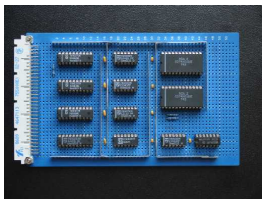
Digital- och datorteknik



Föreläsning #23

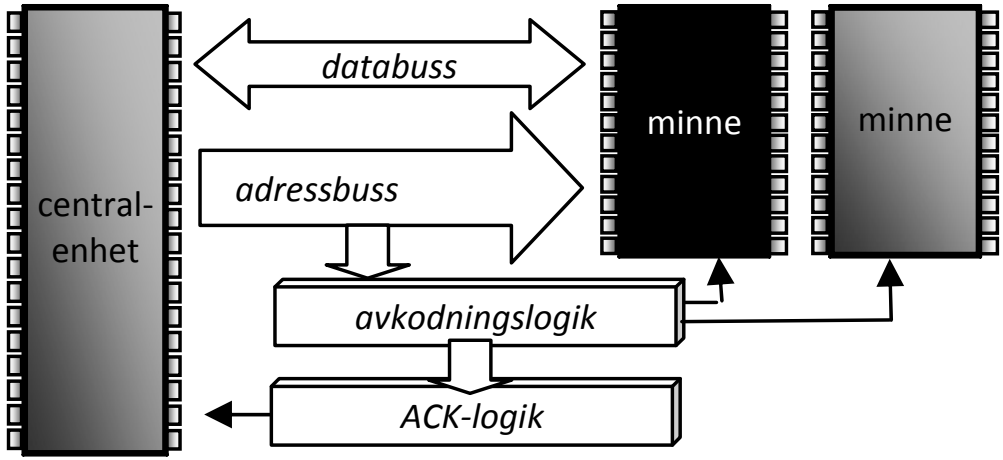
Biträdande professor Jan Jonsson

Institutionen för data- och informationsteknik
Chalmers tekniska högskola

Adressavkodning

Översikt

När flera minnesmoduler placeras i processorns adressrum ansluts modulernas adressingångar till motsvarande ledningar i adressbussen. Övriga adressledningar i bussen används för att välja rätt minnesmodul genom att man bildar selektorsignaler (”chip select”) med hjälp av dem via logik för adressavkodning.



Adressavkodning

Översikt

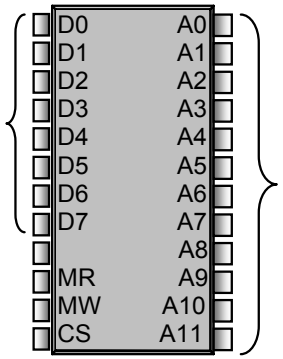
Som målsystem i våra exempel på adressavkodning använder vi ett mindre inbyggt datorsystem med en centralenhet med 64 kbyte adressrum (16-bitars adressbuss) och 8-bitars databuss.

Som minnesmoduler använder vi ROM och RWM i storlekarna 4 kbyte – 32 kbyte. Exempel på 4 kbyte minnesmodul ges nedan.

Minnet har 8-bitars ordbredd, precis som centralenheten

MR och MW är *insignaler* till minneskapslarna

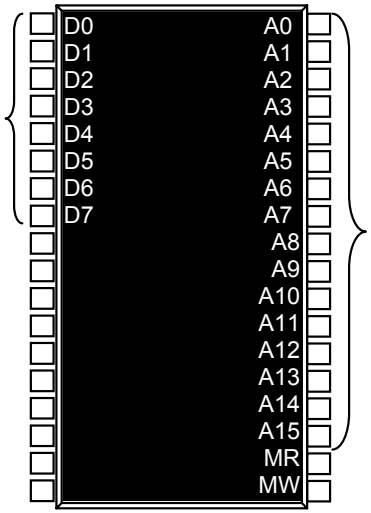
CS (Chip Select) är en *insignal* till kapseln som måste aktiveras för åtkomst minnet



Minnet innehåller 4 kB (2^{12}) bytes och har därmed 12 adress-ledningar

A0-A11

Databuss, 8 signaler, betecknas D0-D7



Adressbuss, 16 signaler, betecknas A0-A15

Styrsignaler MR och MW

Adressavkodning

Placering av minnesmoduler

När man gör adressavkodningen för en dator utgår man ifrån hur stort minne och vilken typ av minne som behövs. Därefter avgörs hur stora minnesmoduler som skall användas. Nästa steg är att placera in de olika modulerna i centralenhetens adressrum. Var varje minnesmodul placeras beror på vad modulen skall innehålla samt vilka konventioner som används för den typen av dator.

