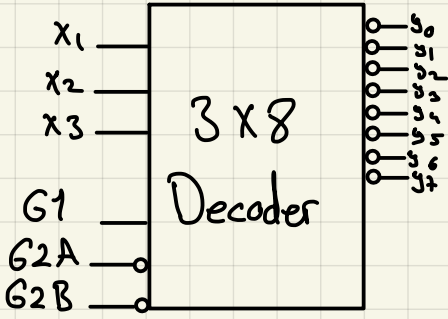


Mikroişlemcili Sistemler ve Laboratuvarı

Hazırlayan : M. Kemal Güvenç

Öğretmen : Murat İskefeyeli

Bellek Organizasyonu



n adet bit ile 2^n adet bölge adreslenebilir.

10 adet bit ile 2^{10} (1024) adet bölge adreslenebilir.

0000 0000 0000 0000 \rightarrow (0000)_h

0000 0011 1111 1111 \rightarrow (03FF)_h

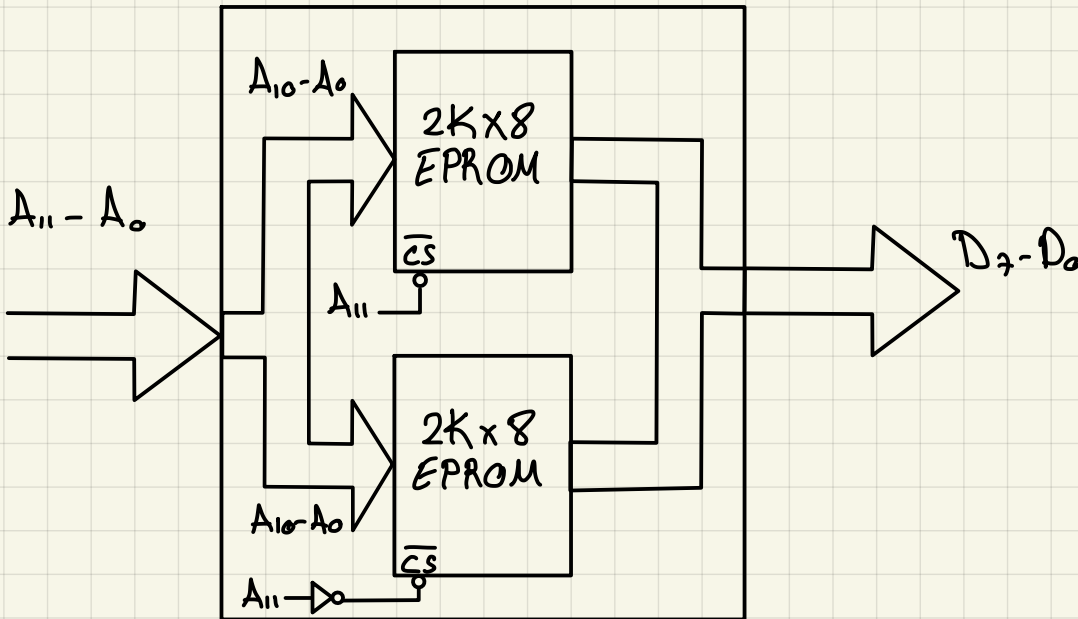
\Rightarrow G1=1 ve G2A=G2B=0 olunca çalışır.

\Rightarrow 1K x 8, 2^{10} adet 8 bitlik bölgeye sahip belleği ifade eder.

ör/ 2 adet 2K x 8 EPROM kullanarak 4K x 8 EPROM elde ediniz.

