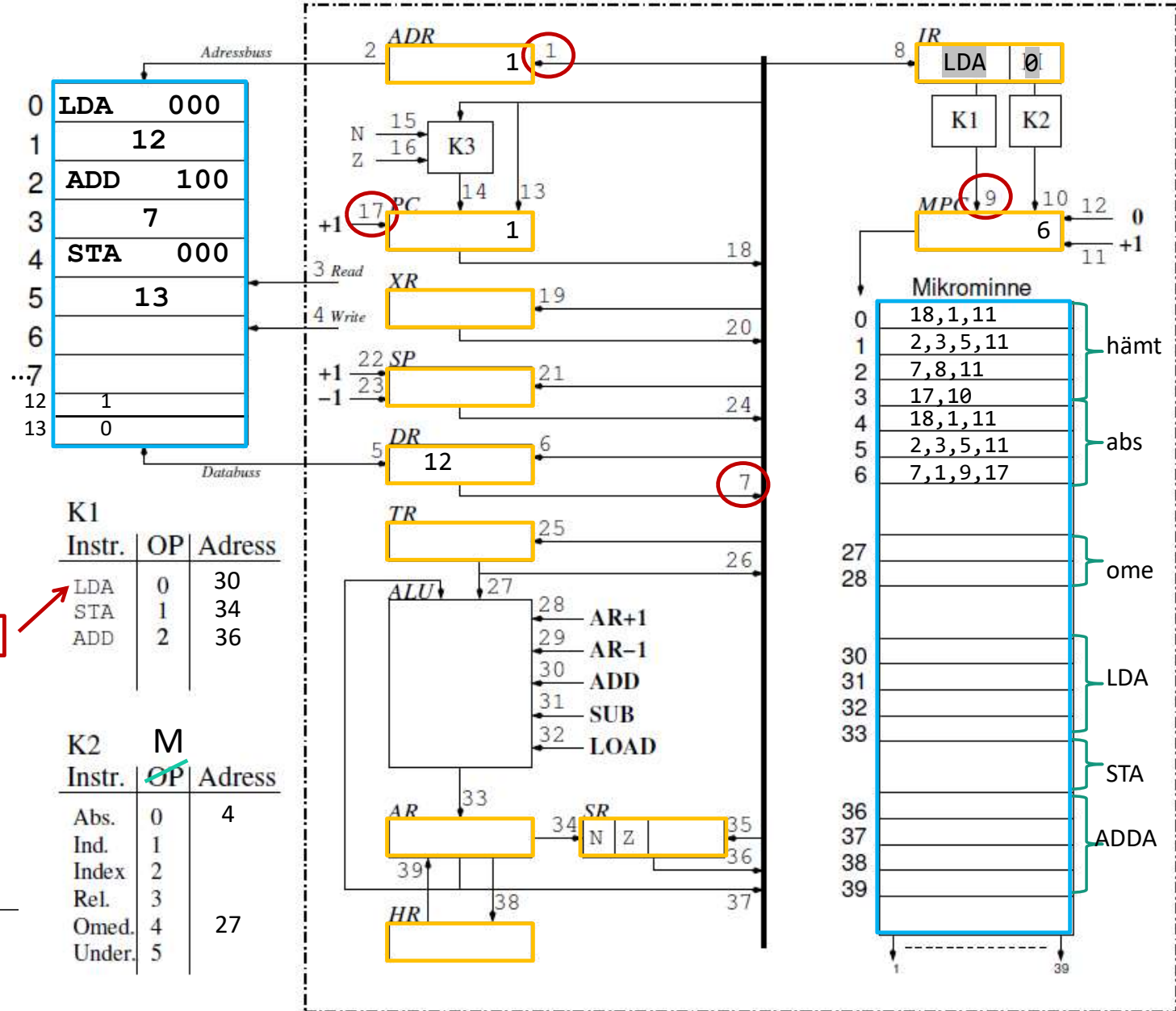
Steg 2 : A-modfas (Abs)

M(PC) -> ADR



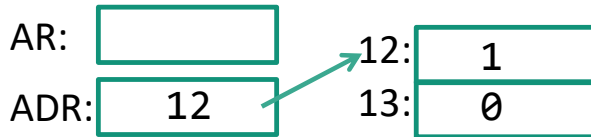
- 4: pc->adr, mpc++ 18, 1, 11
- 5: adr->minne, data->dr, mpc++ 2, 3, 5, 11
- 6: dr->adr, K1->mpc, PC++ 7, 1, 9, 17

Signal 9 aktiverar hopp till exekveringsfas (LDA)

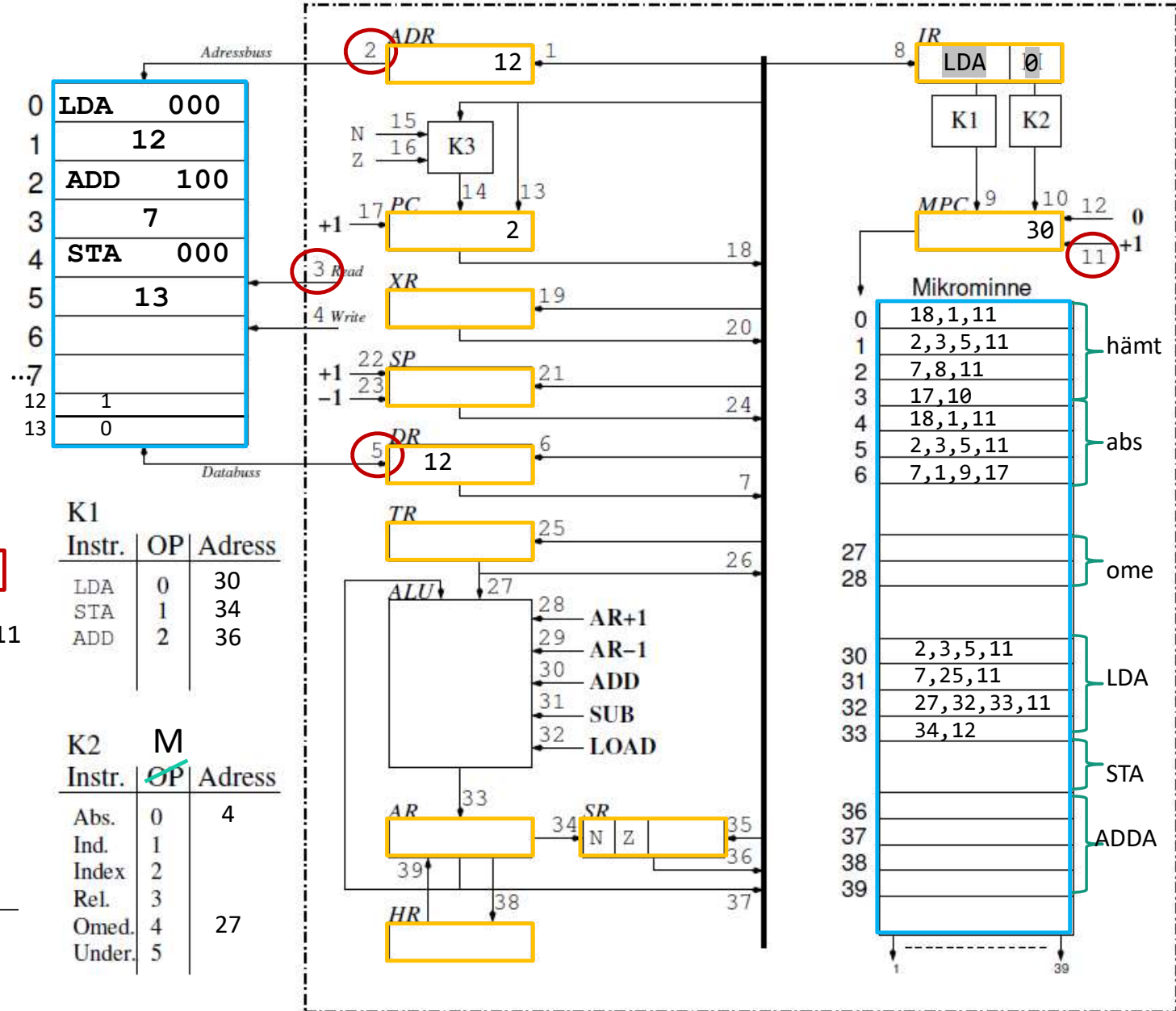


Steg 3 : Exe-fas (LDA)

$$AR = M(ADR)$$

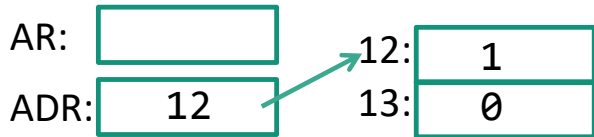


- 30: adr->minne,data->dr,mpc++ 2,3,5,11
- 31: dr->tr,mpc++ 7,25,11
- 32: tr->ar,mpc++ 27,32,33,11
- 33: status, 0->mpc 34,12

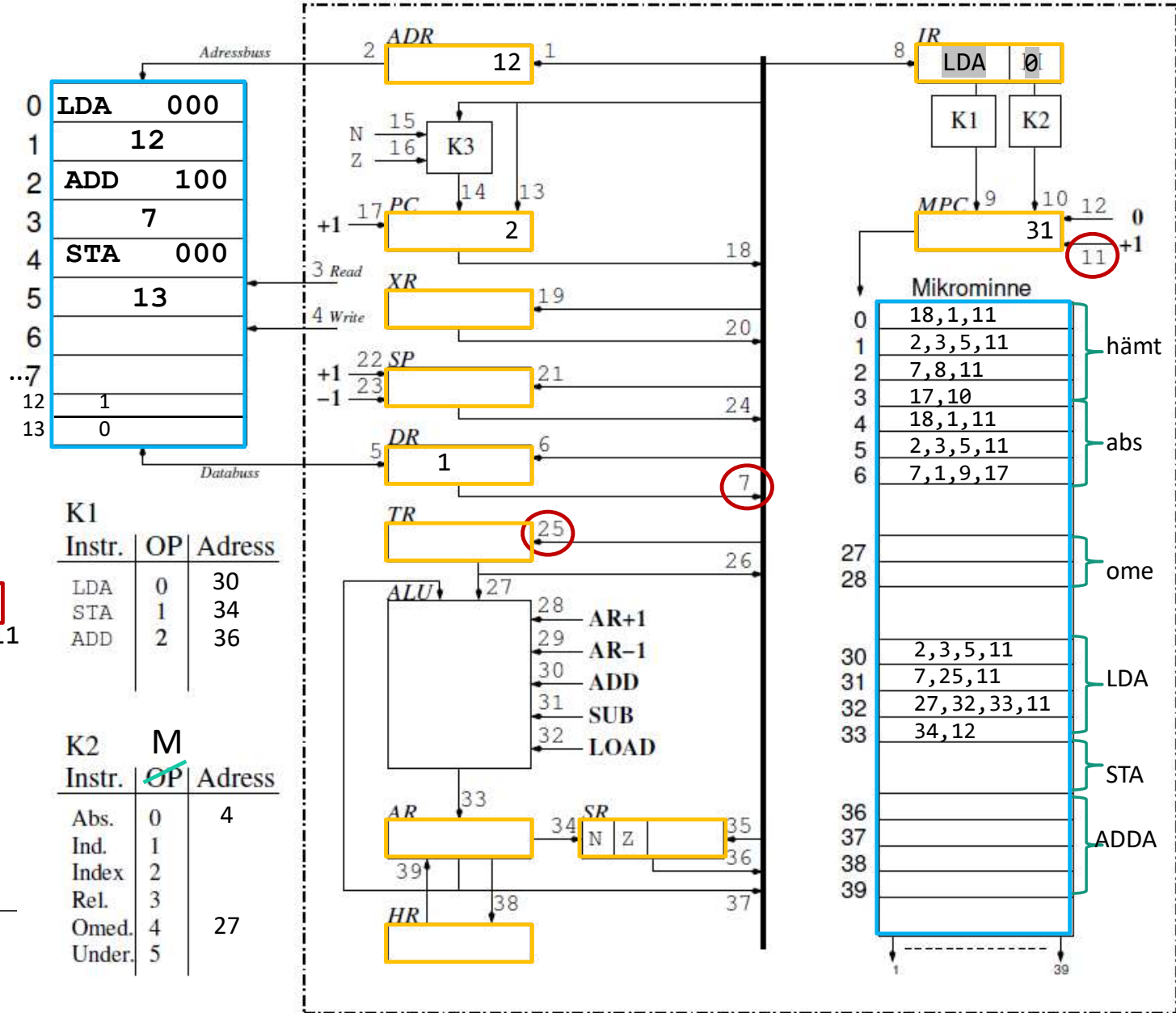


Steg 3 : Exe-fas (LDA)

$$AR = M(ADR)$$

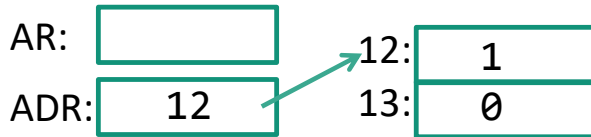


- 30: adr->minne,data->dr,mpc++ 2,3,5,11
- 31: dr->tr,mpc++ 7,25,11
- 32: tr->ar,mpc++ 27,32,33,11
- 33: status, 0->mpc 34,12

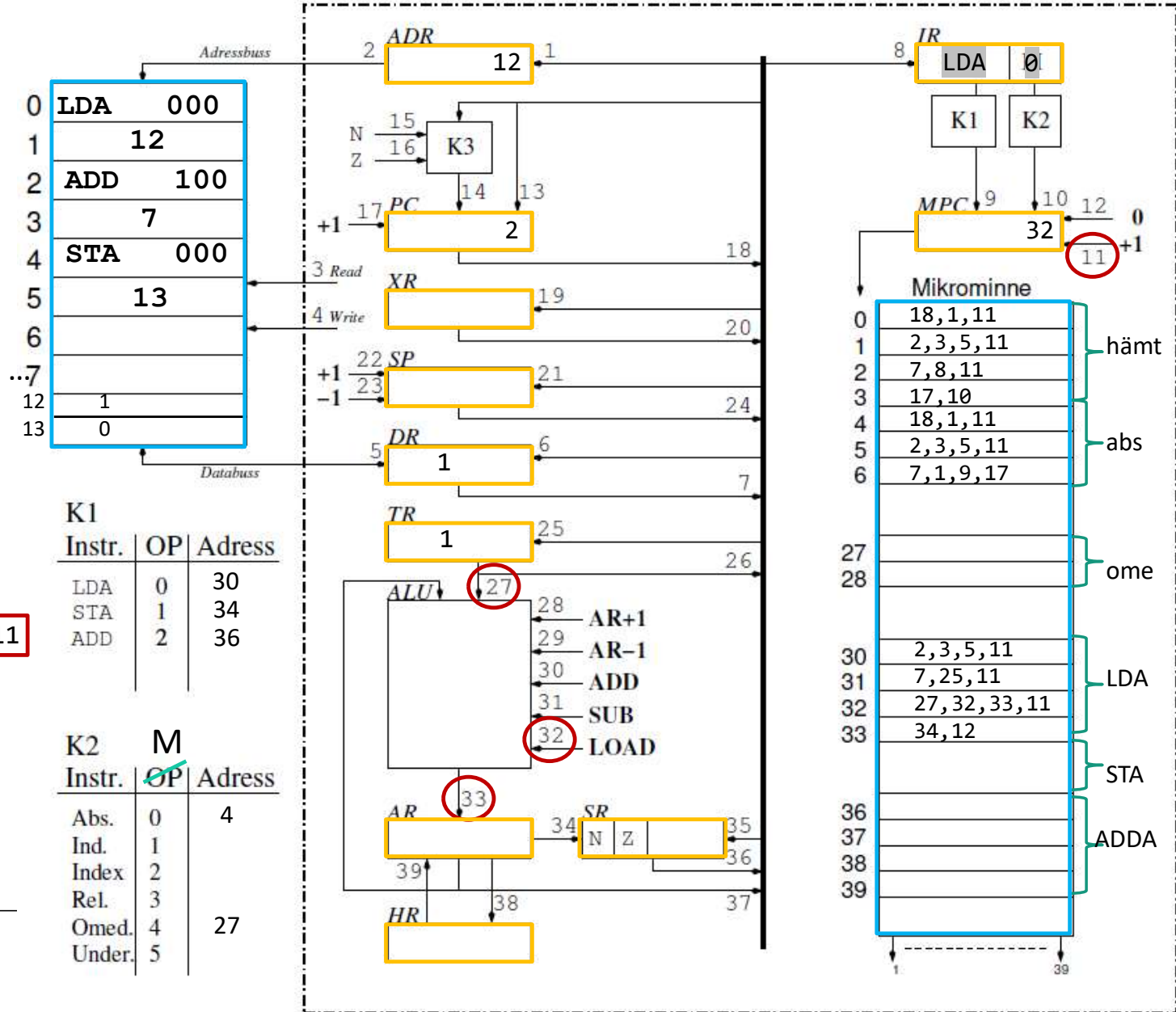


Steg 3 : Exe-fas (LDA)

$$AR = M(ADR)$$



- 30: adr->minne,data->dr,mpc++ 2,3,5,11
- 31: dr->tr,mpc++ 7,25,11
- 32: tr->ar,mpc++ 27,32,33,11
- 33: status, 0->mpc 34,12