För inbyggda datorsystem gäller normalt att:

- ROM placeras där man vill ha data som inte skall förändras, t ex programkod, textsträngar, vektorer för reset- och interrupt.
- RWM placeras där man vill ha data som skall kunna förändras av programmet, t ex variabler och stack.

Adressavkodning

Placering av minnesmoduler (storlekar)

4 kbyte minnesmodul (12 adressledningar: $A_0 – A_{11}$)

- Storlek: 1000_{16} bytes
- Exempel på adressområde: $1000_{16} – 1FFF_{16}$

8 kbyte minnesmodul (13 adressledningar: $A_0 – A_{12}$)

- Storlek: 2000_{16} bytes
- Exempel på adressområde: $2000_{16} – 3FFF_{16}$

16 kbyte minnesmodul (14 adressledningar: $A_0 – A_{13}$)

- Storlek: 4000_{16} bytes
- Exempel på adressområde: $4000_{16} – 7FFF_{16}$

32 kbyte minnesmodul (15 adressledningar: $A_0 – A_{14}$)

- Storlek: 8000_{16} bytes
- Exempel på adressområde: $8000_{16} – FFFF_{16}$

Adressavkodning

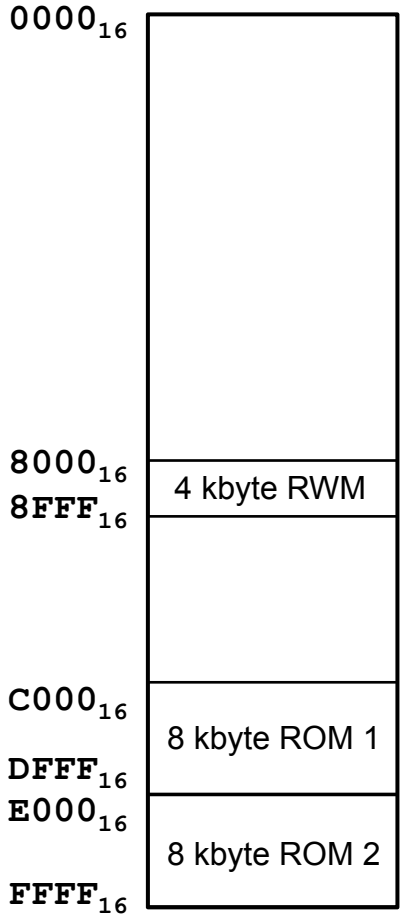
Placering av minnesmoduler (minneskarta)

För att illustrera den valda placeringen av minnesmoduler och periferikretsar används oftast en minneskarta (”memory map”).

För ett datorsystem med 64 kbyte adressrum är det totala adressområdet 0000_{16} – $FFFF_{16}$.

Exempel: system med 20 kbyte minne:

- 4 kbyte RWM
  4 kbyte placerat på 8000_{16} – $8FFF_{16}$
- 16 kbyte ROM
  8 kbyte placerat på $C000_{16}$ – $DFFF_{16}$
  8 kbyte placerat på $E000_{16}$ – $FFFF_{16}$



Adressavkodning

Placering av minnesmoduler (minneskarta)

De olika modulerna skall adresseras inom unika adressintervall (områden). Inom respektive minnesmoduls adressområde är ett litet antal av de mest signifikanta adressbitarna konstanta.

De konstanta bitarna i vårt exempel är markerade med en ram.

Modul	***Adress***	***Adressignal nr***															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RWM	8000H	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	8FFFH	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
ROM nr 1	C000H	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	DFFFH	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM nr 2	E000H	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	FFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Adressavkodning

Placering av minnesmoduler (minneskarta)

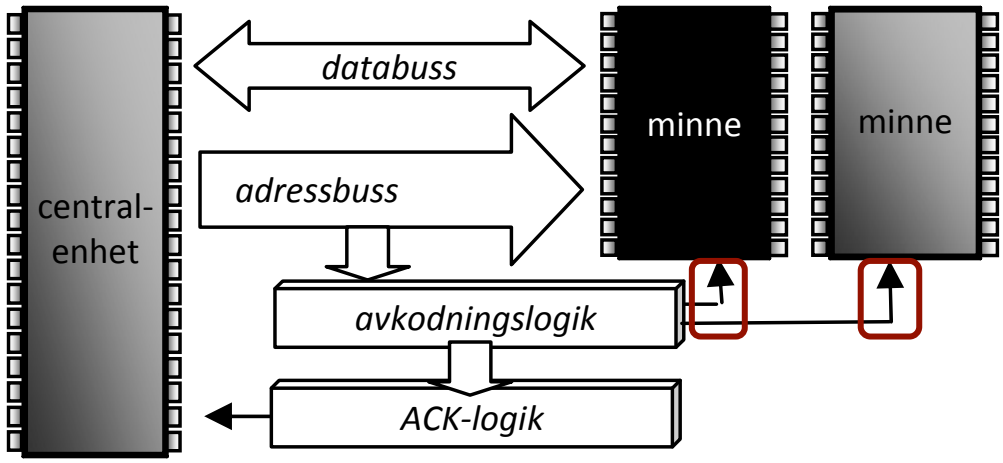
De konstanta adressbitarna har samma (unika) värden för alla adresser inom området och kan ses som en ID-kod för området. ID-koden kan användas för att "peka ut" respektive minnesmodul, när processorn skall läsa eller skriva i just denna modul.