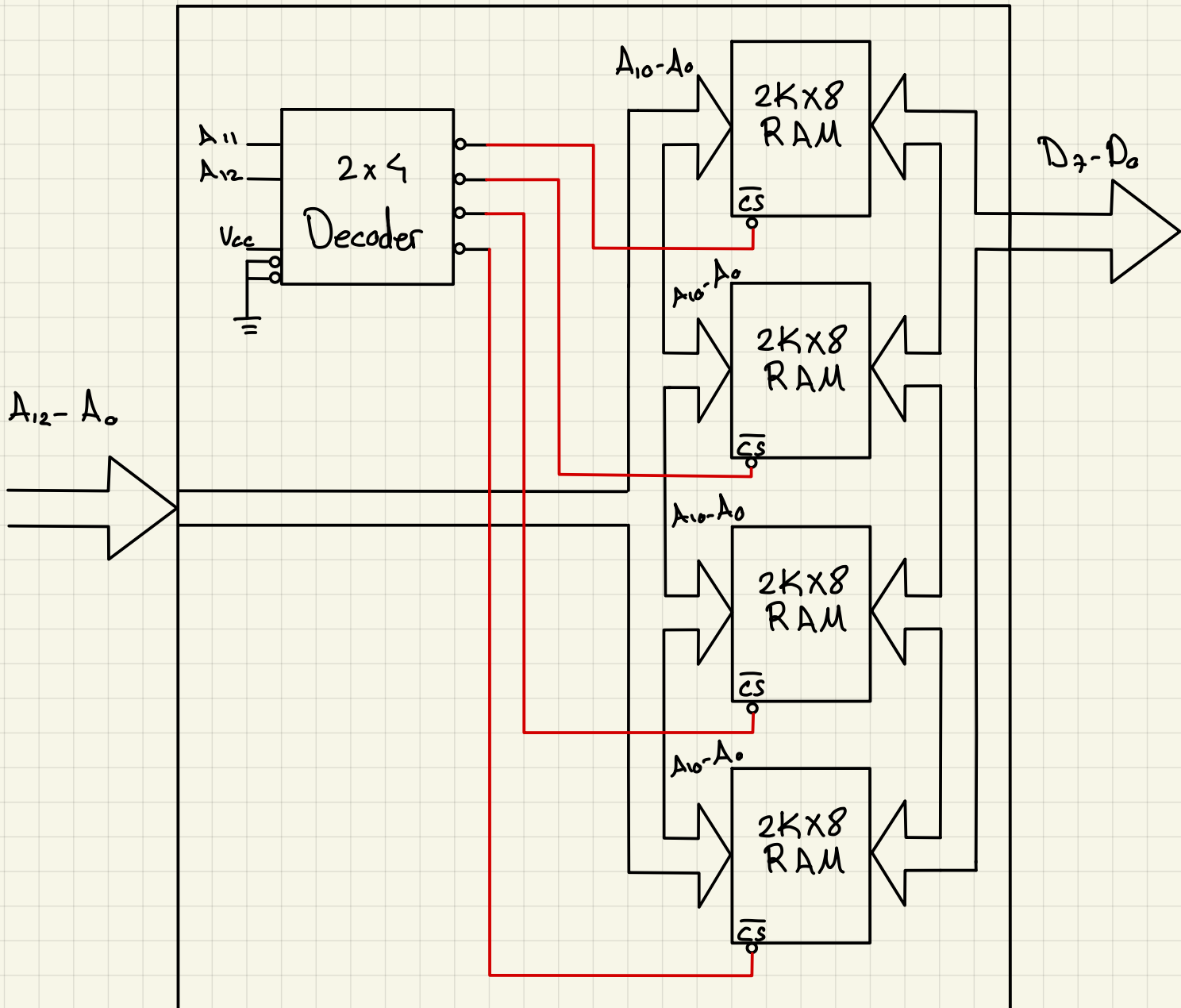
\Rightarrow

0000	0000	0000	0000	\rightarrow (0000) _h	} 2K x 8
0000	0111	1111	1111	\rightarrow (07FF) _h	
0000	1000	0000	0000	\rightarrow (0800) _h	} 2K x 8
0000	1111	1111	1111	\rightarrow (0FFF) _h	

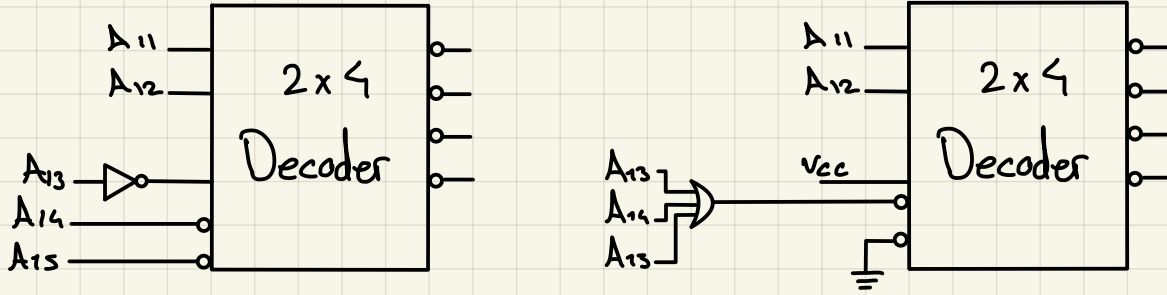


4 adet $2K \times 8$ RAM kullanarak $8K \times 8$ RAM elde ediniz.