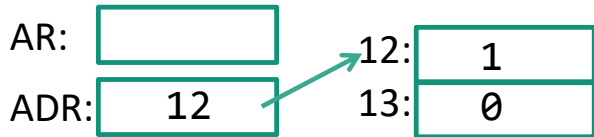


K1		
Instr.	OP	Address
LDA	0	30
STA	1	34
ADD	2	36

K2 M		
Instr.	OP	Address
Abs.	0	4
Ind.	1	
Index	2	
Rel.	3	
Omed.	4	27
Under.	5	

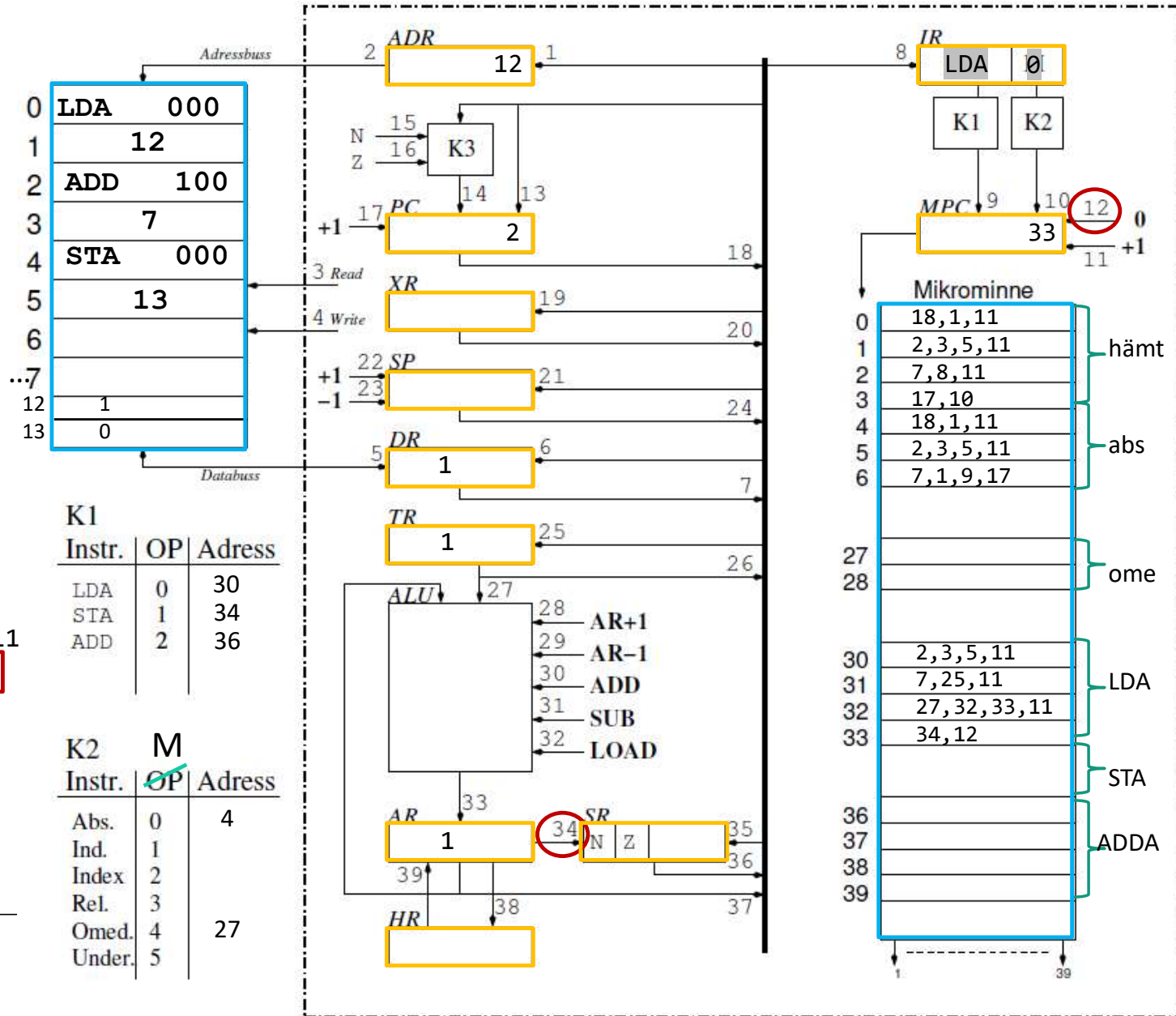
Steg 3 : Exe-fas (LDA)

$$AR = M(ADR)$$



- 30: adr->minne,data->dr,mpc++ 2,3,5,11
- 31: dr->tr,mpc++ 7,25,11
- 32: tr->ar,mpc++ 27,32,33,11
- 33: status, 0->mpc 34,12

Signal 12 aktiverar
nollställning av MPC,
för nästa hämtfas



K1		
Instr.	OP	Address
LDA	0	30
STA	1	34
ADD	2	36

K2 M		
Instr.	OP	Address
Abs.	0	4
Ind.	1	
Index	2	
Rel.	3	
Omed.	4	27
Under.	5	

Mikrokod för ADD #7

Mikrokod för ADD #7

Steg 1 : H-fas, som förut

3: PC++, K2->mpc

17, 10

Steg 2 : A-fas, Omedelbar

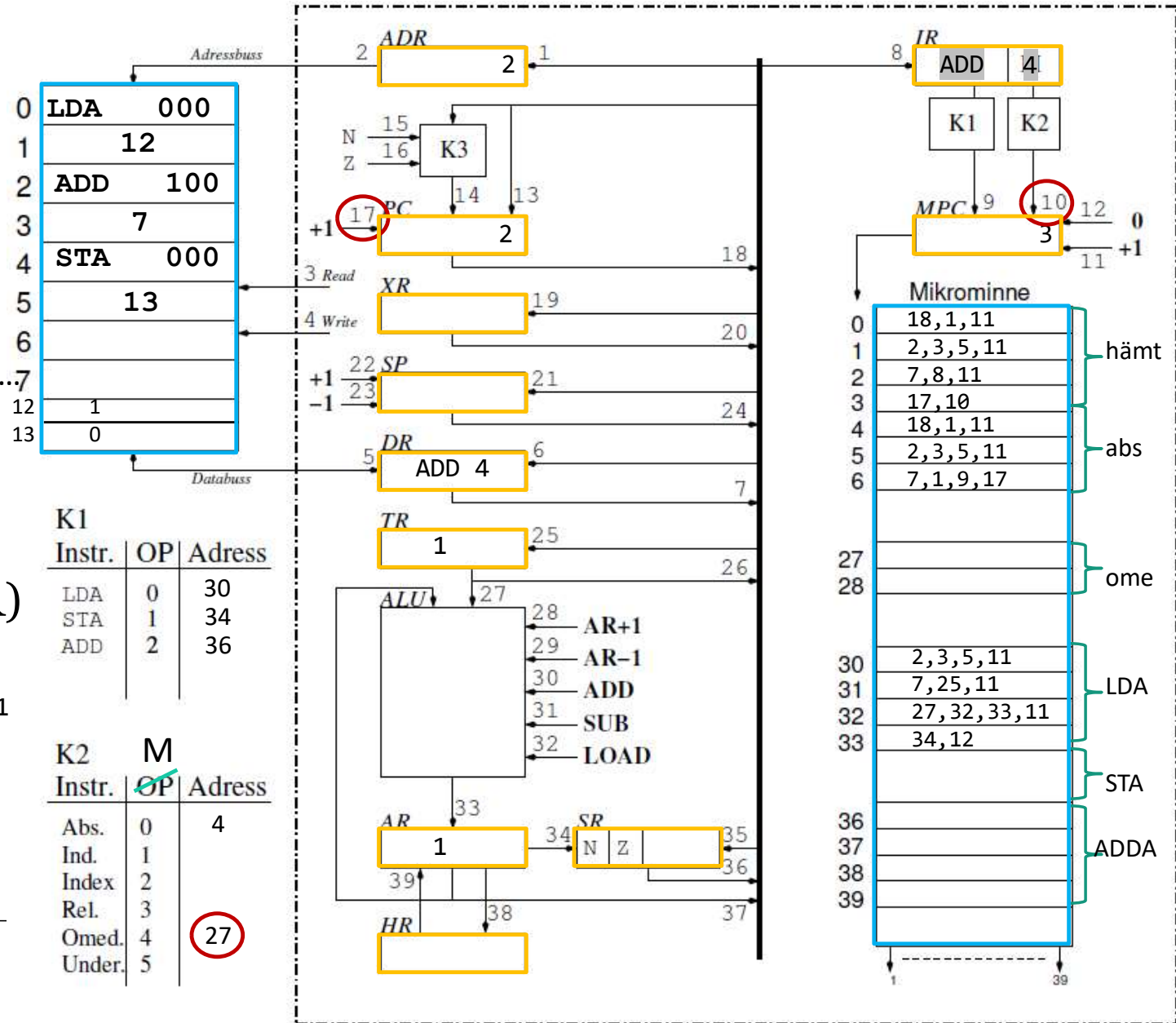
27: PC->adr, mpc++
28: PC++, K1->mpc

18, 1, 11
17, 9

Steg 3 : Exe, AR=AR+M(ADR)

36: data->dr, mpc++
37: dr->tr, mpc++
38: ar+tr->ar, mpc++
39: status, 0->mpc

2, 3, 5, 11
7, 25, 11
27, 33, 30, 11
34, 12



K1	Instr.	OP	Address
	LDA	0	30
	STA	1	34
	ADD	2	36

K2	M	Instr.	OP	Address
	Abs.	0		4
	Ind.	1		
	Index	2		
	Rel.	3		
	Omed.	4		
	Under.	5		

Mikrokod för ADD #7

Steg 1 : H-fas, som förut

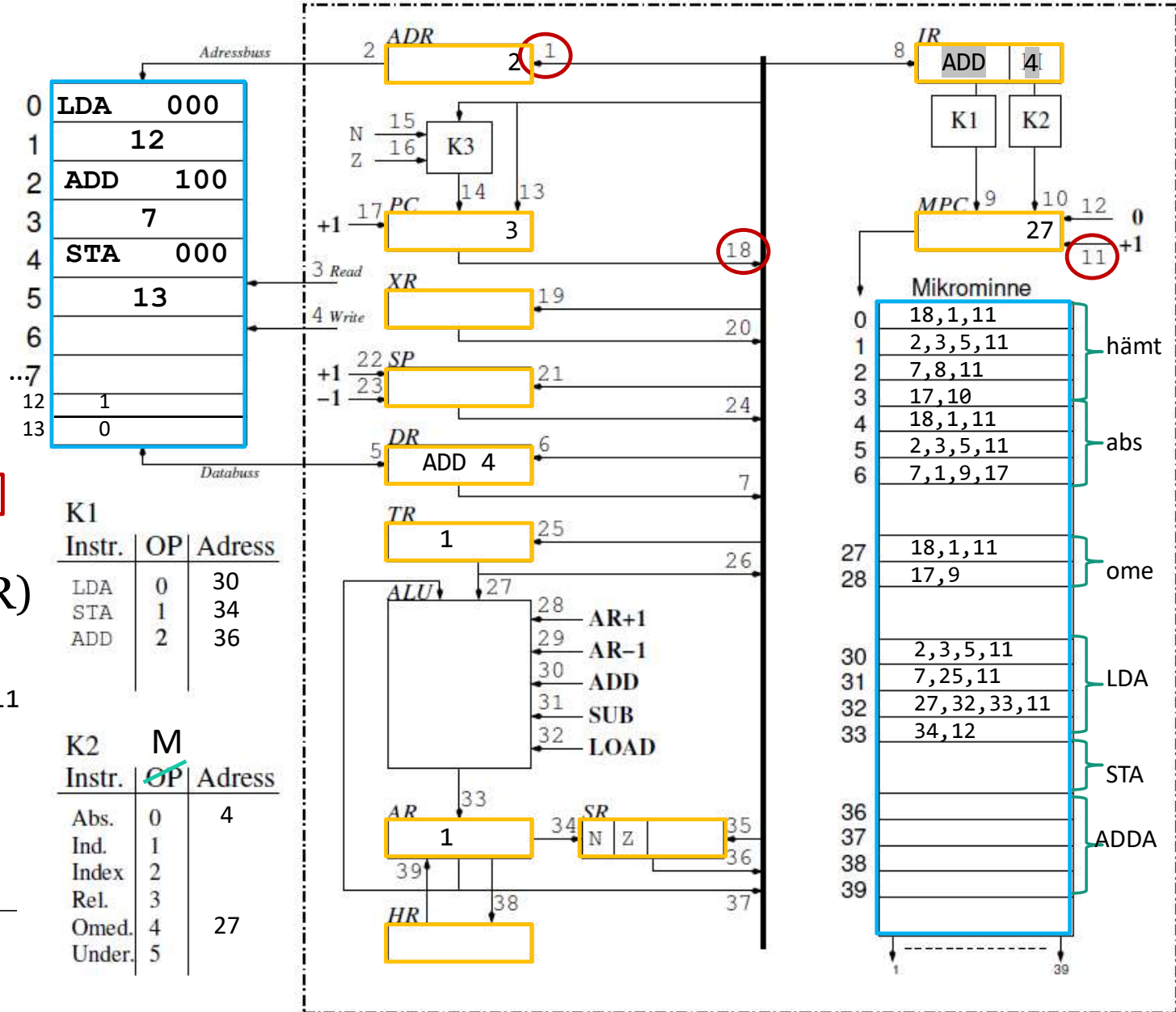
3: PC++, K2->mpc 17,10

Steg 2 : A-fas, Omedelbar

27: PC->adr, mpc++ **18,1,11**
 28: PC++, K1->mpc 17,9

Steg 3 : Exe, AR=AR+M(ADR)

36: data->dr, mpc++ 2,3,5,11
 37: dr->tr, mpc++ 7,25,11
 38: ar+tr->ar, mpc++ 27,33,30,11
 39: status, 0->mpc 34,12



Mikrokod för ADD #7

Steg 1 : H-fas, som förut

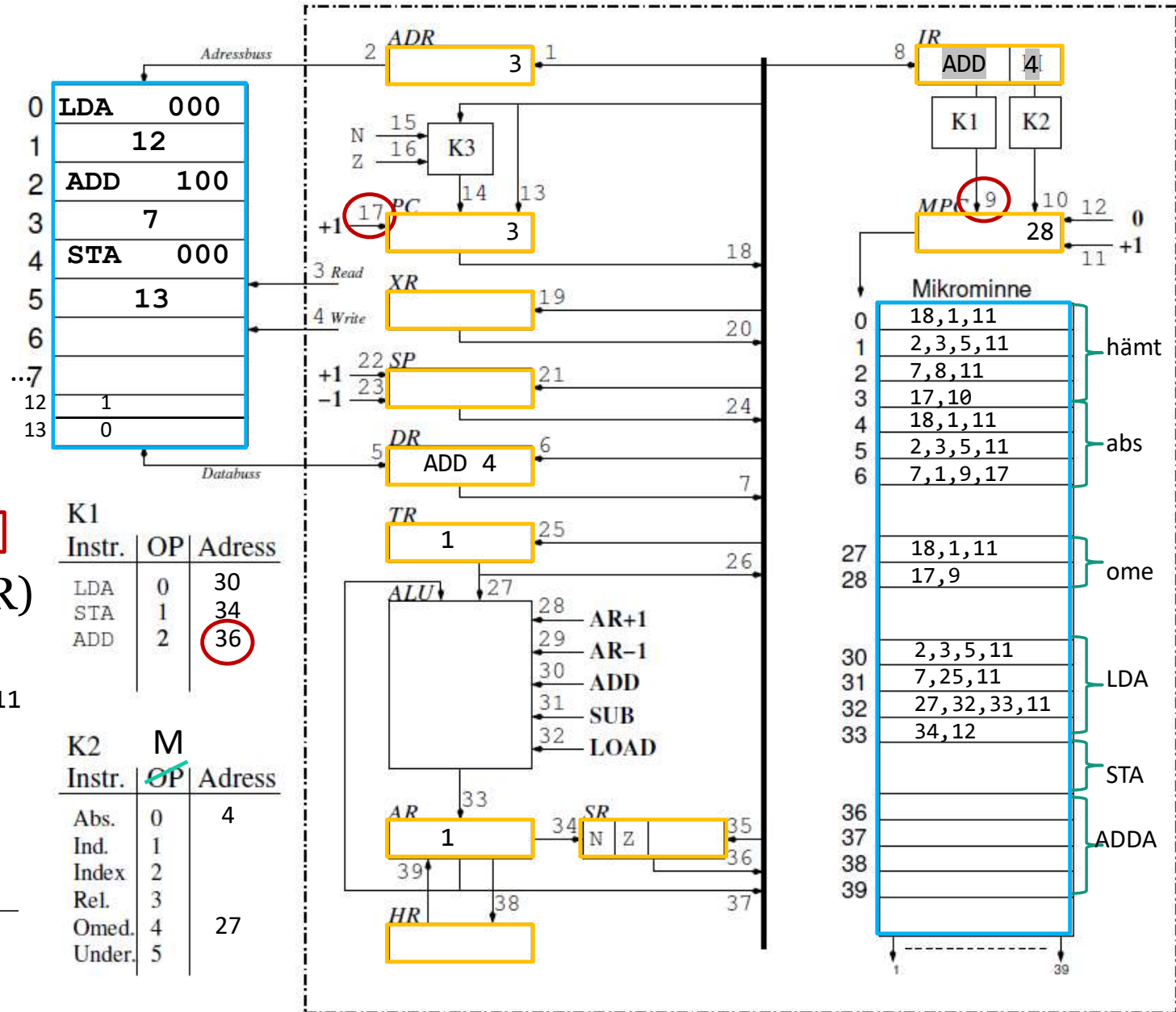
...
3: PC++, K2->mpc 17,10

Steg 2 : A-fas, Omedelbar

27: PC->adr, mpc++ 18,1,11
28: PC++, K1->mpc 17,9

Steg 3 : Exe, AR=AR+M(ADR)

36: data->dr, mpc++ 2,3,5,11
37: dr->tr, mpc++ 7,25,11
38: ar+tr->ar, mpc++ 27,33,30,11
39: status, 0->mpc 34,12



Mikrokod för ADD #7

Steg 1 : H-fas, som förut

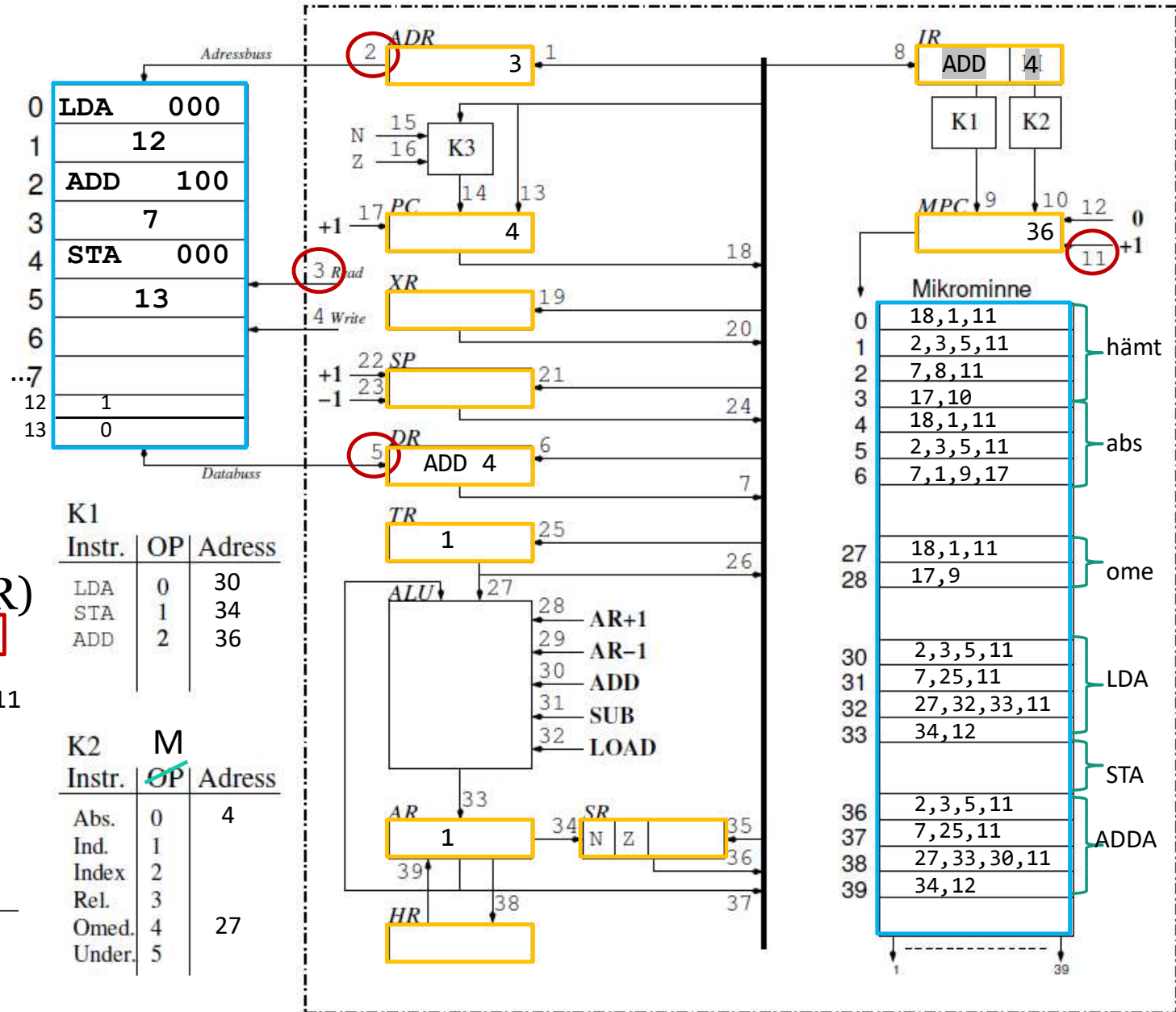
...
3: PC++, K2->mpc 17,10

Steg 2 : A-fas, Omedelbar

27: PC->adr, mpc++ 18,1,11
28: PC++, K1->mpc 17,9

Steg 3 : Exe, AR=AR+M(ADR)