Modul	***Adress***	***Adressignal nr***															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RWM	8000H	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	8FFFH	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
ROM nr 1	C000H	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	DFFFH	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM nr 2	E000H	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	FFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Adressavkodning

Placering av minnesmoduler (avkodning)

Adressavkodningen (”chip select”-logiken) sker i logikblock med de mest signifikanta adressbitarna samt, när så behövs, signalen MR som insignaler. Utsignalerna består av "chip select"-signaler till de olika modulerna.

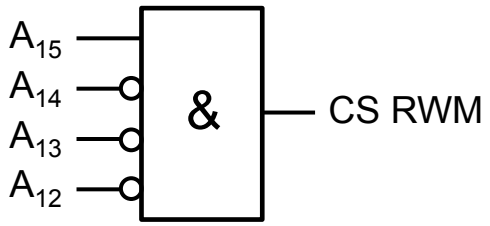


Adressavkodning

Placering av minnesmoduler (avkodning)

Vi betraktar nu adressavkodningen för RWM-modulen i vårt exempel. När modulstorleken är 4 kbyte = 2^{12} byte används 12 stycken adressbitar för intern adressavkodning inom modulen. De övriga fyra adressbitarna, $A_{15} – A_{12}$, används för att bestämma var någonstans i processorns adressrum modulen placeras.

RWM-modulen placeras i adressintervallet $8000_{16} – 8FFF_H$. I detta intervall har adressbitarna $A_{15} – A_{12}$ värdena 1000_2. Modulen kan då pekas ut med en ”chip select”-signal, som bildas på följande sätt:

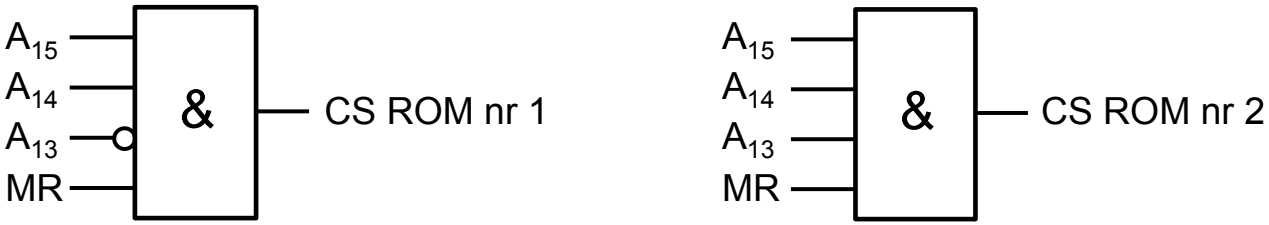


Adressavkodning

Placering av minnesmoduler (avkodning)

På ett liknande sätt kan vi ta fram adressavkodningen för ROM-modulerna i vårt exempel. De två modulerna placeras i adressintervallet $C000_{16}$ – $DFFF_H$ (ROM nr 1) respektive $E000_{16}$ – $FFFF_H$ (ROM nr 2).

Här används även MR-signalen vid bildandet av ”chip select”. Om MR inte används så skulle en eventuell skrivning i ROM (t ex på grund av ett programmeringsfel) innebära att både processorn och ROM-modulen släpper ut data på databussen och orsakar en ”busskollision”.