A_{12}	A_{11}	A_{10}		
0000	0000	0000	0000	$\rightarrow 0x0000$
0000	0111	1111	1111	$\rightarrow 0x07FF$
0000	1000	0000	0000	$\rightarrow 0x0800$
0000	1111	1111	1111	$\rightarrow 0x0FFF$
0001	0000	0000	0000	$\rightarrow 0x1000$
0001	0111	1111	1111	$\rightarrow 0x17FF$
0001	1000	0000	0000	$\rightarrow 0x1800$
0001	1111	1111	1111	$\rightarrow 0x1FFF$

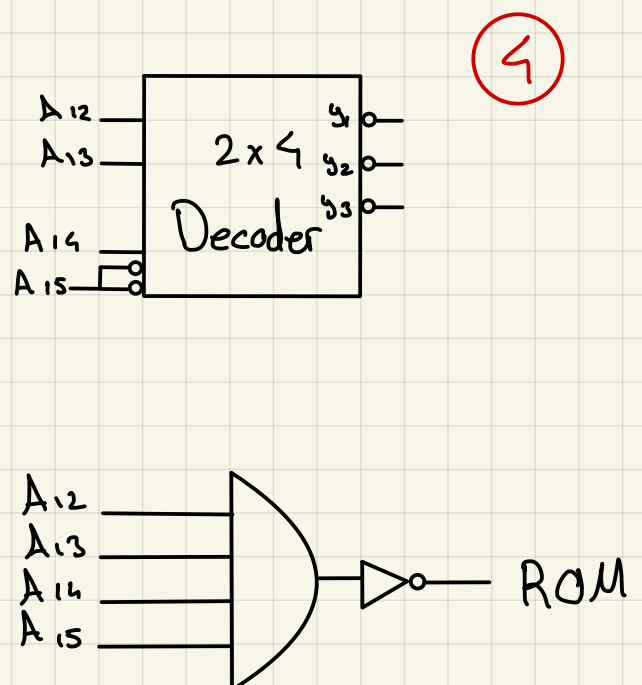
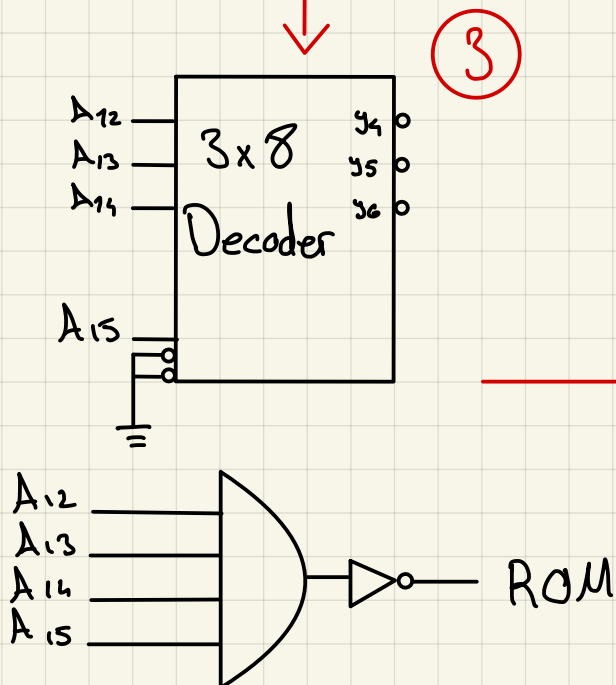
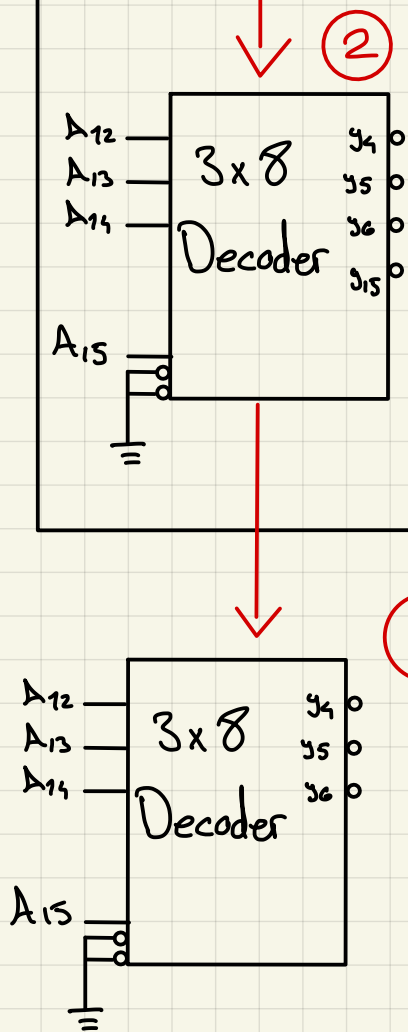
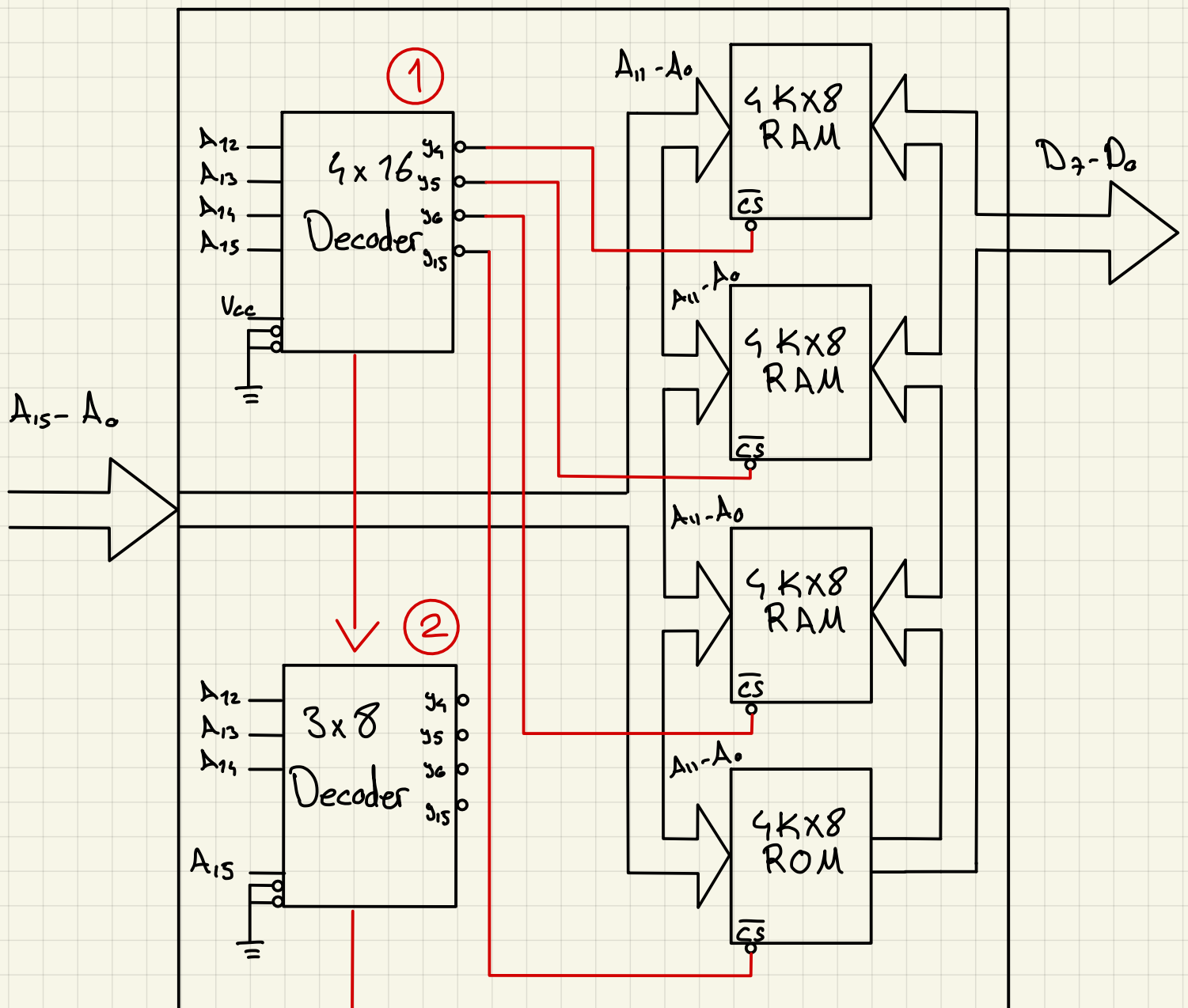