36: data->dr, mpc++ 2, 3, 5, 11
37: dr->tr, mpc++ 7, 25, 11
38: ar+tr->ar, mpc++ 27, 33, 30, 11
39: status, 0->mpc 34, 12



Mikrokod för ADD #7

Steg 1 : H-fas, som förut

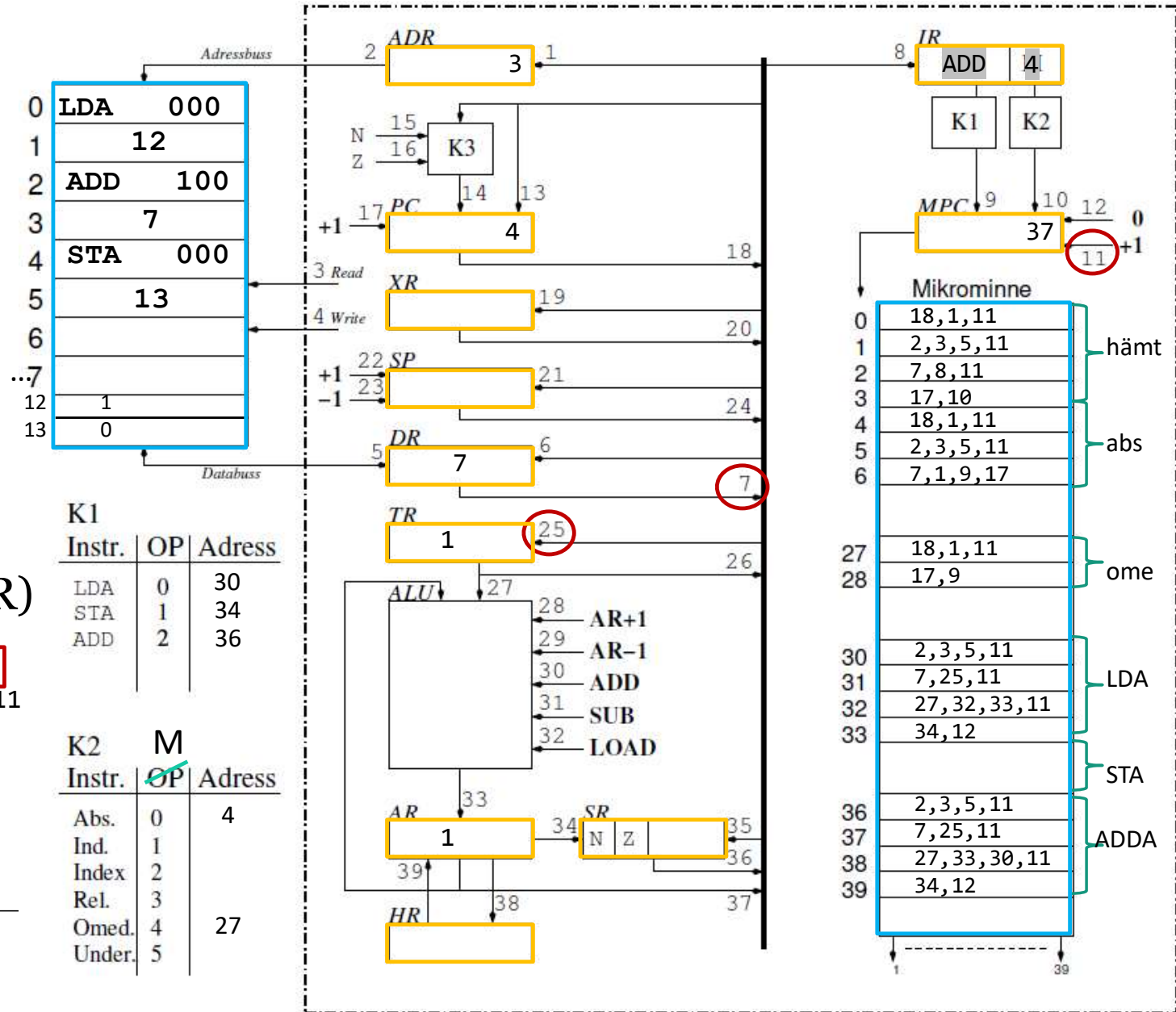
...
3: PC++, K2->mpc 17,10

Steg 2 : A-fas, Omedelbar

27: PC->adr, mpc++ 18,1,11
28: PC++, K1->mpc 17,9

Steg 3 : Exe, AR=AR+M(ADR)

36: data->dr, mpc++ 2,3,5,11
37: dr->tr, mpc++ **7,25,11**
38: ar+tr->ar, mpc++ 27,33,30,11
39: status, 0->mpc 34,12



Mikrokod för ADD #7

Steg 1 : H-fas, som förut

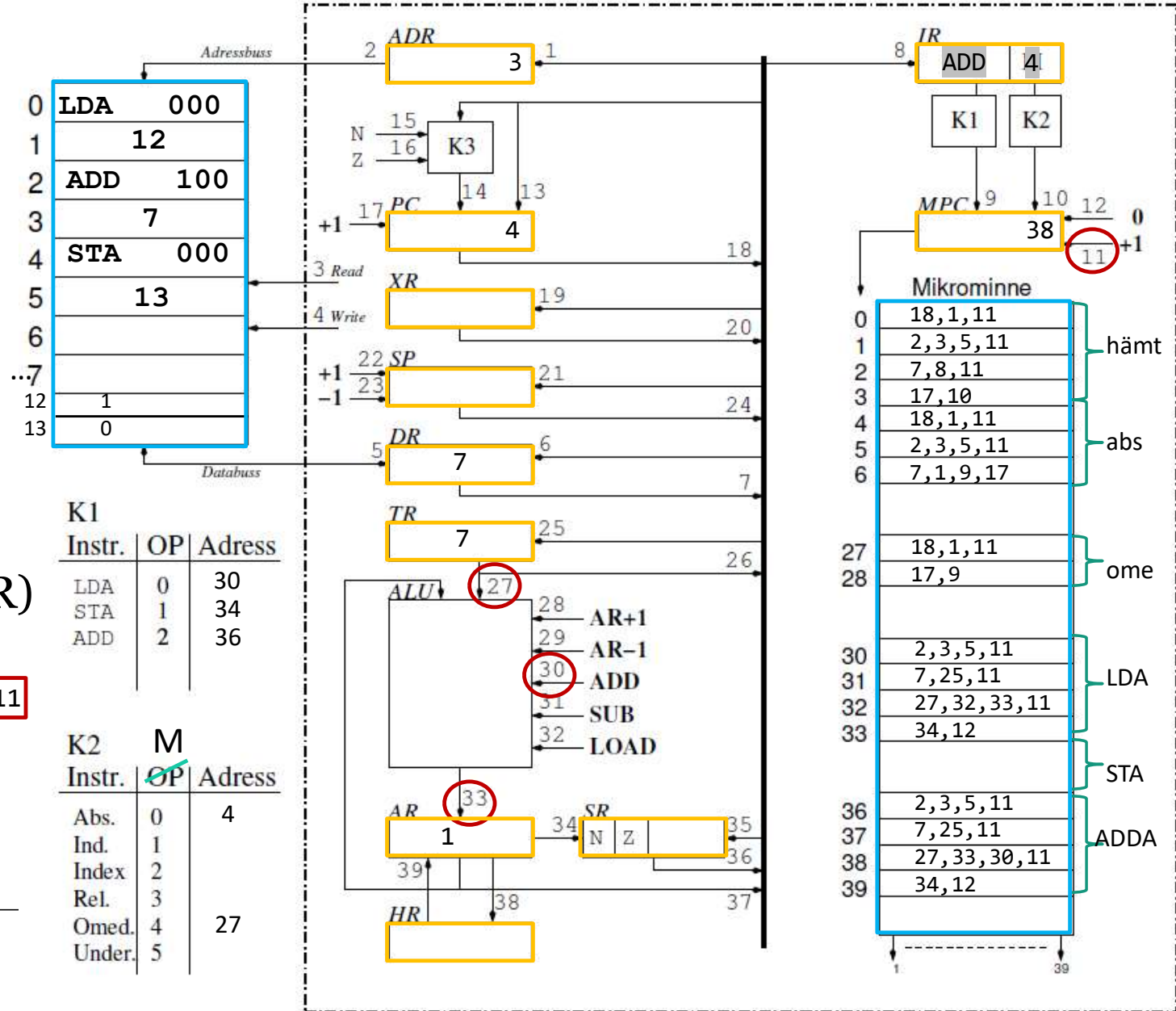
...
3: PC++, K2->mpc 17,10

Steg 2 : A-fas, Omedelbar

27: PC->adr, mpc++ 18,1,11
28: PC++, K1->mpc 17,9

Steg 3 : Exe, AR=AR+M(ADR)

36: data->dr, mpc++ 2,3,5,11
37: dr->tr, mpc++ 7,25,11
38: ar+tr->ar, mpc++ **27,33,30,11**
39: status, 0->mpc 34,12



Mikrokod för ADD #7

Steg 1 : H-fas, som förut

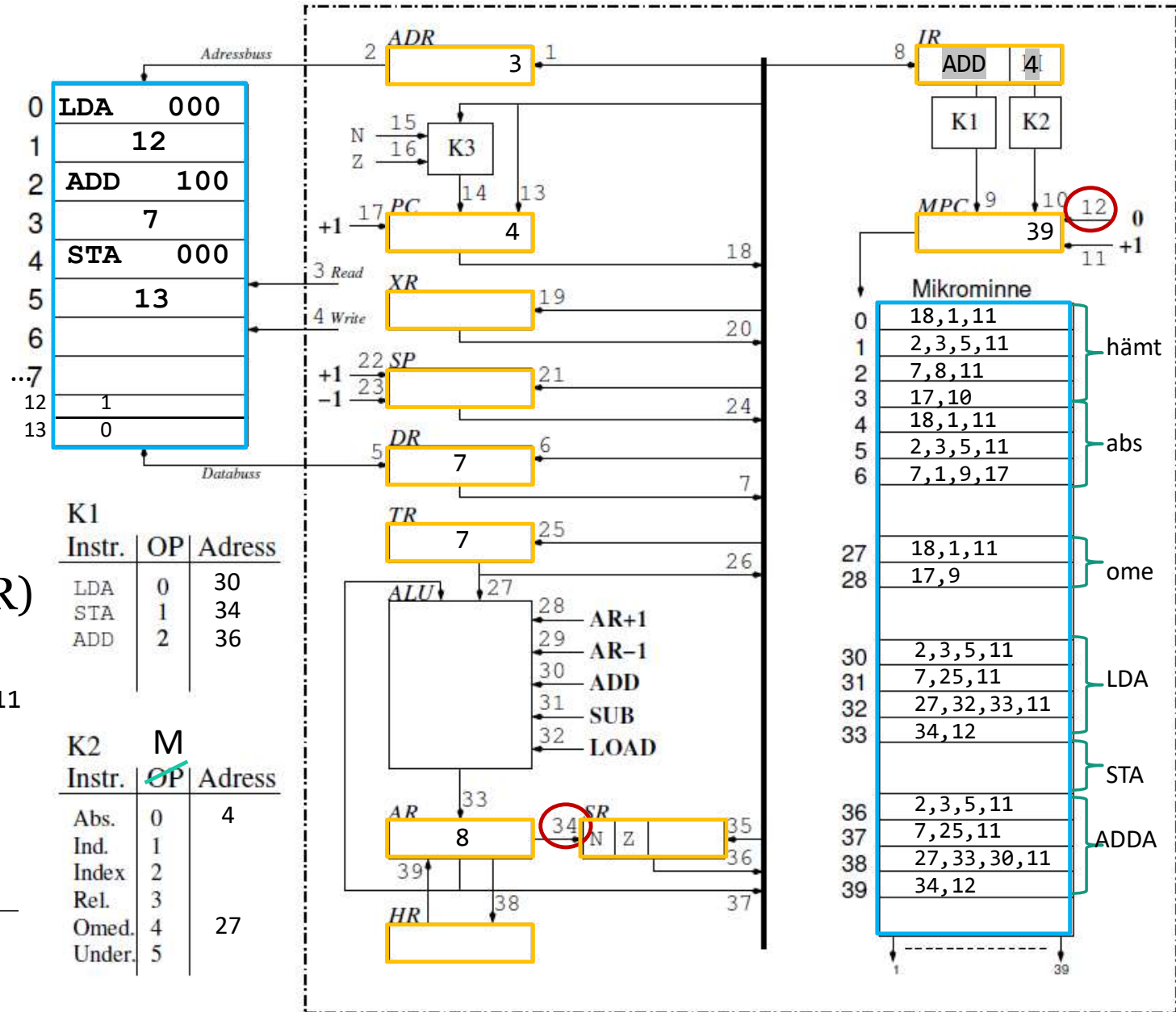
...
3: PC++, K2->mpc 17,10

Steg 2 : A-fas, Omedelbar

27: PC->adr, mpc++ 18,1,11
28: PC++, K1->mpc 17,9

Steg 3 : Exe, AR=AR+M(ADR)

36: data->dr, mpc++ 2,3,5,11
37: dr->tr, mpc++ 7,25,11
38: ar+tr->ar, mpc++ 27,33,30,11
39: status, 0->mpc 34,12



Mikrokod för STA (13)

Mikrokod för STA 13

Steg 1 : H-fas, som förut

3: PC++, K2->mpc

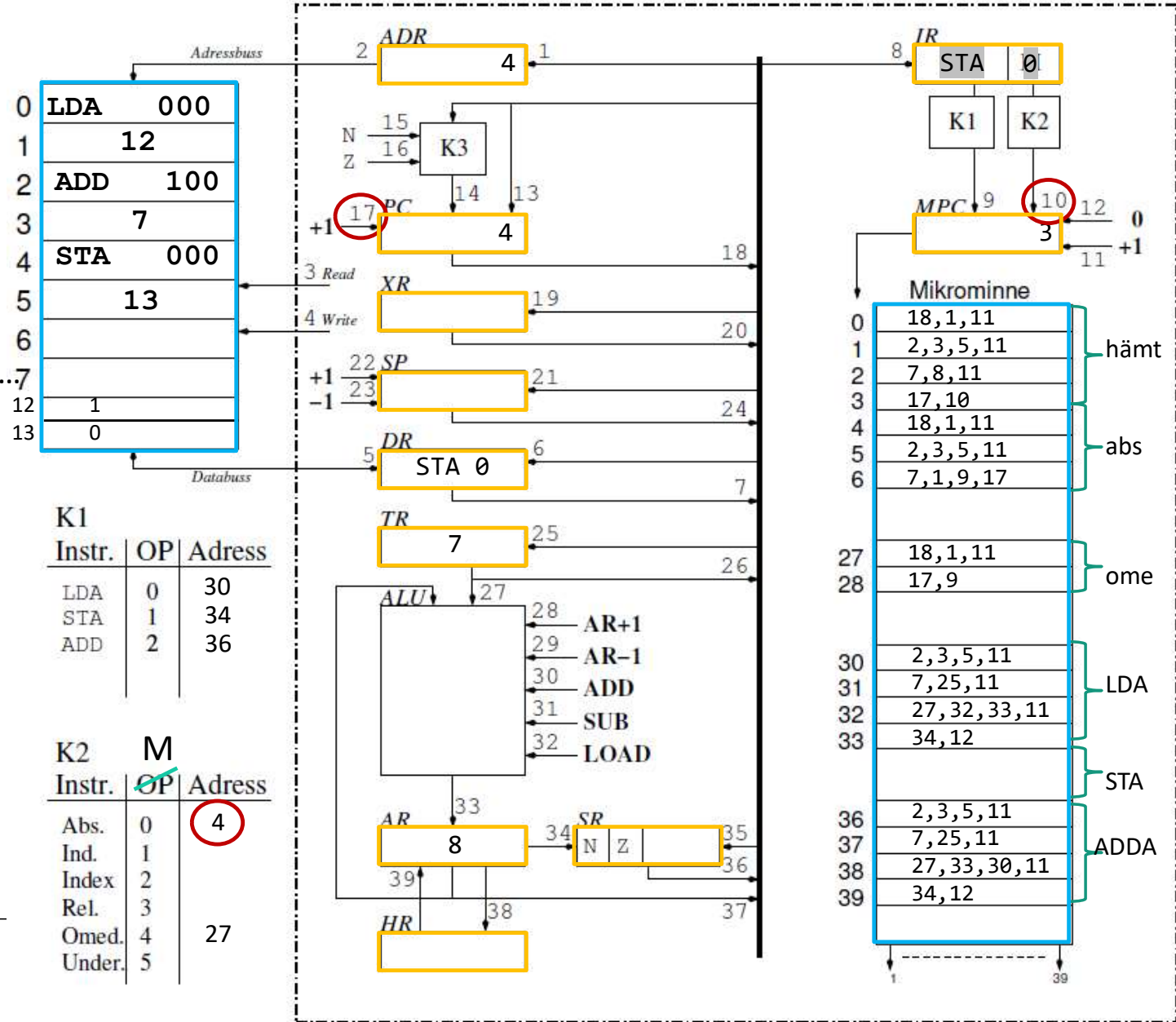
17, 10

Steg 2 : A-fas, Absolut

6: dr->adr, K1->mpc, PC++ 7, 1, 9, 17

Steg 3 : Exe, AR->M(ADR)