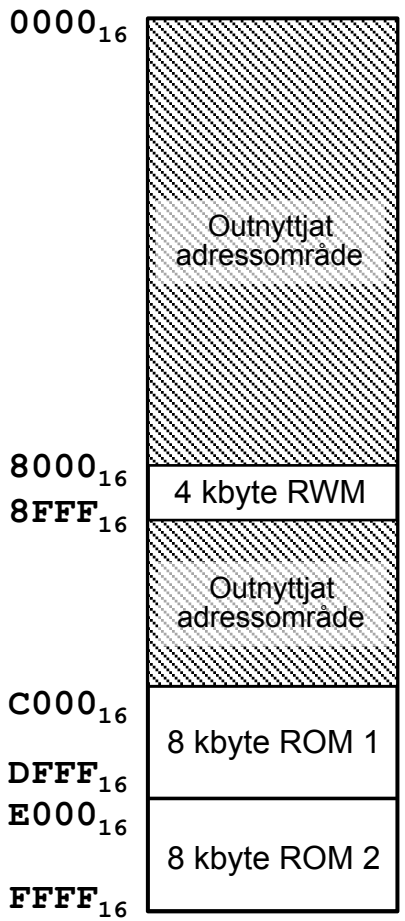
Adressavkodning

Fullständig adressavkodning

Vi har i vår lösning använt alla adressbitar som krävs för att aktivera en modul enbart inom dess avsedda adressområde. Detta kallas för fullständig adressavkodning.

Vid fullständig adressavkodning kommer adressrummet att utnyttjas maximalt. Vi kan exempelvis bygga ut med fler minnesmoduler eller lägga till periferikretsar, utan att behöva göra om den tidigare adressavkodningen.

Denna typ av avkodning kan dock i många fall kräva relativt komplicerade logiknät.

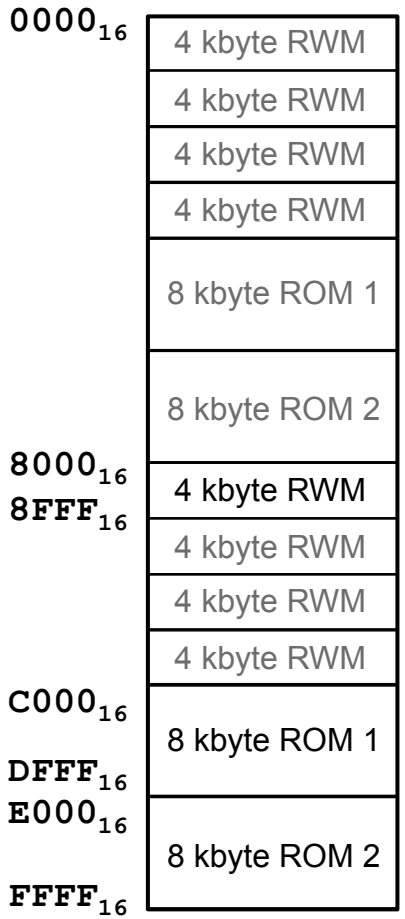


Adressavkodning

Ofullständig adressavkodning

För att förenkla adressavkodningen kan man ibland låta bli att använda alla adressbitarna. Detta får till följd att avkodningen inte blir entydig, d v s samma ”chip select”-signal kan bli aktiv i fler än ett adressintervall. Man kallar denna metod för ofullständig adressavkodning.

Om vi i vårt exempel skulle använda enklast möjliga avkodning skulle minnesmodulerna fortfarande ha sina ursprungliga unika adressområden, men det skulle också bli möjligt att adressera dem på andra ställen i adressrummet.



Adressavkodning

Ofullständig adressavkodning

Vi ser i vårt exempel att endast adressbit A_{14} är helt nödvändig för att skilja på adressområdena för RWM- och ROM-modulerna.

För att välja mellan de två ROM-modulerna behöver man sedan lägga till adressbit A_{13}.

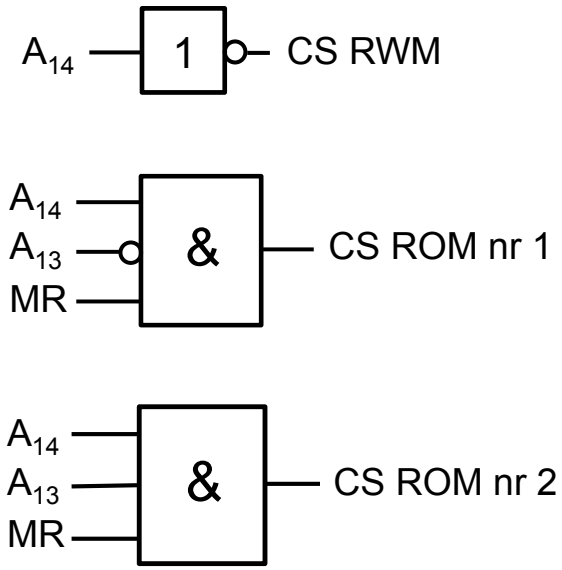
Adressavkodningen kan alltså göras med ett färre antal grindar, och med grindar med ett färre antal ingångar.