Yapılan çözümde **yansımada** problemi bulunmaktadır. A_{13} ve A_{14} çıkışlarını bağlanmadığımız için $0x0FFF, 0x2FFF, 0x4FFF, 0x6FFF$ vs. olan adreslerin hepsi aynı yeri gösterecektir. Bunu çözmek için birden fazla yol olabilir. Örnek olarak 2 çözüm aşağıdaki gibidir.



~~ör~~ $4K \times 8$ RAM ve $4K \times 8$ ROM kullanarak $12K \times 8$ RAM ve $4K \times 8$ ROM elde edilecektir. Ayrıca RAM bellek adresi $(4000)_n$ 'den ROM bellek adresi ise $(F000)_n$ adresinden başlayacaktır.

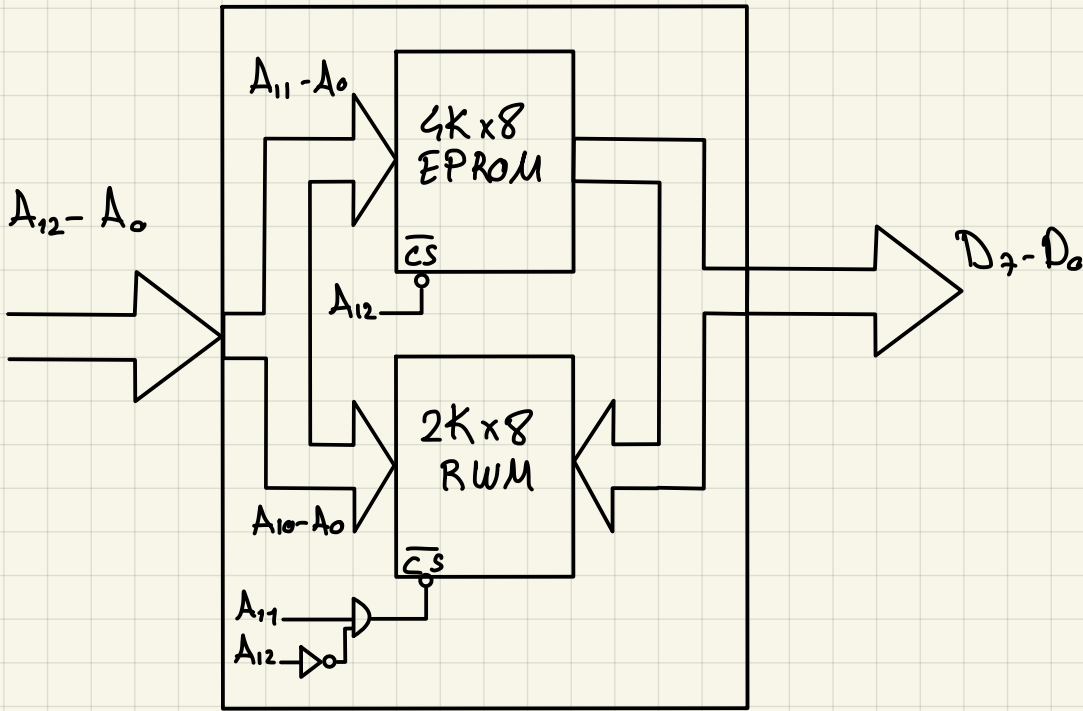
0000	0000	0000	$0000 \rightarrow 0x0000$	} Boş
0011	1111	1111	$1111 \rightarrow 0x3FFF$	
0100	0000	0000	$0000 \rightarrow 0x4000$	} RAM 1
0100	1111	1111	$1111 \rightarrow 0x4FFF$	
0101	0000	0000	$0000 \rightarrow 0x5000$	} RAM 2
0101	1111	1111	$1111 \rightarrow 0x5FFF$	
0110	0000	0000	$0000 \rightarrow 0x6000$	} RAM 3
0110	1111	1111	$1111 \rightarrow 0x6FFF$	
0111	0000	0000	$0000 \rightarrow 0x7000$	} Boş
1110	1111	1111	$1111 \rightarrow 0xEFFF$	
1111	0000	0000	$0000 \rightarrow 0xF000$	} ROM
1111	1111	1111	$1111 \rightarrow 0xFFFF$	



ör 4Kx8 EPROM ve 2Kx8 RWM kullanarak 6Kx8'lik bellek elde ediniz.