34: ar->dr, mpc++ 37, 6, 11
 35: dr->M, 0->mpc 2, 4, 5, 12



K1		
Instr.	OP	Address
LDA	0	30
STA	1	34
ADD	2	36

K2		
Instr.	OP	Address
Abs.	0	4
Ind.	1	
Index	2	
Rel.	3	
Omed.	4	27
Under.	5	

Mikrokod för STA 13

Steg 1 : H-fas, som förut

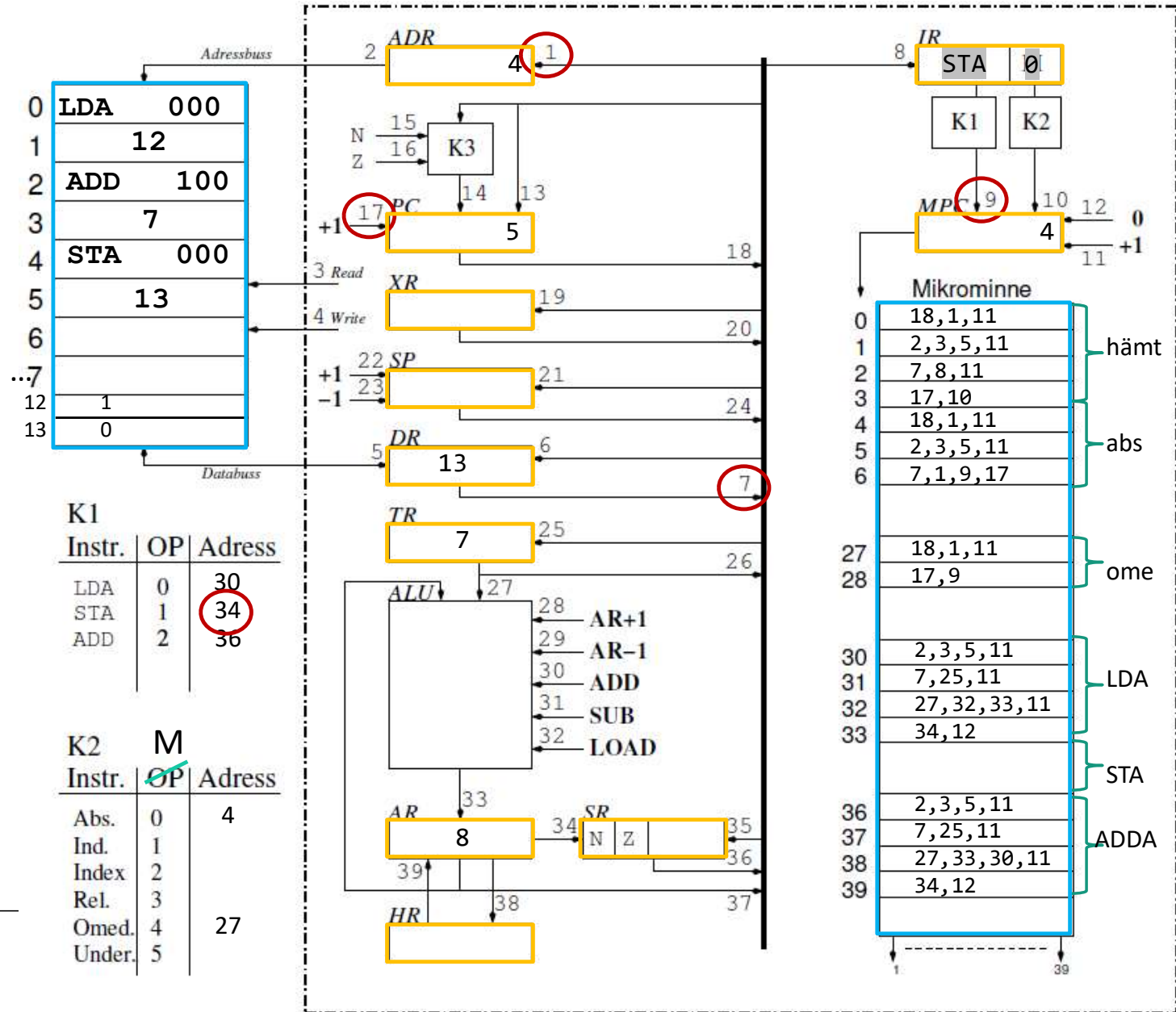
...
3: PC++, K2->mpc 17,10

Steg 2 : A-fas, Absolut

...
6: dr->adr, K1->mpc, PC++ 7,1,9,17

Steg 3 : Exe, AR->M(ADR)

34: ar->dr, mpc++ 37,6,11
35: dr->M, 0->mpc 2,4,5,12



Mikrokod för STA 13

Steg 1 : H-fas, som förut

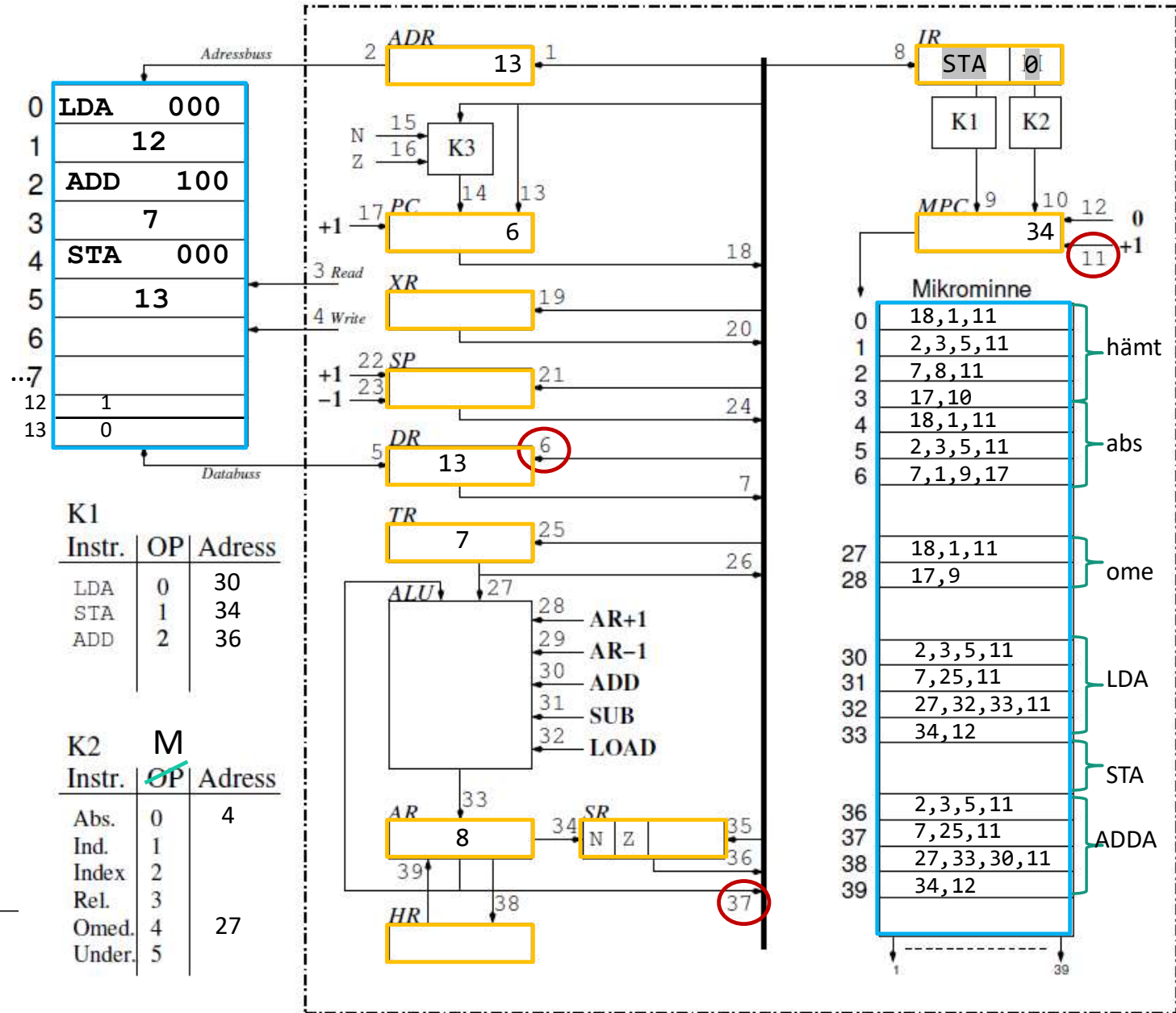
...
3: PC++, K2->mpc 17, 10

Steg 2 : A-fas, Absolut

...
6: dr->adr, K1->mpc, PC++ 7, 1, 9, 17

Steg 3 : Exe, AR->M(ADR)

34: ar->dr, mpc++ 37, 6, 11
35: dr->M, 0->mpc 2, 4, 5, 12



Mikrokod för STA 13

Steg 1 : H-fas, som förut

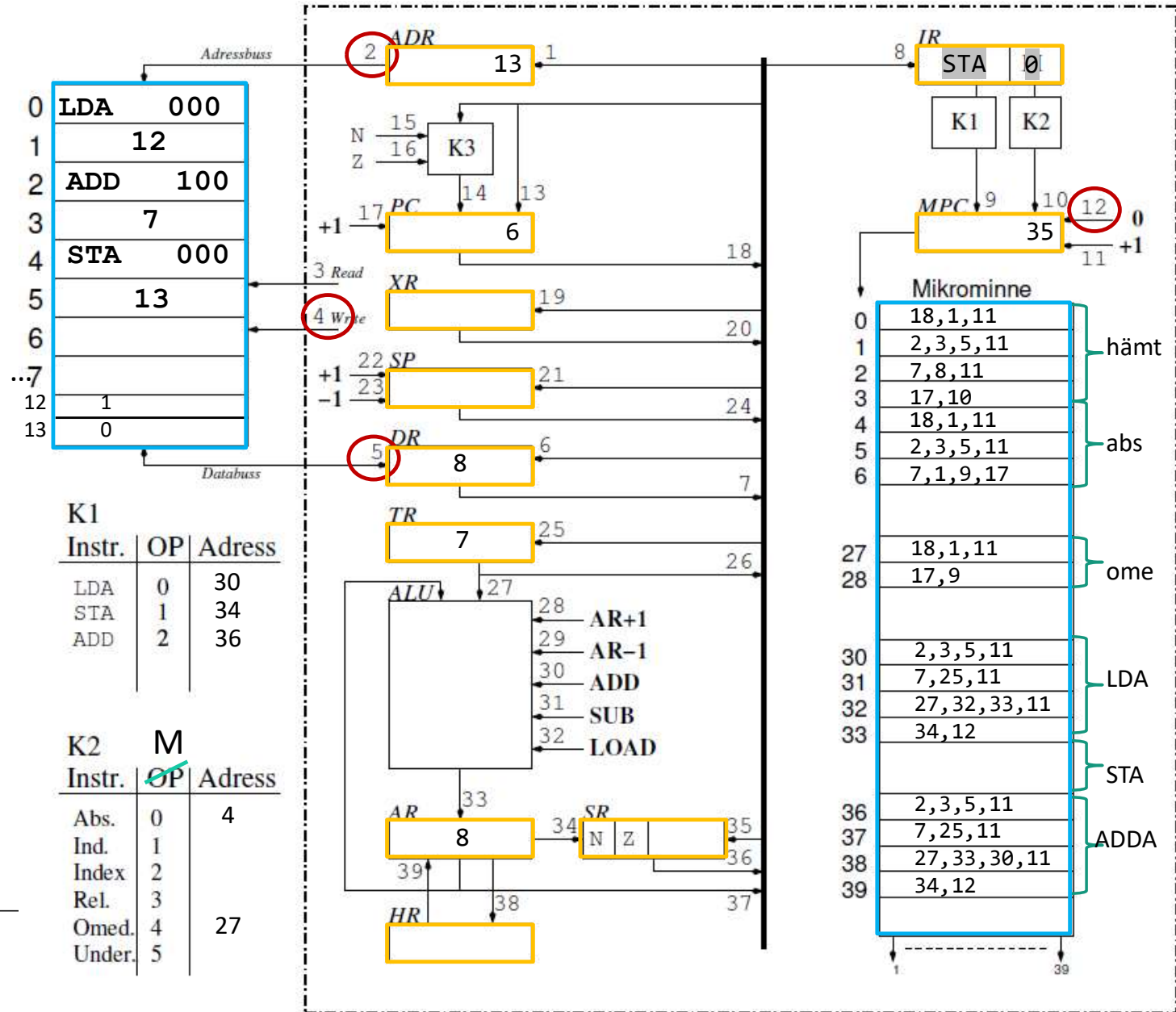
...
3: PC++, K2->mpc 17,10

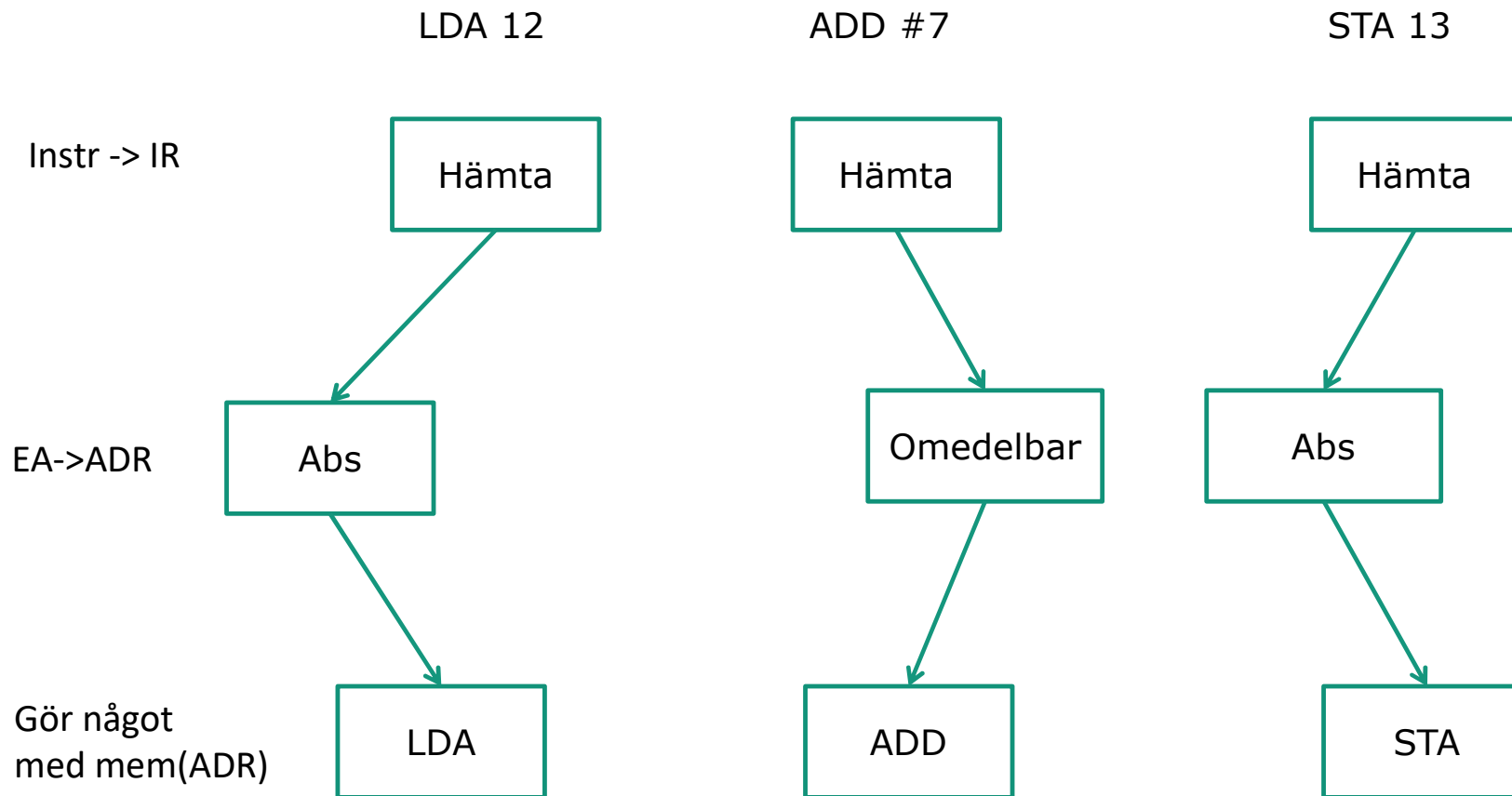
Steg 2 : A-fas, Absolut

...
6: dr->adr, K1->mpc, PC++ 7,1,9,17

Steg 3 : Exe, AR->M(ADR)

34: ar->dr, mpc++ 37,6,11
35: dr->M, 0->mpc 2,4,5,12





Mikrokod för LDA 3(X)

Indexerad adressering

Mikrokod för LDA 3(X)

