2014-04-16

Adressavkodning

Övningsuppgifter

2014

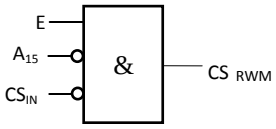
Lösningar

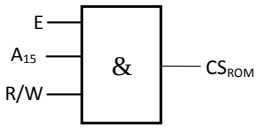
(Med reservation för diverse fel!)

1.

RWM: 32k = $= 2^5 \cdot 2^{10} = 2^{15}$

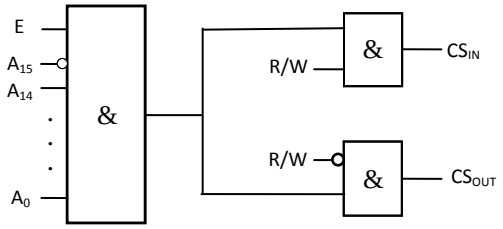
	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: 0000H =		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: 7FFFH =		0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS	15 st														





ROM: 32k = $= 2^5 \cdot 2^{10} = 2^{15}$

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: 8000H =		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: FFFFH =		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS	15 st														



I/O:

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adress: 7FFFH =		0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
										CS							

2.

ROM: $2^{14} = 16k$

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: C000H =		1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: FFFFH =		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS	14 st														

RWM: $2^{13} = 8k$

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: 4000H =		0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: 5FFFH =		0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS	13 st														

UTPORT:

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr:		0	0	0	0	0	0	0	1	-	-	-	-	-	-	-	-
Start: 0100H =		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Slut: 01FFH =		0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
						CS				8st							

Samma utport ”finns” på 2^8 olika adresser.

INPORT:

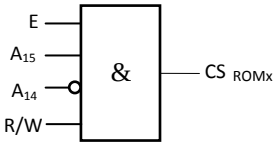
	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr:		0	0	0	0	0	0	1	0	-	-	-	-	-	-	-	-
Start: 0200H =		0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Slut: 02FFH =		0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1
						CS				8st							

Samma inport ”finns” på 2^8 olika adresser.

Inkoppling av ytterligare en 16k ROM-modul:
Det krävs att det finns en ledig adresslucka med konstanta $A_{15}A_{14}$ för att rymma 16k.
$A_{15}A_{14} = 10$ är ledig.

ROMx: $2^{14} = 16k$

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: 8000H =		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: BFFFH =		1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS	14 st														



RWM används för variabler, dvs ändringsbara data och för program under utvecklingsskedet. ROM används för program som inte ändras och för konstanter. RWM är flyktigt, dvs det tappar informationen vid spänningsbortfall. ROM är icke flyktigt, dvs det behåller informationen vid spänningsbortfall.

3.

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Modul 1: 2^{13}= 8k — Start: A000H =		1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: BFFFH =		1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS			13 st												

Detta bör vara en RWM-modul med kapaciteten 8kbyte. (RWM eftersom R/W-signalen saknas vid CS-avkodningen.)

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Modul 2: 2^{14}= 16k — Start: C000H =		1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: FFFFH =		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS		14 st													

Detta bör vara en ROM-modul med kapaciteten 16kbyte. (ROM eftersom R/W-signalen finns med vid CS-avkodningen.)

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Modul 3: Adr: 600EH =		0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0
							CS										

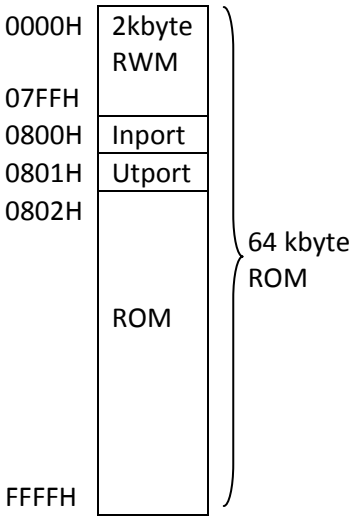
Detta är en inport. (Endast en adress och modulen aktiveras vid läsning i ”minnet”.)

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Modul 4: Adr: 600FH =		0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1
							CS										

Detta är en utport. (Endast en adress och modulen aktiveras vid skrivning i ”minnet”.)

E-signalen finns med i CS-avkodningen eftersom adressbitarna har giltiga värden endast när E = 1.