Modul	Adress	Adressignal nr															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RWM	0000H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	0FFFH	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
RWM	1000H	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	1FFFH	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
RWM	2000H	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	2FFFH	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
RWM	3000H	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	3FFFH	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM nr 1	4000H	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	5FFFH	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM nr 2	6000H	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	7FFFH	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RWM	8000H	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	8FFFH	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
RWM	9000H	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	9FFFH	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
RWM	A000H	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	AFFFH	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
RWM	B000H	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	BFFFH	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM nr 1	C000H	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	DFFFH	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM nr 2	E000H	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	FFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Adressavkodning

Ofullständig adressavkodning

Adressavkodningen kan alltså göras med ett färre antal grindar, och med grindar med ett färre antal ingångar.



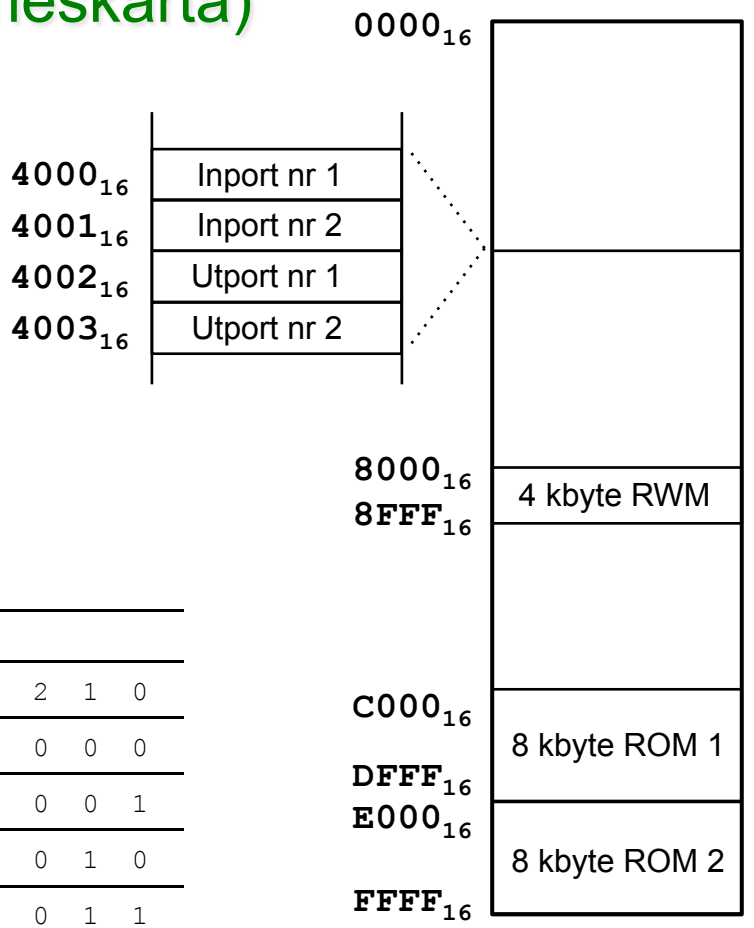
Modul	Adress	Adressignal nr															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RWM	0000H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	0FFFH	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
RWM	1000H	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	1FFFH	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
RWM	2000H	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	2FFFH	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
RWM	3000H	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	3FFFH	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM nr 1	4000H	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	5FFFH	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM nr 2	6000H	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	7FFFH	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RWM	8000H	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	8FFFH	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
RWM	9000H	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	9FFFH	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
RWM	A000H	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	AFFFH	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
RWM	B000H	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	BFFFH	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM nr 1	C000H	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	DFFFH	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM nr 2	E000H	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	:									:							
	FFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Adressavkodning

Placering av periferikretsar (minneskarta)

Antag att man nu skulle vilja utöka vårt exempel med två inportar och två utportar, och placera dem i adressområdet 4000_{16} – 4003_{16}