M(XR+3) -> AR

Steg 1 : H-fas

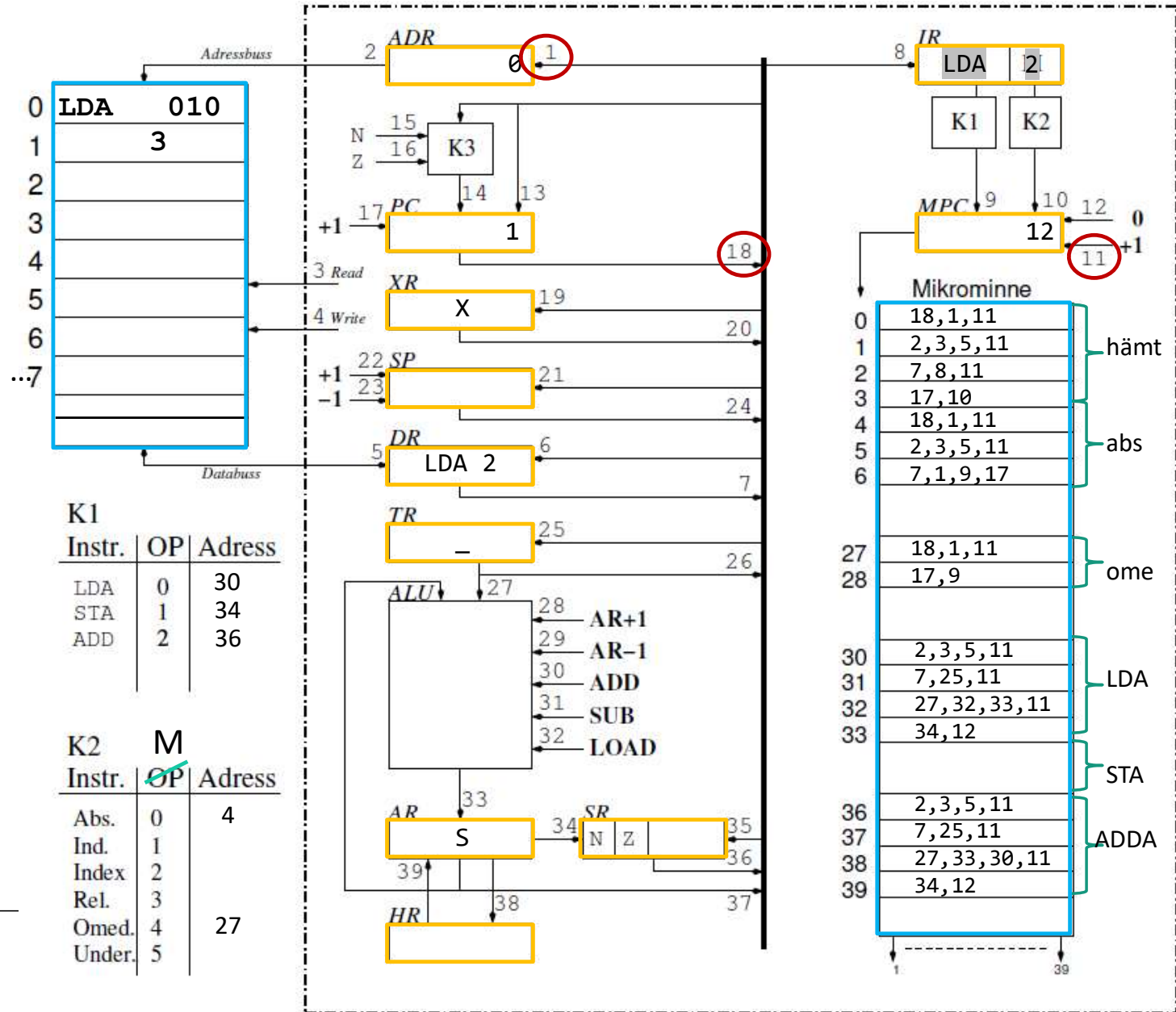
Finns redan

Steg 2 : A-fas, Indexerad

- 12: PC->ADR, PC++, MPC++
- 13: M->DR, XR->TR, MPC++
- 14: DR->TR, TR->AR, AR->HR, MPC++
- 15: AR+TR->AR, MPC++
- 16: HR->AR, AR->ADR, K1->MPC

Steg 3 : Exe, LDA

Finns redan



K1		
Instr.	OP	Address
LDA	0	30
STA	1	34
ADD	2	36

K2		
Instr.	OP	Address
Abs.	0	4
Ind.	1	
Index	2	
Rel.	3	
Omed.	4	27
Under.	5	

Mikrokod för LDA 3(X)

M(XR+3) -> AR

Steg 1 : H-fas

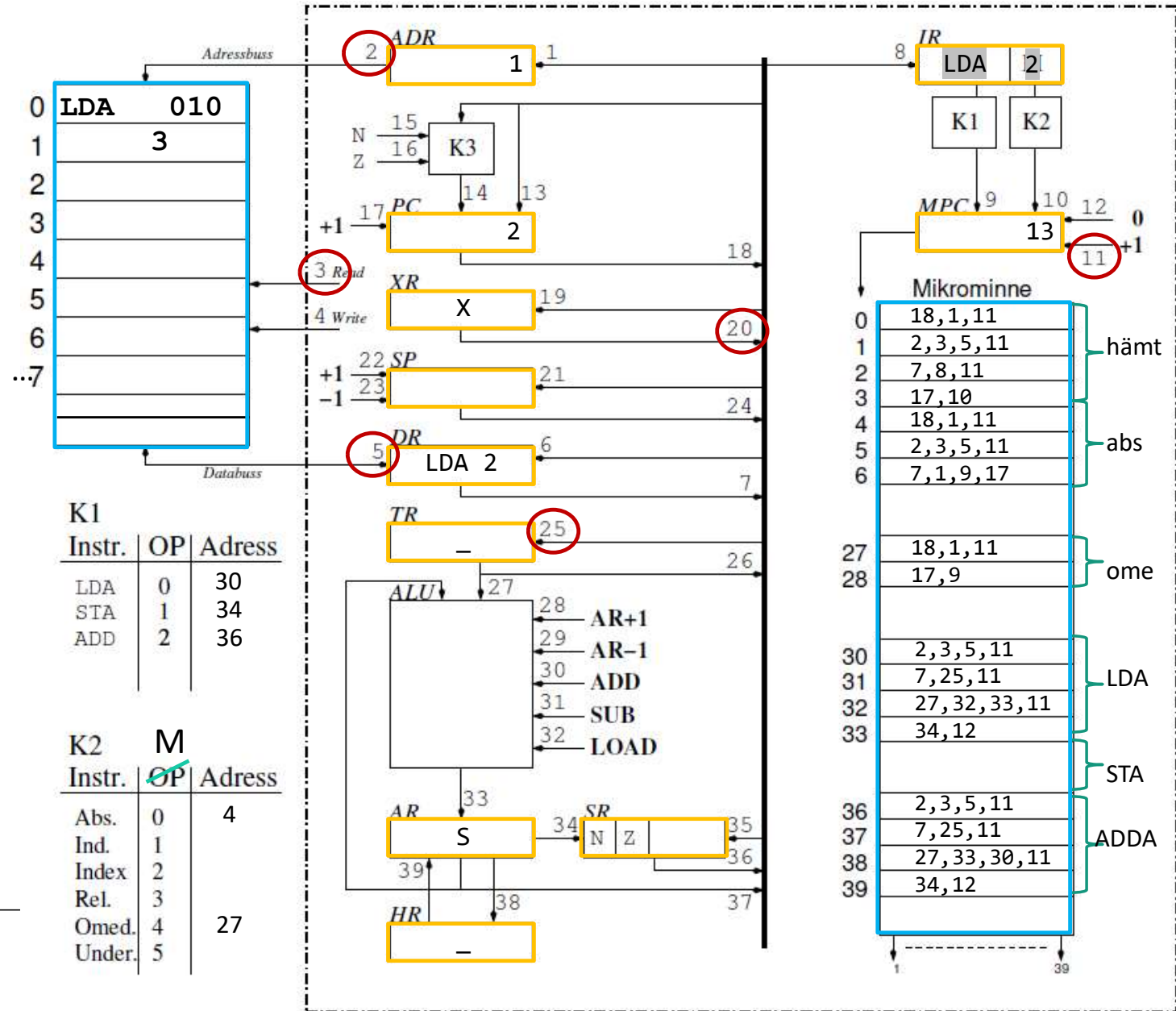
Finns redan

Steg 2 : A-fas, Indexerad

- 12: PC->ADR, PC++, MPC++
- 13: M->DR, XR->TR, MPC++
- 14: DR->TR, TR->AR, AR->HR, MPC++
- 15: AR+TR->AR, MPC++
- 16: HR->AR, AR->ADR, K1->MPC

Steg 3 : Exe, LDA

Finns redan



Mikrokod för LDA 3(X)

M(XR+3) -> AR

Steg 1 : H-fas

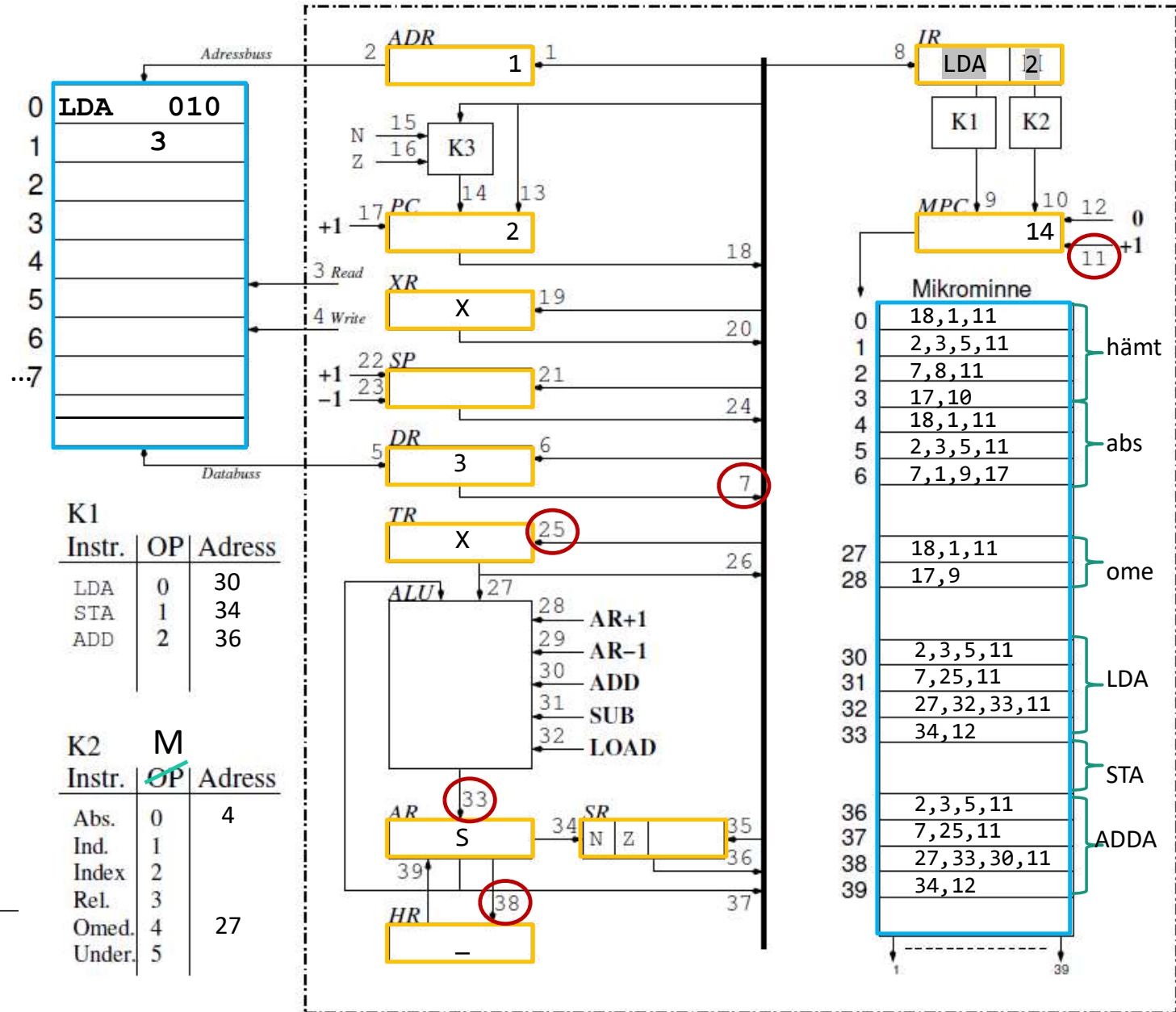
Finns redan

Steg 2 : A-fas, Indexerad

- 12: PC->ADR, PC++, MPC++
- 13: M->DR, XR->TR, MPC++
- 14: DR->TR, TR->AR, AR->HR, MPC++
- 15: AR+TR->AR, MPC++
- 16: HR->AR, AR->ADR, K1->MPC

Steg 3 : Exe, LDA

Finns redan



Mikrokod för LDA 3(X)

M(XR+3) -> AR

Steg 1 : H-fas

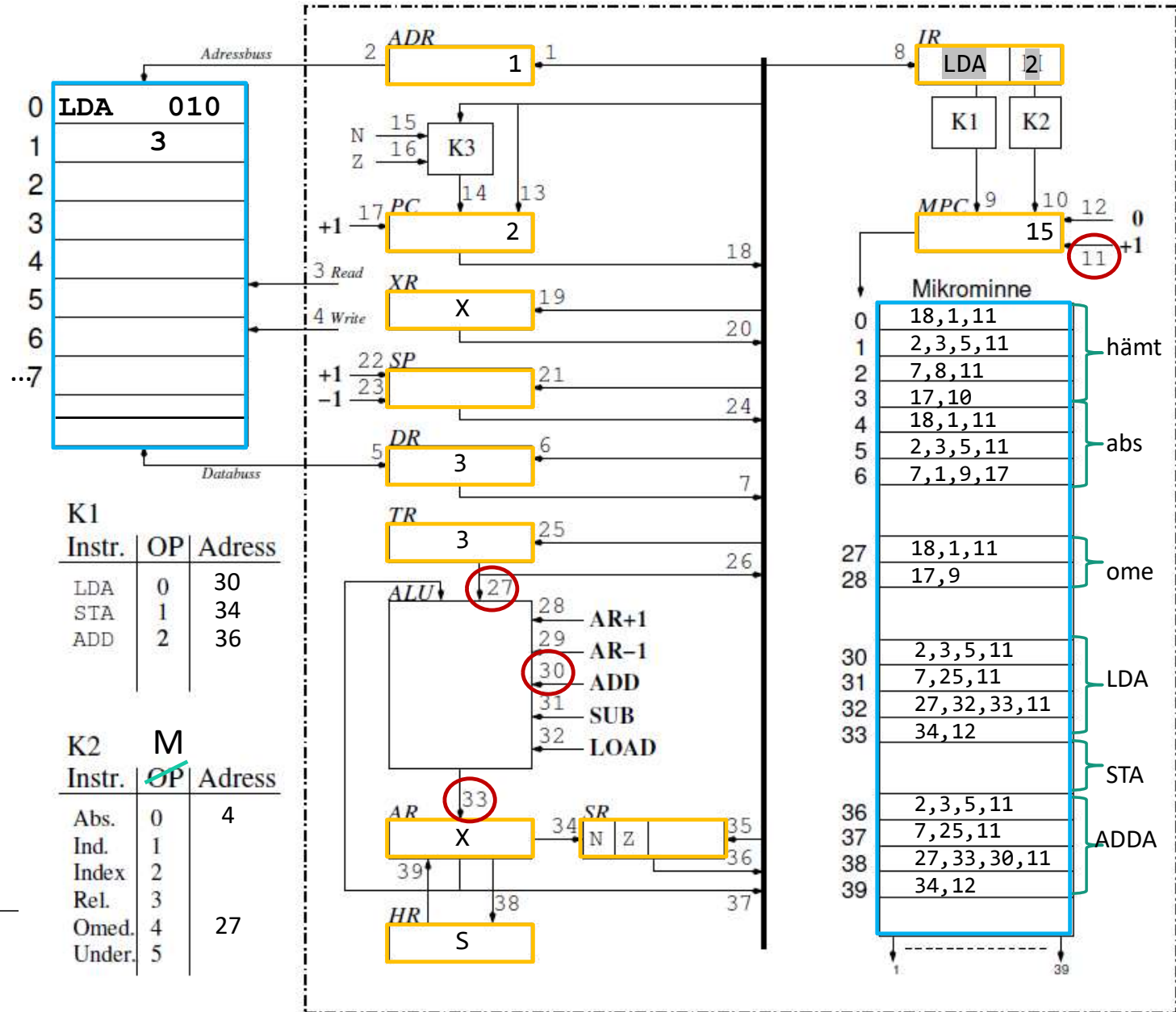
Finns redan

Steg 2 : A-fas, Indexerad

- 12: PC->ADR, PC++, MPC++
- 13: M->DR, XR->TR, MPC++
- 14: DR->TR, TR->AR, AR->HR, MPC++
- 15: **AR+TR->AR, MPC++**
- 16: HR->AR, AR->ADR, K1->MPC

Steg 3 : Exe, LDA

Finns redan



Mikrokod för LDA 3(X)

M(XR+3) -> AR

Steg 1 : H-fas

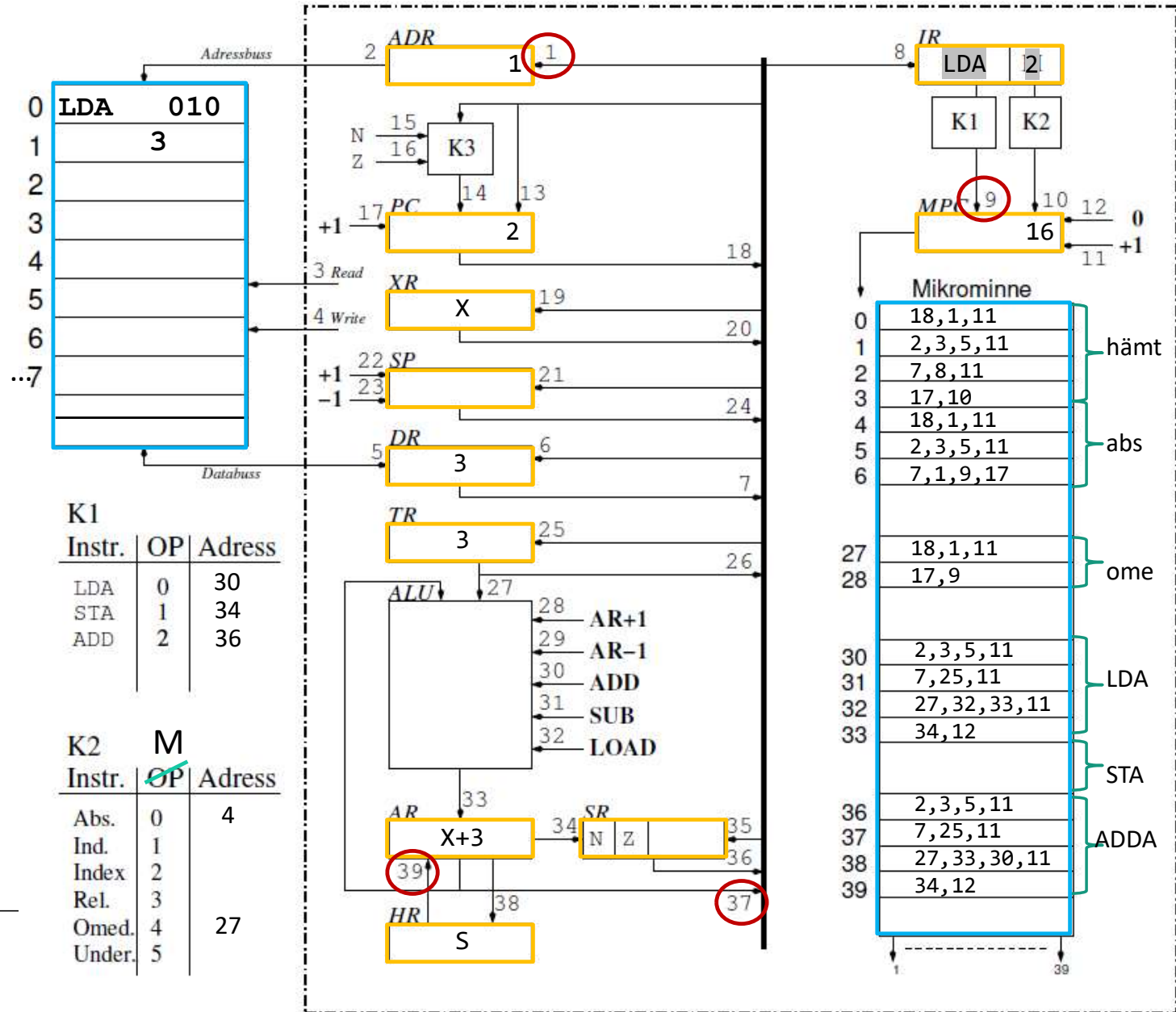
Finns redan

Steg 2 : A-fas, Indexerad

- 12: PC->ADR, PC++, MPC++
- 13: M->DR, XR->TR, MPC++
- 14: DR->TR, TR->AR, AR->HR, MPC++
- 15: AR+TR->AR, MPC++
- 16: HR->AR, AR->ADR, K1->MPC