4.



RWM: $2k = 2 \cdot 2^{10} = 2^{11}$

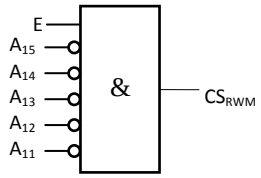
	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: 0000H =		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: 07FFH =		0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
						CS											
							11 st										

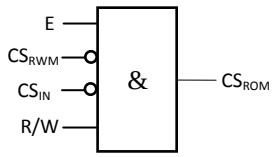
Inport:

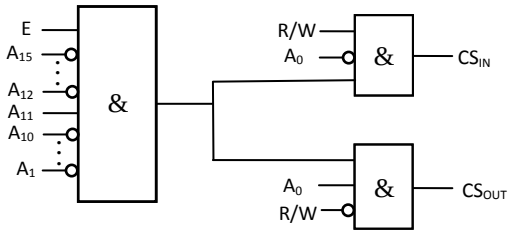
	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: 0800H =		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
									CS								

Utport:

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: 0801H =		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
									CS								







5.

$x_0 = \overline{A}_{13}\overline{A}_{12}$ $x_4 = \overline{A}_{15}\overline{A}_{14}$ $CS_{RWM} = E\ x_5 = E\ \overline{A}_{15}A_{14}$

$x_1 = \overline{A}_{13}A_{12}$ $x_5 = \overline{A}_{15}A_{14}$ $CS_{ROM} = E\ x_7(x_3 + x_2)R/W = E\ A_{15}A_{14}\ (A_{13}A_{12} + A_{13}\overline{A}_{12})\ R/W = E\ A_{15}A_{14}A_{13}\ R/W$

$x_2 = A_{13}\overline{A}_{12}$ $x_6 = A_{15}\overline{A}_{14}$ $CS_{IN} = E\ x_4\ x_0\ R/W = E\ \overline{A}_{15}\overline{A}_{14}\overline{A}_{13}\overline{A}_{12}R/W$

$x_3 = A_{13}A_{12}$ $x_7 = A_{15}A_{14}$ $CS_{UT} = E\ x_4\ x_0\ \overline{R/W} = E\ \overline{A}_{15}\overline{A}_{14}\overline{A}_{13}\overline{A}_{12}\overline{R/W}$

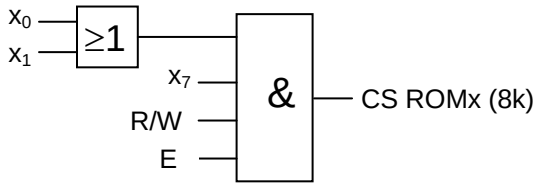
RWM:	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2^{14}= 16k Start: 4000H =		0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: 7FFFH =		0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS		14 st													

ROM:	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2^{13}= 8k Start: E000H =		1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: FFFFH =		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS			13 st												

I/U-PORT:	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr:		0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-
Start: 0000H =		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: 0FFFH =		0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
		CS				12 st											

Samma portar ”finns” på 2^{12} olika adresser.

För att bilda en sammanhängande ROM-area måste den nya 8k-ROM-modulen (ROMx) ha slutadressen DFFFH. Adressbitarna $A_{15}A_{14}A_{13}$ måste då ha värdena 110. Detta ger $x_7(x_1+x_0) = A_{15}A_{14}\ (\overline{A}_{13}A_{12} + \overline{A}_{13}\overline{A}_{12}) = A_{15}A_{14}\overline{A}_{13}$



6.

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROM: 2^{14}= 16k	Start: C000H =	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Slut: FFFFH =	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS		14 st													

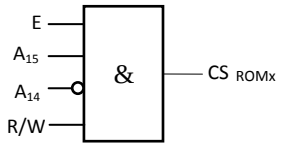
	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RWM: 2^{14}= 16k	Start: 4000H =	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Slut: 5FFFH =	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS		14 st													

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I/U-PORT:																	
	Adr:	0	0	0	0	0	1	0	1	1	-	-	-	-	-	-	-
	Start: 0580H =	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0
	Slut: 05FFH =	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1
						CS					7st						

Samma I/U-port ”finns” på 2^7 olika adresser.