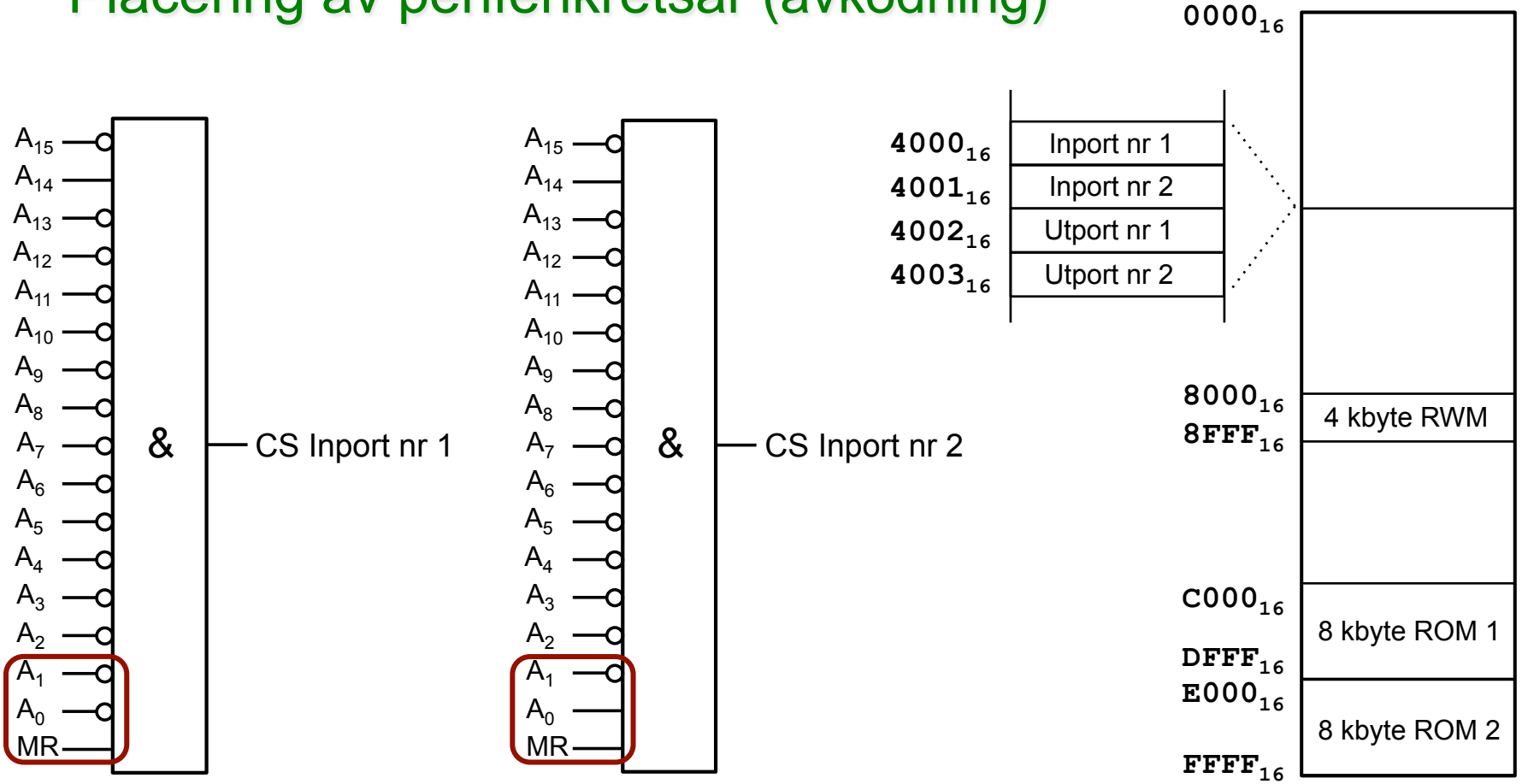
Med fullständig adressavkodning kommer alla 16 adressbitar att krävas vid avkodningen.



Modul	*Adress*	*Adressignal nr*															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Inport 1	4000H	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Inport 2	4001H	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Utport 1	4002H	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Utport 2	4003H	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1