Steg 3 : Exe, LDA

Finns redan



Mikrokod för LDA 3(X)

M(XR+3) -> AR

Steg 1 : H-fas

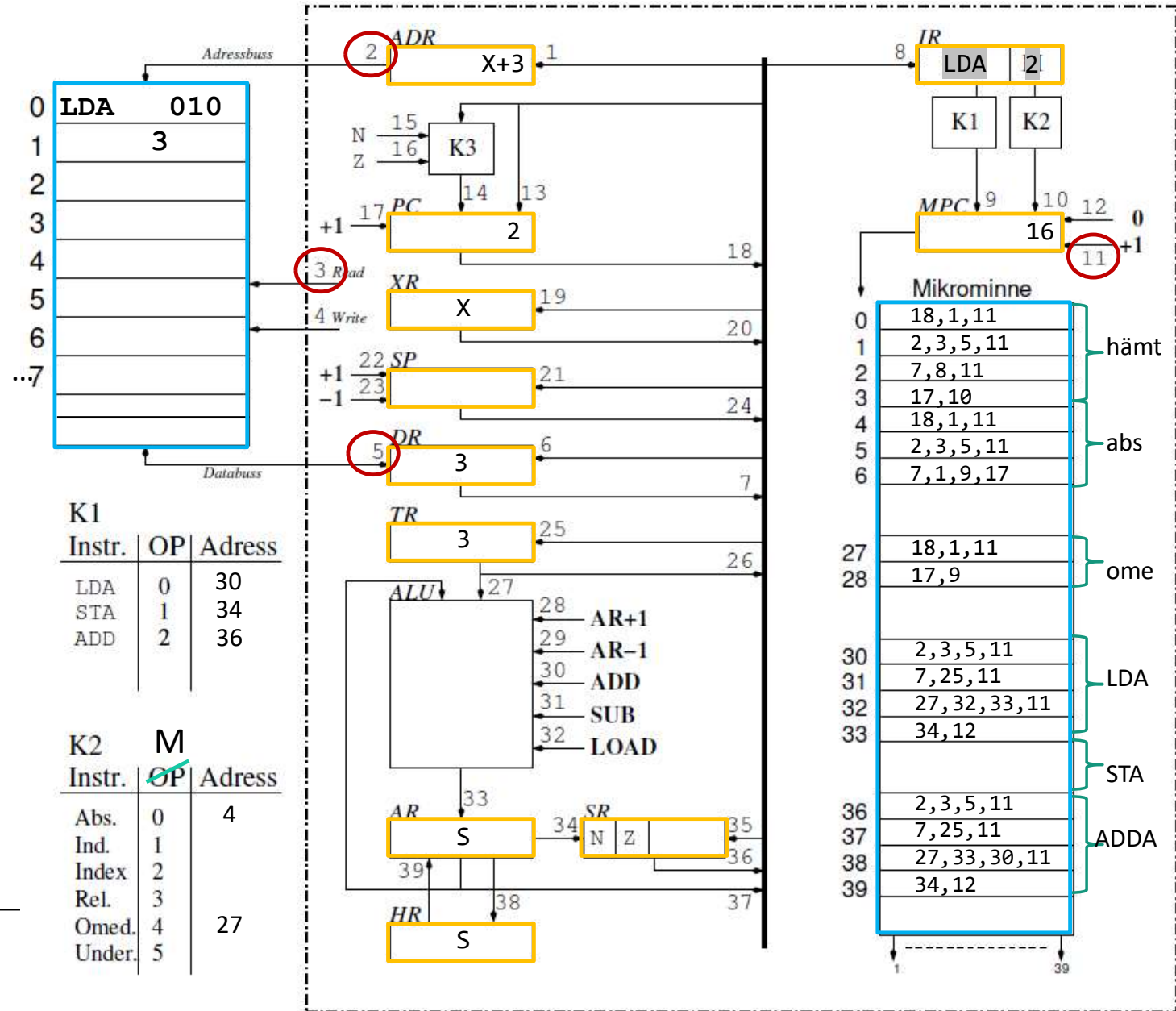
Finns redan

Steg 2 : A-fas, Indexerad

- 12: PC->ADR, PC++, MPC++
- 13: M->DR, XR->TR, MPC++
- 14: DR->TR, TR->AR, AR->HR, MPC++
- 15: AR+TR->AR, MPC++
- 16: HR->AR, AR->ADR, K1->MPC

Steg 3 : Exe, LDA

Finns redan



Mikrokod för INCA

Underförstådd adressering

Mikrokod för INCA

AR+1 -> AR

Steg 1 : H-fas

Finns redan

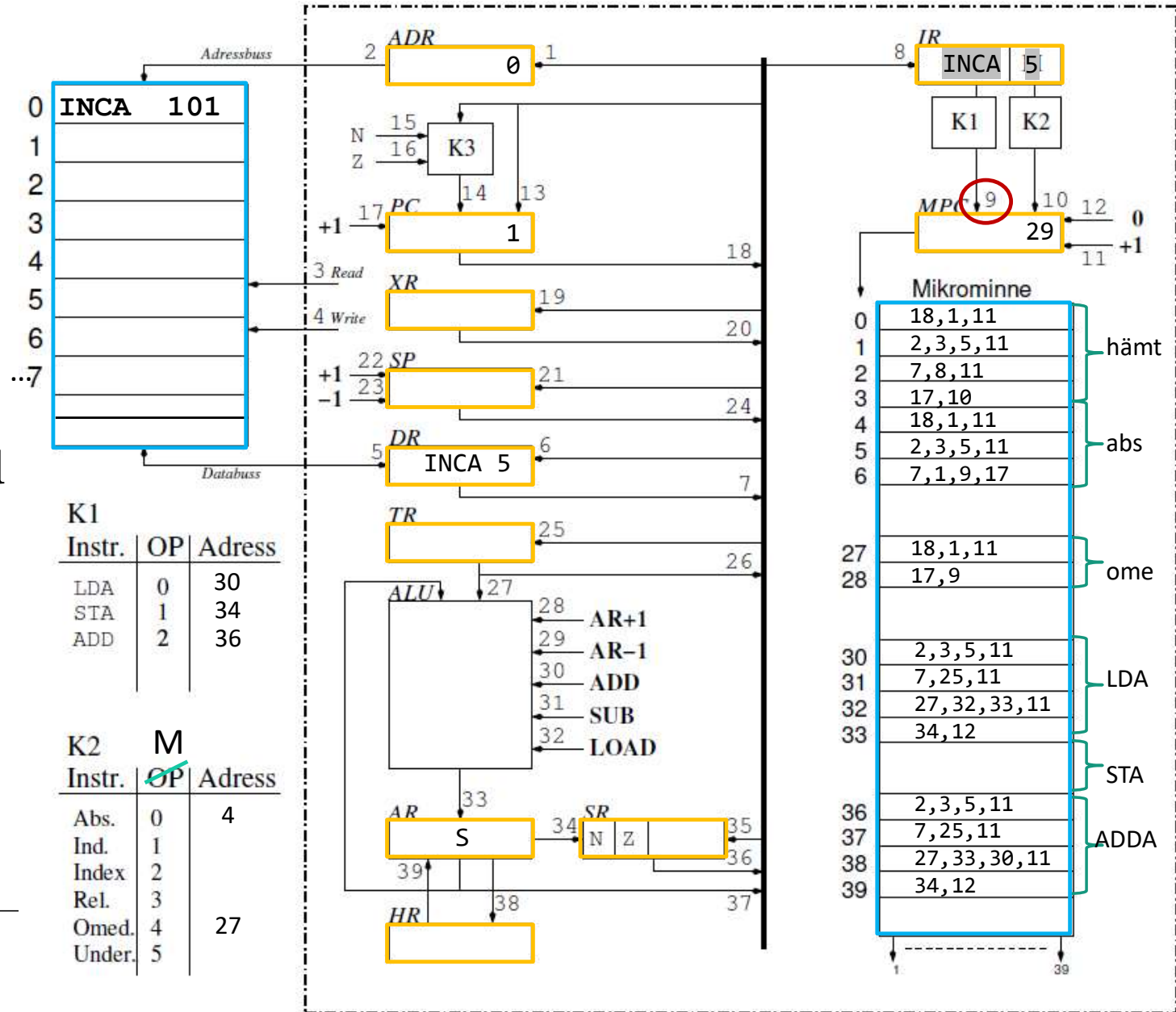
Steg 2 : A-fas, Underförstådd

29: K1->MPC

Steg 3 : Exe, INCA

44: AR+1->AR, MPC++

45: status, 0->MPC



K1		
Instr.	OP	Address
LDA	0	30
STA	1	34
ADD	2	36

K2			M
Instr.	OP	Address	
Abs.	0	4	
Ind.	1		
Index	2		
Rel.	3		
Omed.	4	27	
Under.	5		

Mikrokod för INCA

AR+1 -> AR

Steg 1 : H-fas

Finns redan

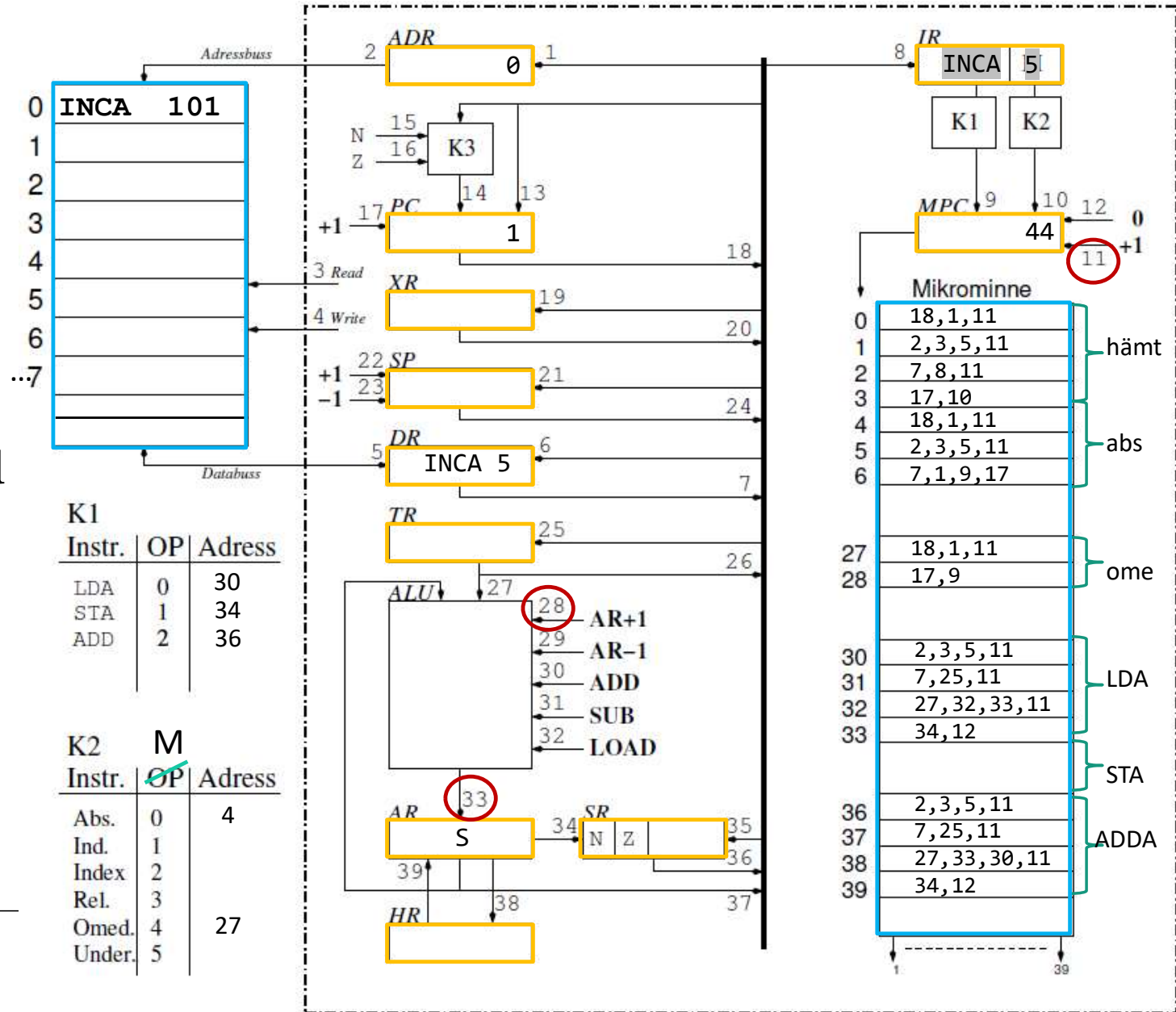
Steg 2 : A-fas, Underförstådd

29: K1->MPC

Steg 3 : Exe, INCA

44: AR+1->AR, MPC++

45: status, 0->MPC



K1		
Instr.	OP	Address
LDA	0	30
STA	1	34
ADD	2	36

K2		
Instr.	OP M	Address
Abs.	0	4
Ind.	1	
Index	2	
Rel.	3	
Omed.	4	27
Under.	5	

Mikrokod för INCA

AR+1 -> AR

Steg 1 : H-fas

Finns redan

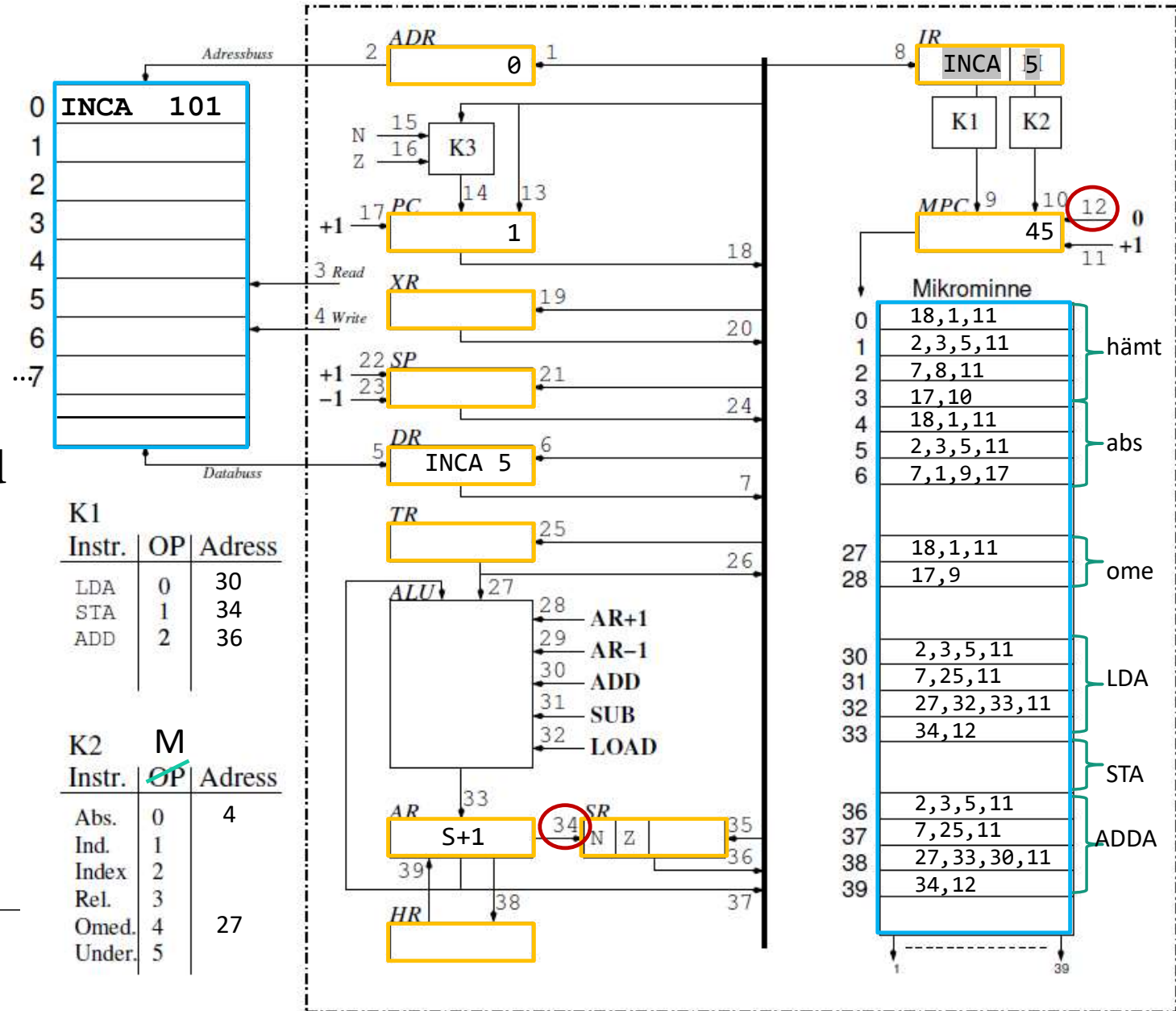
Steg 2 : A-fas, Underförstådd

29: K1->MPC

Steg 3 : Exe, INCA

44: AR+1->AR, MPC++

45: status, 0->MPC



Mikrokod för LDA (3)

Indirekt adressering

Mikrokod för LDA (3)