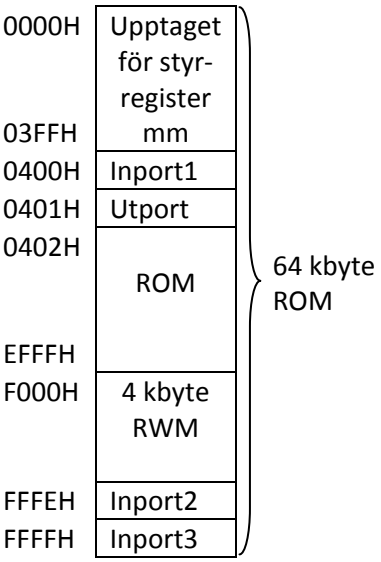
Inkoppling av ytterligare en 16k ROM-modul:
Det krävs att det finns en ledig adresslucka med konstanta $A_{15}A_{14}$ för att rymma 16k.
$A_{15}A_{14}$ = 10 är ledig.

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROMx: 2^{14}= 16k	Start: 8000H =	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Slut: BFFFH =	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS		14 st													



RWM används för variabler, dvs ändringsbara data och för program under utvecklingsskedet. ROM används för program som inte ändras och för konstanter. RWM är flyktigt, dvs det tappar informationen vid spännings-bortfall. ROM är icke flyktigt, dvs det behåller informationen vid spänningsbortfall.

7.



Passiv area
$1k = 2^{10}$

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: 0000H =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: 03FFH =	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	CS						10 st									

RWM
$4k = 2^{12}$

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: F000H =	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Slut: FFFFH =	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	CS				12 st											

Inport1:

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: 0400H =	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
						CS										

Utport:

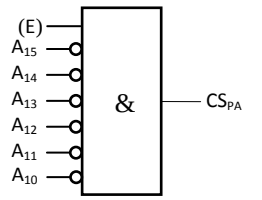
A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: 0401H =	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
						CS										

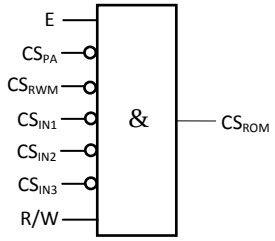
Inport2:

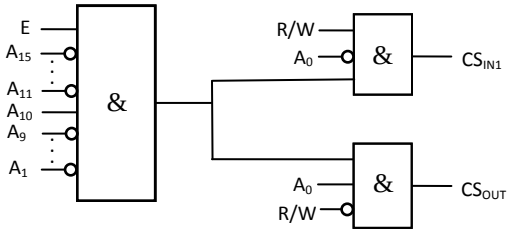
A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: FFFEH =	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
						CS										

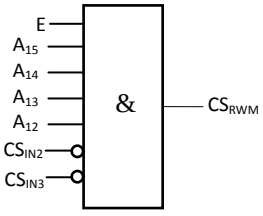
Inport3:

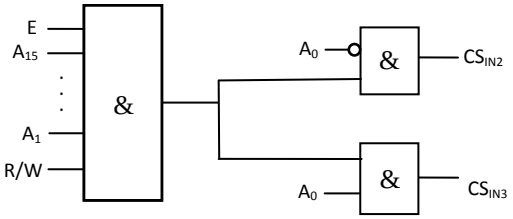
A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: FFFFH =	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
						CS										









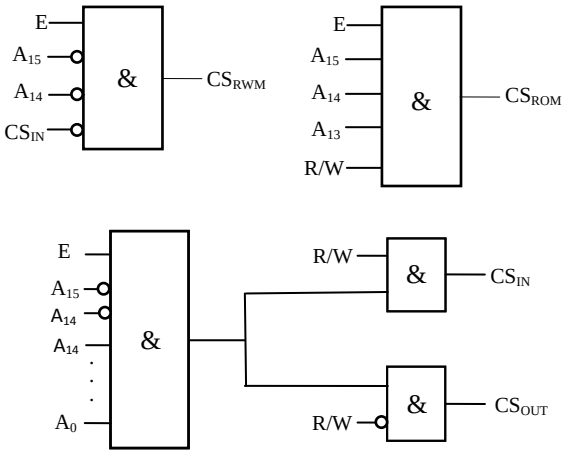


8.