①

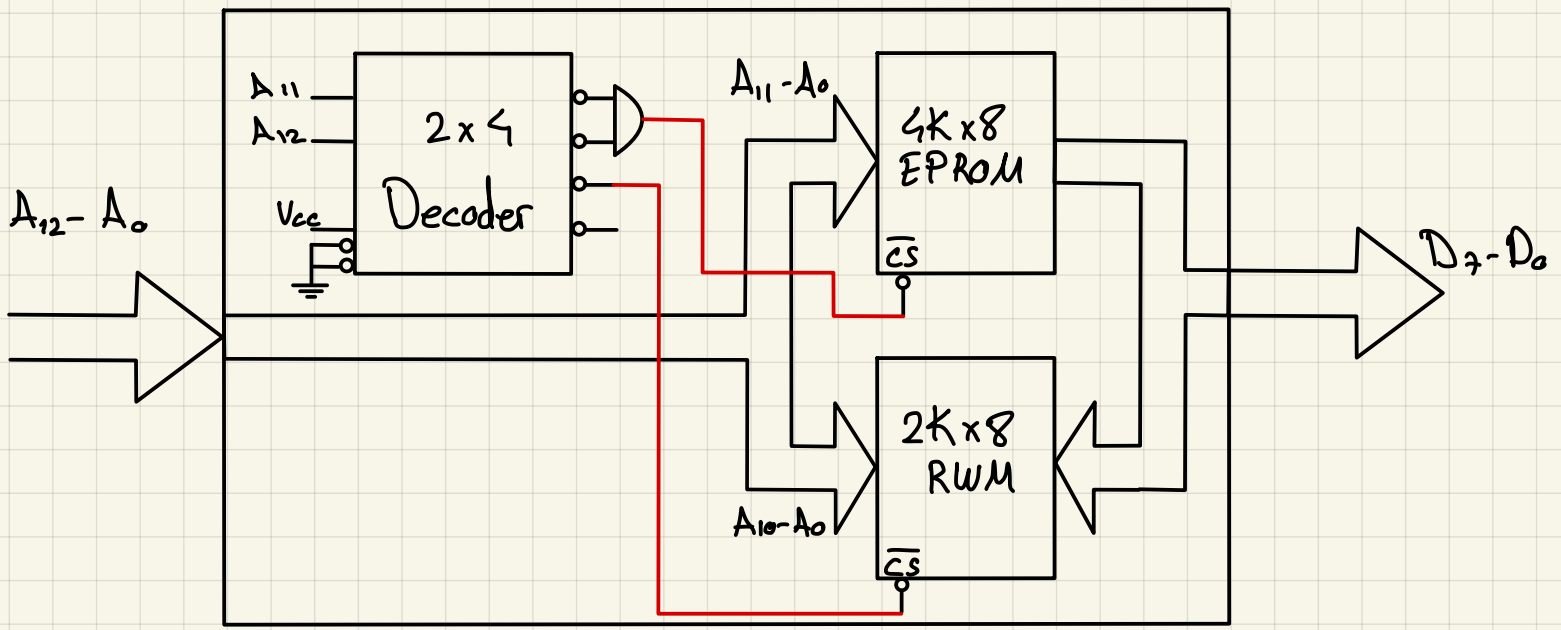
0000	0000	0000	0000	→ 0x0000	} 4Kx8 EPROM
0000	1111	1111	1111	→ 0x0FFF	
0001	0000	0000	0000	→ 0x1000	
0001	0111	1111	1111	→ 0x17FF	} 2Kx8 RWM



Not: Her ne kadar bu soruda bu şekilde çözebilmiş olsak da farklı boyutlara sahip bellekleri organize etmek için genel bir çözümümüz vardır. Bu çözüme göre bellek boyutlarının EBOB'u bulunur. Sonra diğer bellekler bu boyuta göre parçalanır.