M(M(3)) -> AR

Steg 1 : H-fas

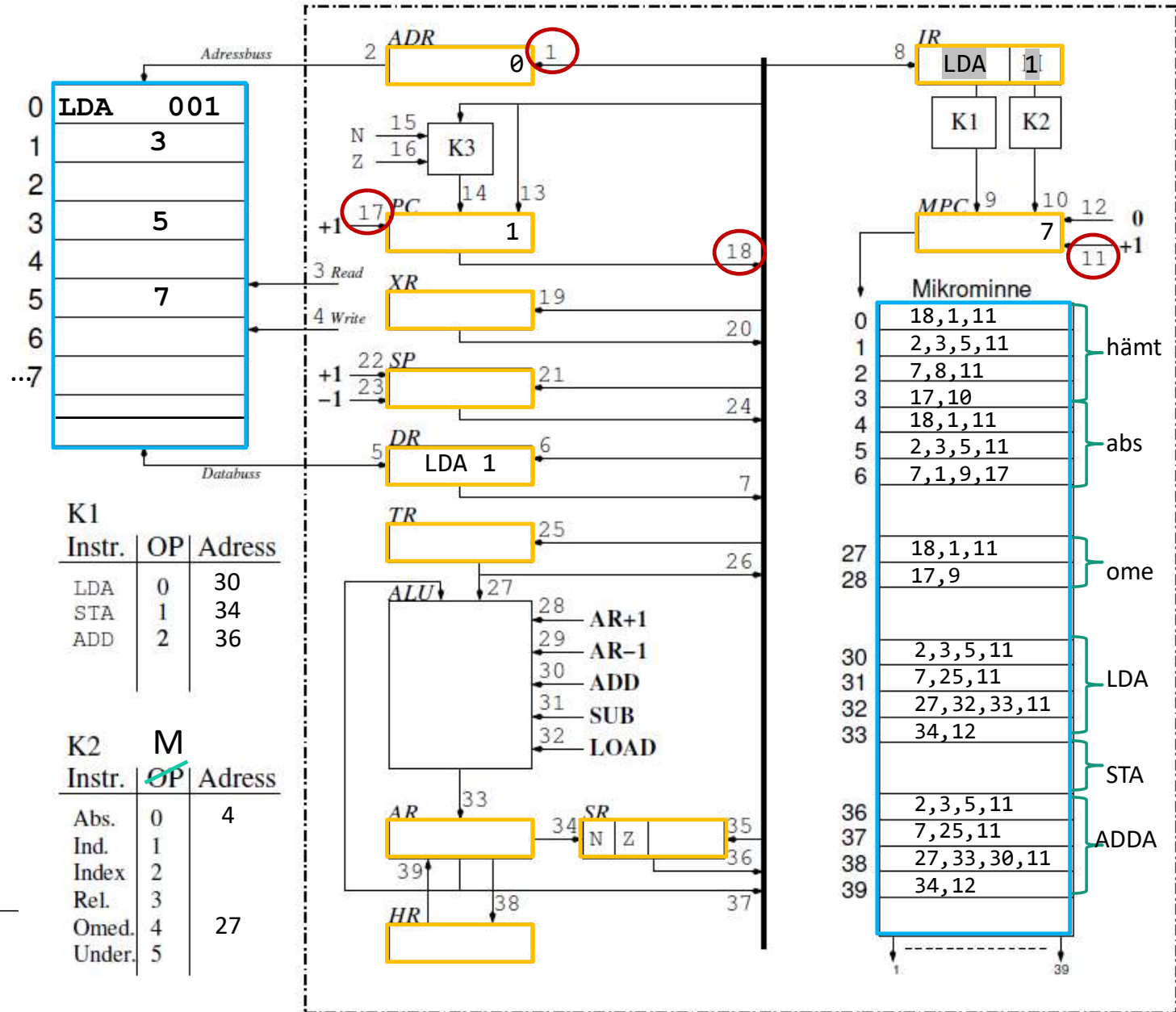
Finns redan

Steg 2 : A-fas, Indirekt

- 7: PC->ADR, PC++, MPC++
- 8: M->DR, MPC++
- 9: DR->ADR, MPC++
- 10: M->DR, MPC++
- 11: DR->ADR, K1->MPC

Steg 3 : Exe, LDA

Finns redan



Mikrokod för LDA (3)

M(M(3)) -> AR

Steg 1 : H-fas

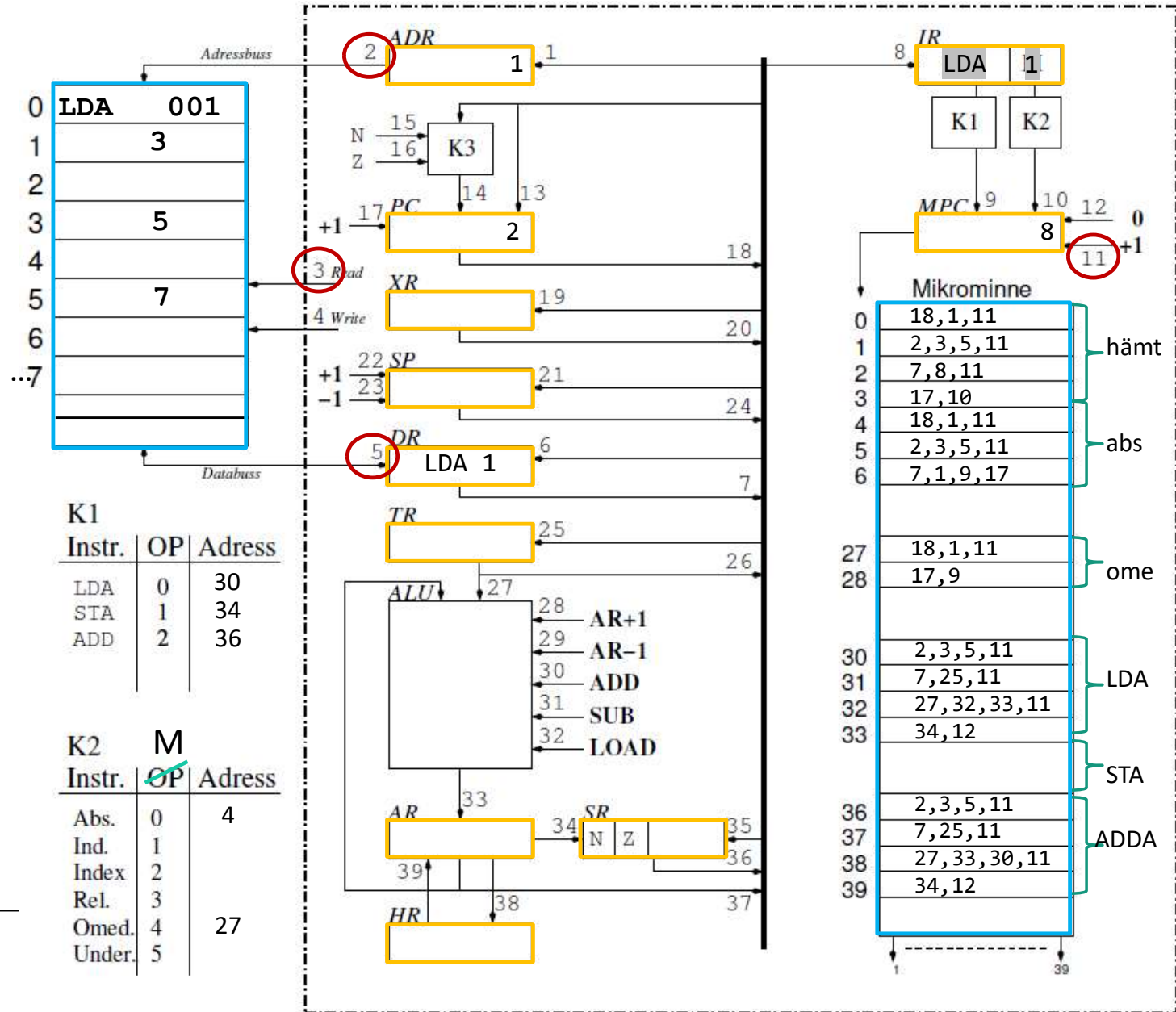
Finns redan

Steg 2 : A-fas, Indirekt

- 7: PC->ADR, PC++, MPC++
- 8: M->DR, MPC++
- 9: DR->ADR, MPC++
- 10: M->DR, MPC++
- 11: DR->ADR, K1->MPC

Steg 3 : Exe, LDA

Finns redan



Mikrokod för LDA (3)

M(M(3)) -> AR

Steget 1 : H-fas

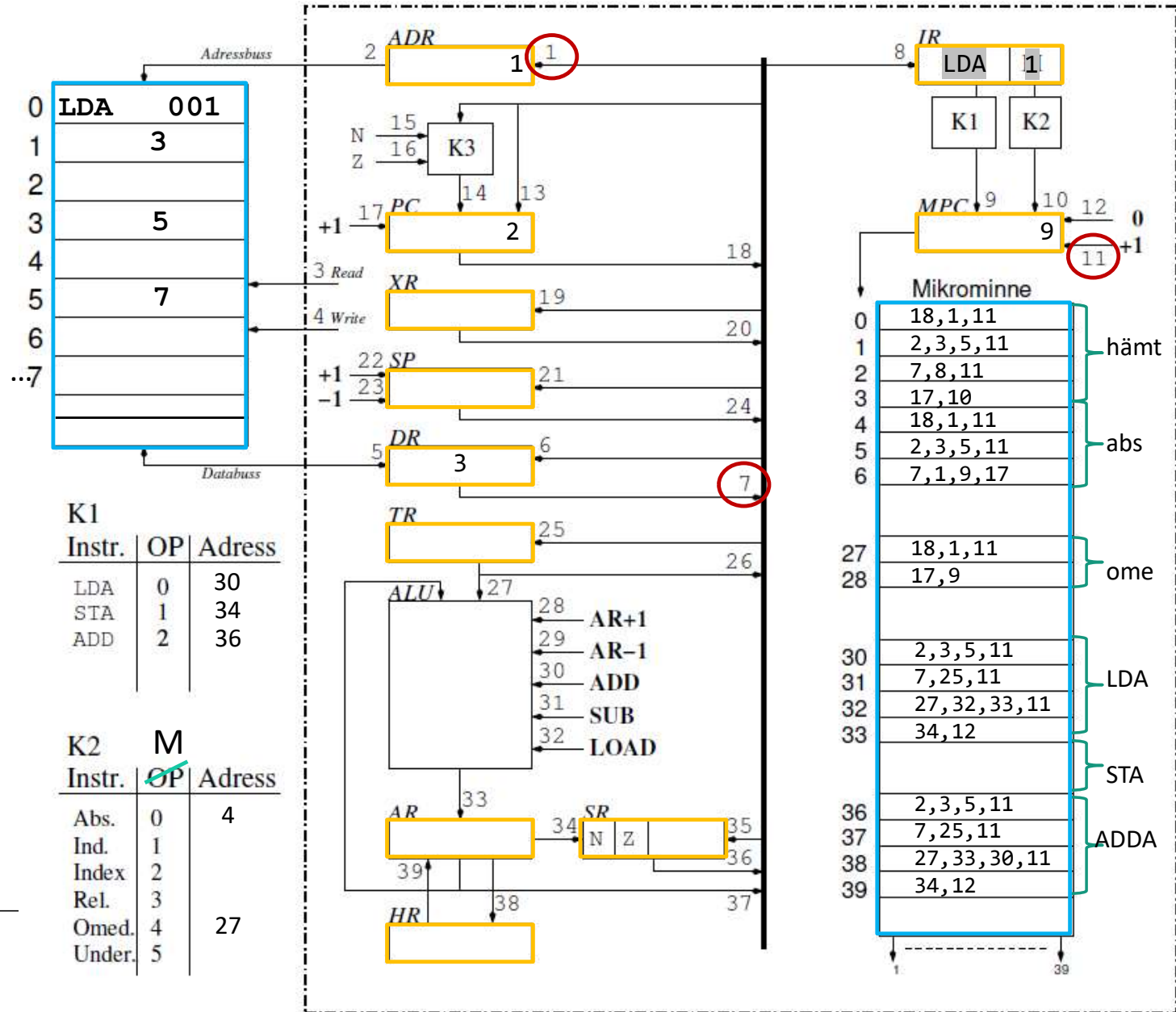
Finns redan

Steget 2 : A-fas, Indirekt

- 7: PC->ADR, PC++, MPC++
- 8: M->DR, MPC++
- 9: DR->ADR, MPC++
- 10: M->DR, MPC++
- 11: DR->ADR, K1->MPC

Steget 3 : Exe, LDA

Finns redan



Mikrokod för LDA (3)

M(M(3)) -> AR

Steg 1 : H-fas

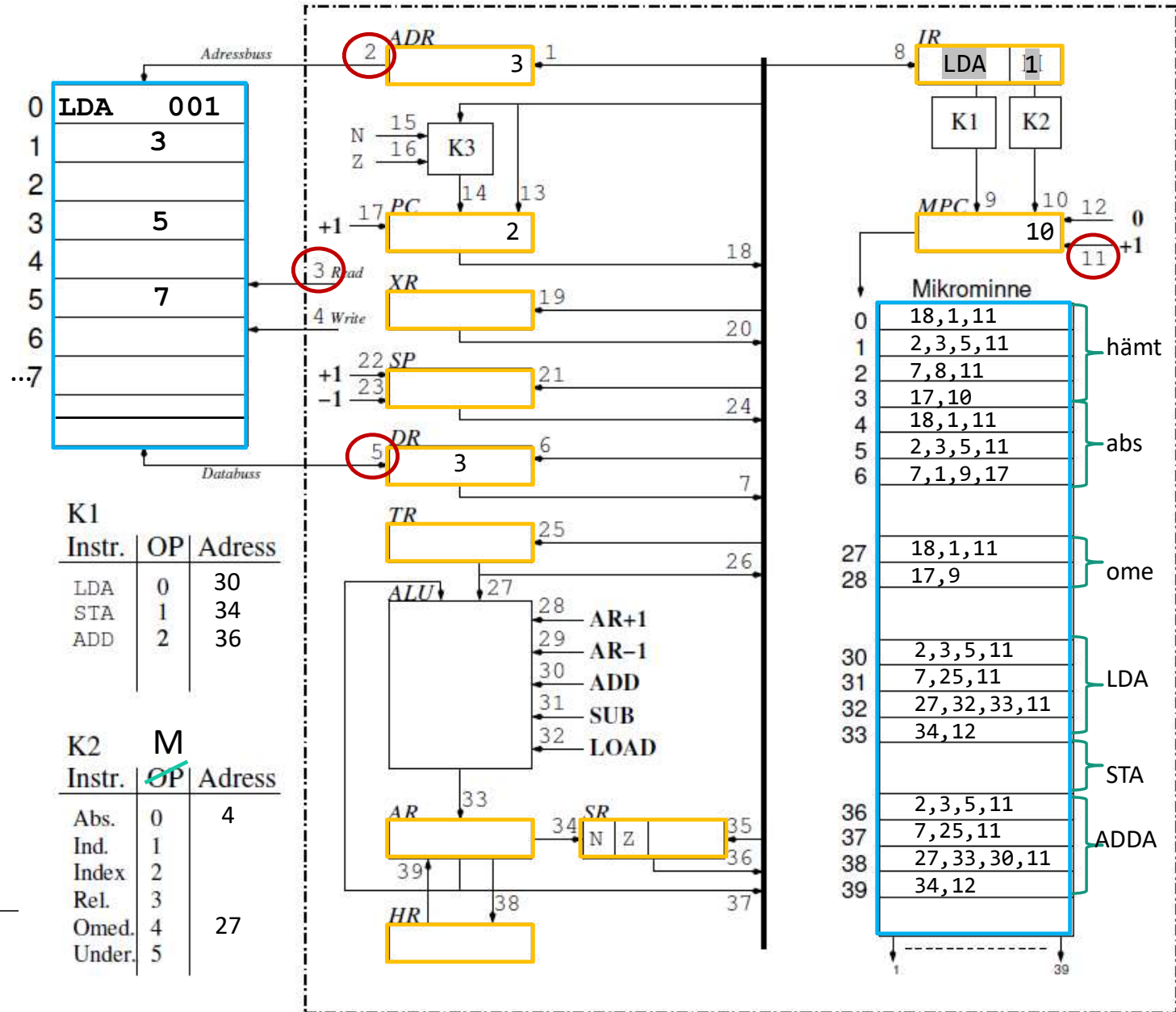
Finns redan

Steg 2 : A-fas, Indirekt

- 7: PC->ADR, PC++, MPC++
- 8: M->DR, MPC++
- 9: DR->ADR, MPC++
- 10: M->DR, MPC++
- 11: DR->ADR, K1->MPC

Steg 3 : Exe, LDA

Finns redan



K1		
Instr.	OP	Address
LDA	0	30
STA	1	34
ADD	2	36

K2		
Instr.	OP	Address
Abs.	0	4
Ind.	1	
Index	2	
Rel.	3	
Omed.	4	27
Under.	5	

Mikrokod för LDA (3)

M(M(3)) -> AR

Steg 1 : H-fas

Finns redan

Steg 2 : A-fas, Indirekt

7: PC->ADR, PC++, MPC++

8: M->DR, MPC++

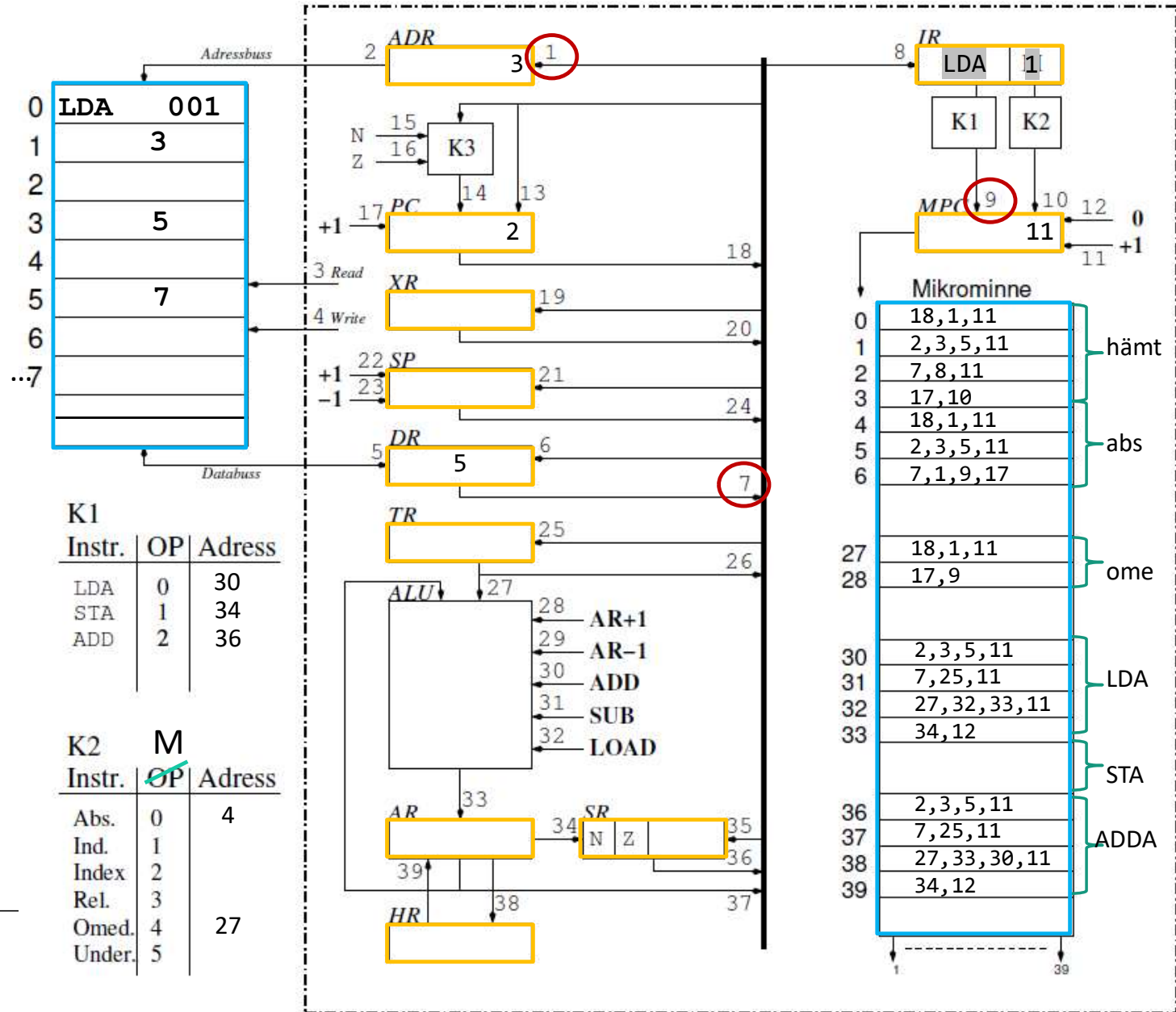
9: DR->ADR, MPC++

10: M->DR, MPC++

11: DR->ADR, K1->MPC

Steg 3 : Exe, LDA

Finns redan



Mikrokod för LDA (3)