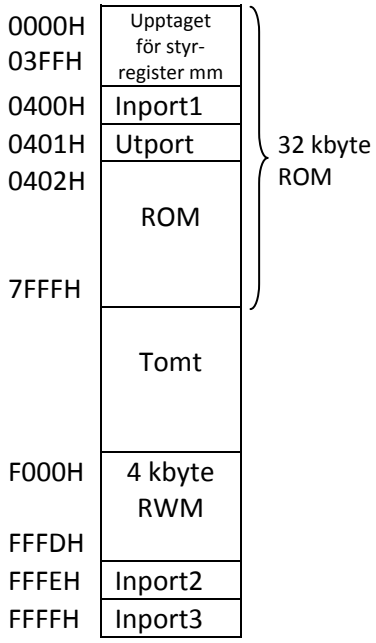
		A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RWM:																		
16k = 2^{14}	Start: 0000H =		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Slut: 3FFFH =		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
			CS															

		A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROM:																		
8k = 2^{13}	Start: E000H =		1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	Slut: FFFFH =		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
			CS															

		A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I/O:	Adress: 3FFFH =		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
										CS								



9.



Passiv area, 1k = 2^{10}

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: 0000H =		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: 03FFH =		0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
		CS						10 st									

ROM, 32k = 2^{15}

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: 0000H =		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: 7FFFH =		0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS	15 st														

RWM, 4k = 2^{12}

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: F000H =		1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Slut: FFFFH =		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS				12 st											

Inport1:

	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: 0400H =		0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
		CS															

Utport:

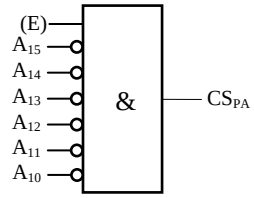
	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: 0401H =		0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
		CS															

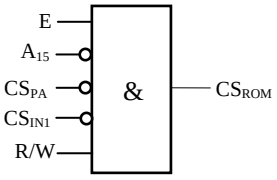
Inport2:

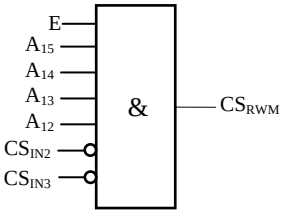
	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: FFFEH =		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
		CS															

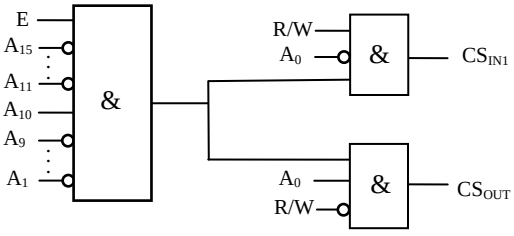
Inport3:

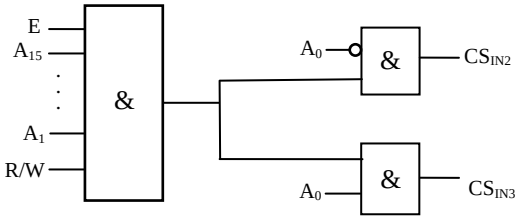
	A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: FFFFH =		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		CS															



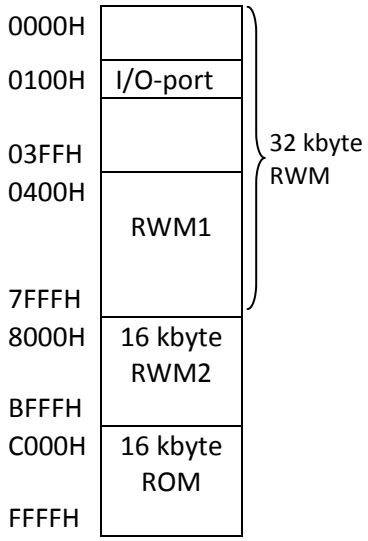








10.



Passiv area
$1k = 2^{10}$

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: 0000H =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: 03FFH =	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	CS						10 st									

I/O-port:

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Adr: 0100H =	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	CS															

RWM1
$32k = 2^{15}$

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: 0000H =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: 7FFFH =	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	CS	15 st														

RWM2
$16k = 2^{14}$

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: 8000H =	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: BFFFH =	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	CS		14 st													

ROM
$16k = 2^{14}$

A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start: C000H =	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slut: FFFFH =	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	CS		14 st