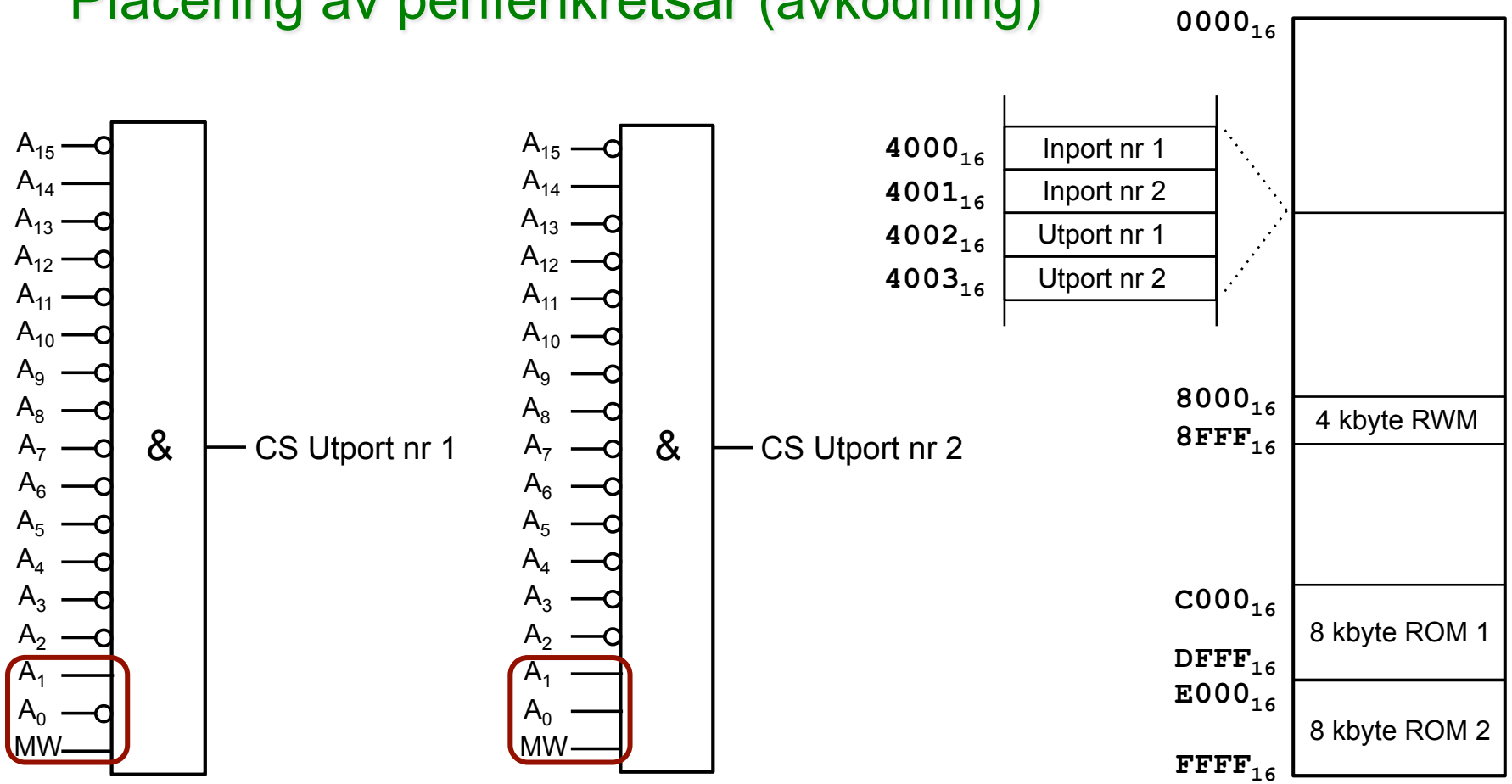
Adressavkodning

Placering av periferikretsar (avkodning)



Adressavkodning

Placering av periferikretsar (avkodning)