M(M(3)) -> AR

Steg 1 : H-fas

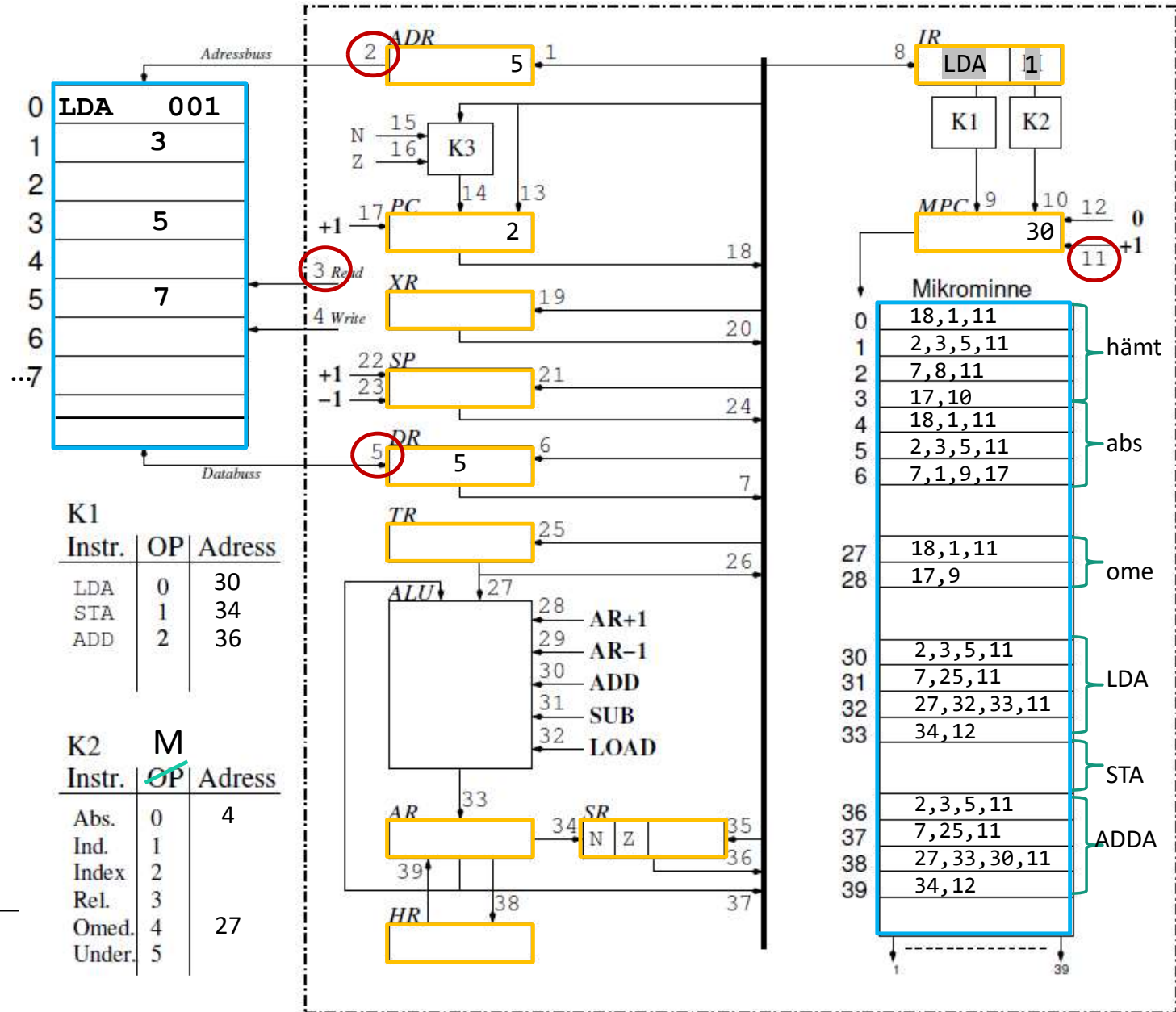
Finns redan

Steg 2 : A-fas, Indirekt

- 7: PC->ADR, PC++, MPC++
- 8: M->DR, MPC++
- 9: DR->ADR, MPC++
- 10: M->DR, MPC++
- 11: DR->ADR, K1->MPC

Steg 3 : Exe, LDA

Finns redan



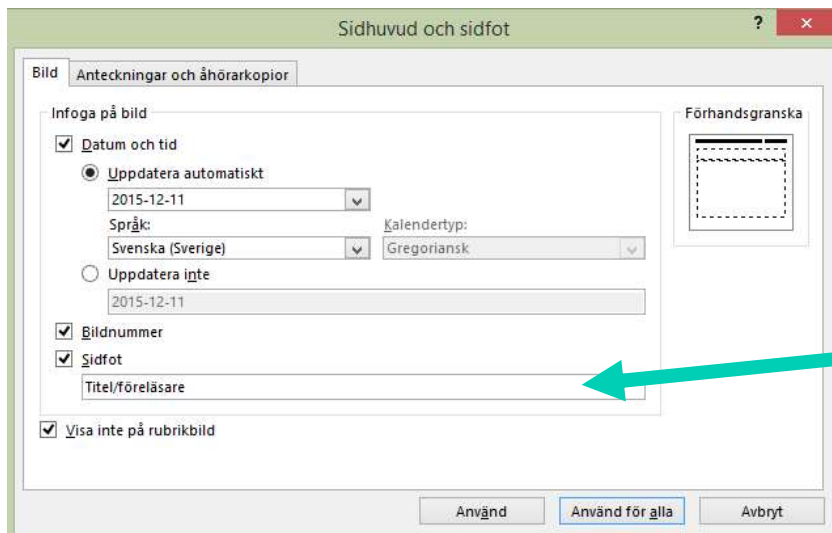
Anders Nilsson

www.liu.se

Lägga till text eller ändra i sidhuvud -fot

Klicka på fliken Infoga, sedan på Sidhuvud/sidfot

Följande visas:



- I rutan för sidfot ändrar du titel/föreläsare till aktuella värden
- Du kan också välja att kryssa bort detaljer som du inte vill ha med
- Klicka på "Använd för alla" så kommer alla bilder att uppdateras. Klickar du på "Använd" gäller det endast aktuell bild

Rubrik på en rad med punktlista

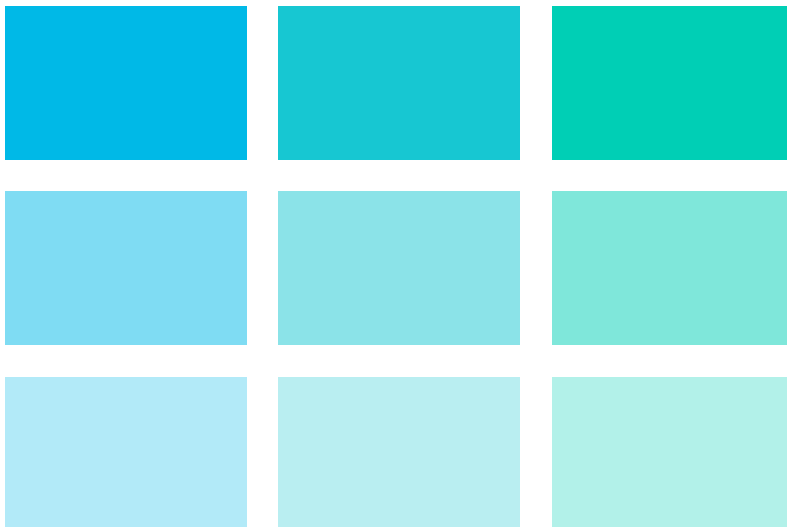
Lång rubrik på en rad med bild till höger

Avsnittsbild

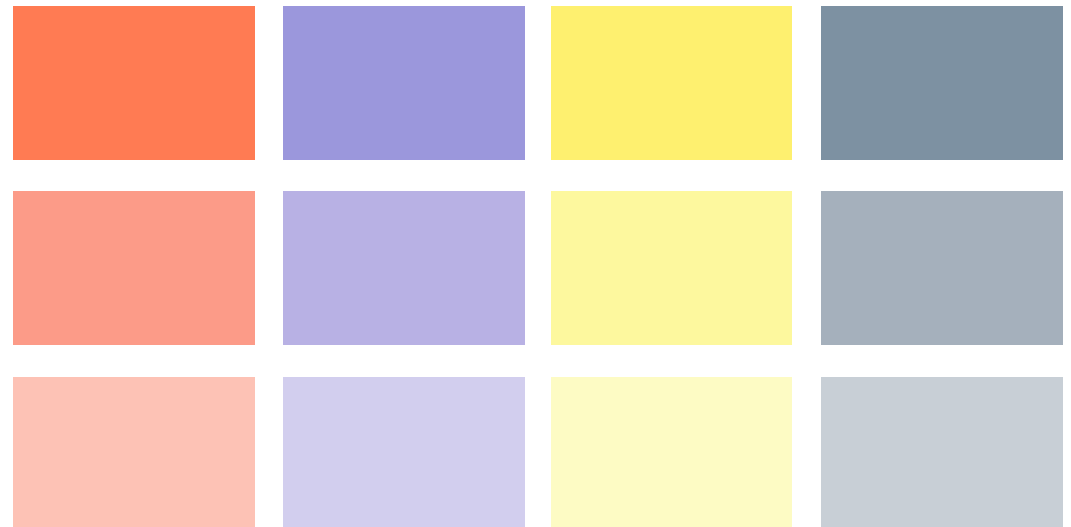
Använd dessa för att markera övergång till ett nytt avsnitt/ämne i långa presentationer

Färger att använda till t.ex. diagram

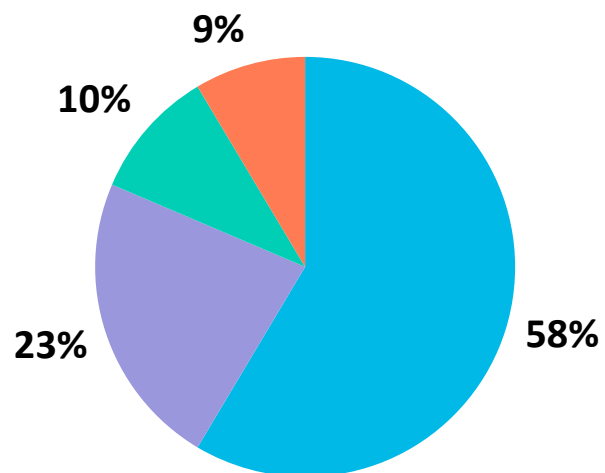
Basfärger



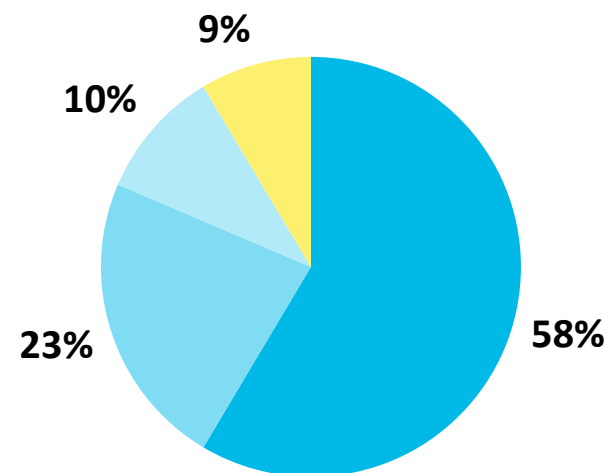
Komplementfärger



Exempel diagram

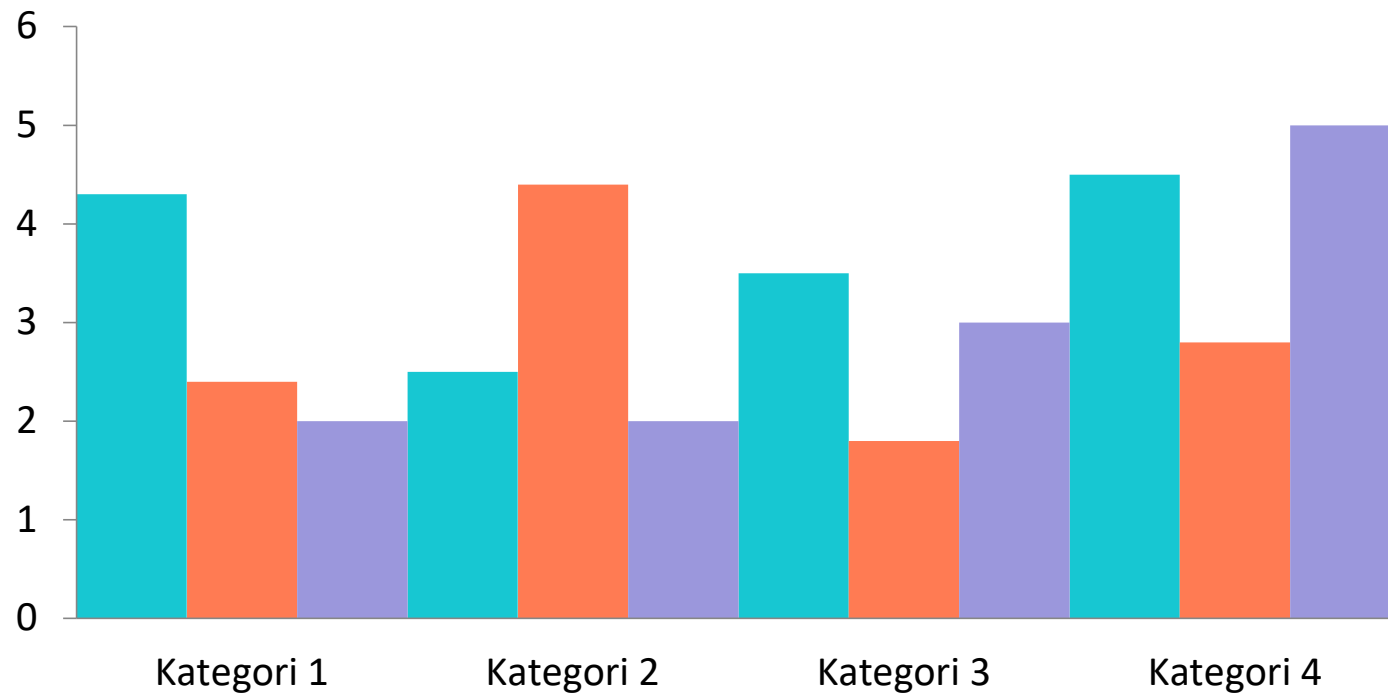


Kategori 1	3 357
Kategori 2	1 147
Kategori 3	886
Kategori 4	562

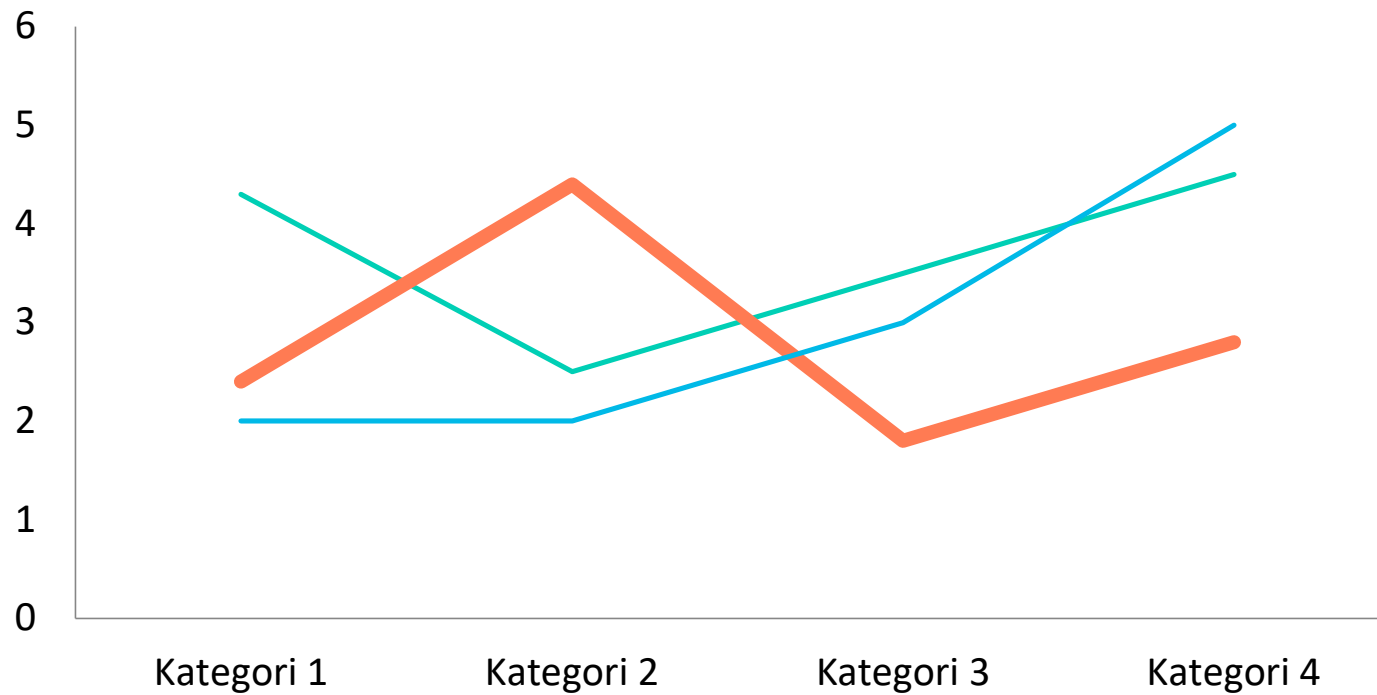


Kategori 1	3 357
Kategori 2	1 147
Kategori 3	886
Kategori 4	562

Exempel diagram



Exempel diagram



www.liu.se