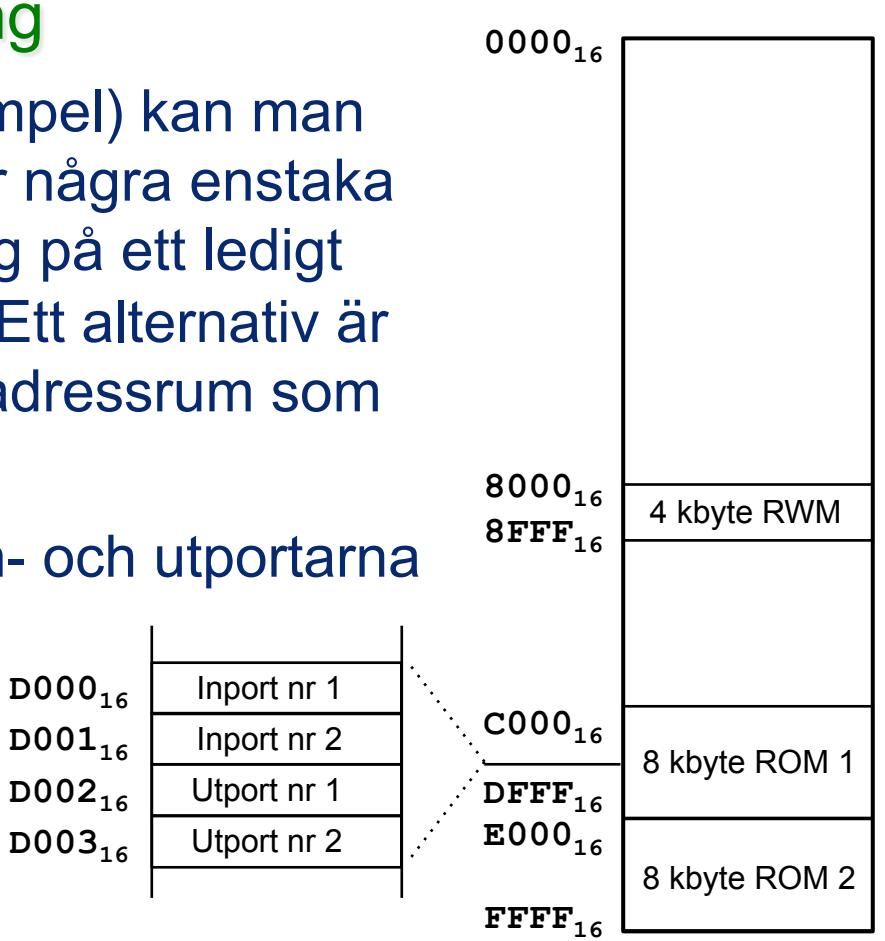


Adressavkodning

Överlagrad minnesavbildning

I en del system (som i vårt exempel) kan man få en situation där man behöver några enstaka portar, men inte vill lägga beslag på ett ledigt adressrum enbart för portarna. Ett alternativ är då att lägga portarna i samma adressrum som en redan existerande modul.

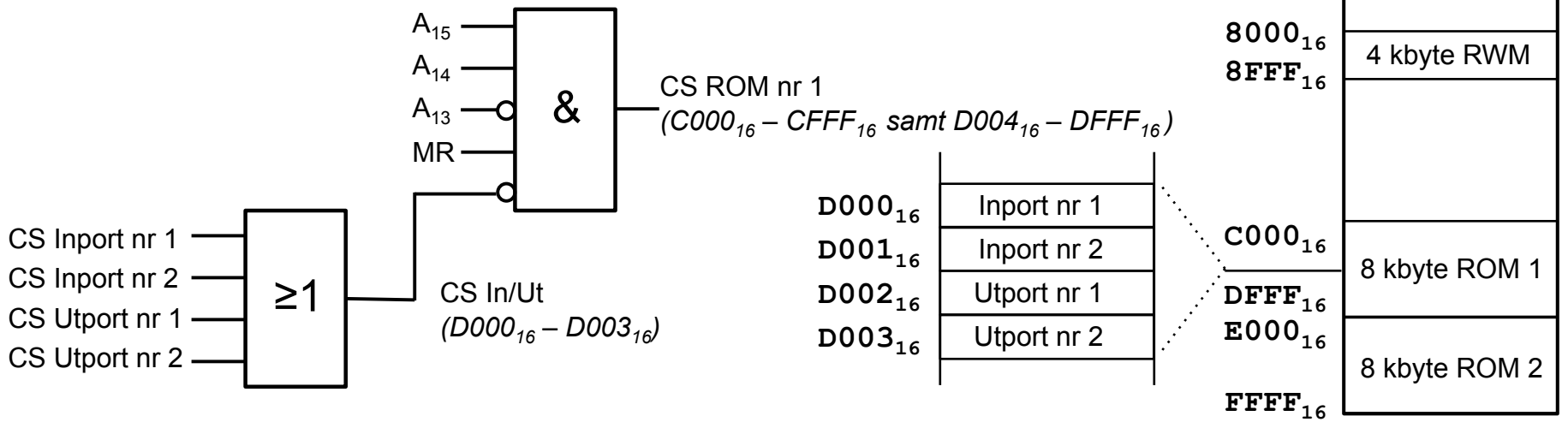
Antag att man väljer att lägga in- och utportarna i området $D000_{16}$ – $D003_{16}$, d v s i samma del av adressrummet som en av ROM-modulerna.



Adressavkodning

Överlagrad minnesavbildning

Detta löses genom överlagrad minnesavbildning, där man först skapar en fullständig avkodning för portarna och sedan använder inversen av den erhållna avkodningen som kompletterande villkor för ”chip select”-signalen för minnesmodulen.



Adressavkodning

Generell avkodningslogik

I stället för att bygga upp adressavkodningen med diskreta grindar använder man i praktiken ofta färdiga kretsar för att bilda ”chip select”-signalerna, t ex binäravkodare, fördelare (”demultiplexer”), eller PLA-/PAL-kretsar.

PLA- och PAL-kretsar är exempel på s k programmerbar logik, PLD (Programmable Logic Device). Adressavkodning som utförs med denna typ av kretsar blir därmed den mest flexibla.

Att fundera på:

Hur skulle adressavkodningen för vårt exempel göras om vi använde binäravkodare eller fördelare?