②

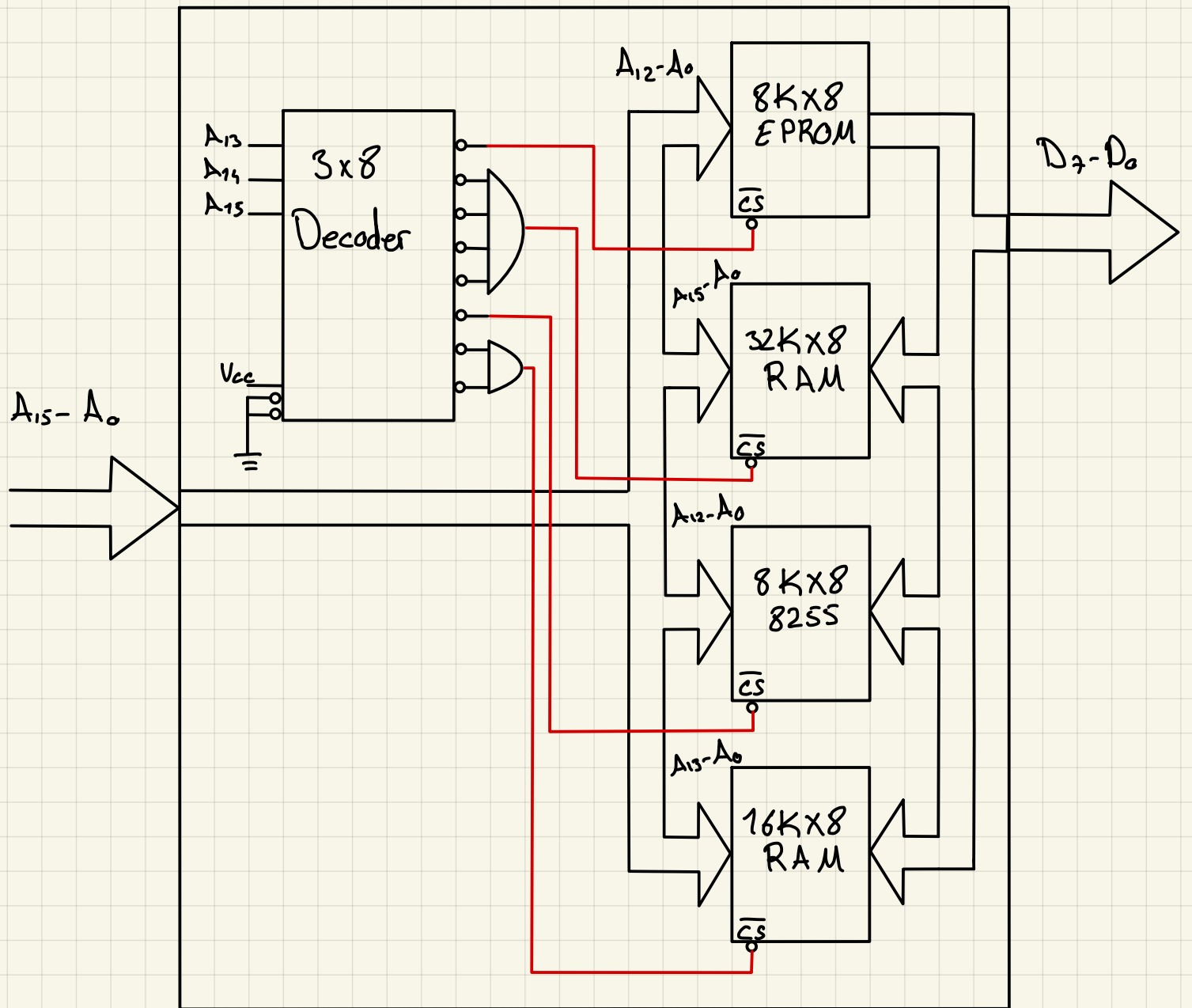
0000	0000	0000	0000	→ 0x0000	} 4Kx8 EPROM
0000	0111	1111	1111	→ 0x07FF	
0000	1000	0000	0000	→ 0x0800	
0000	1111	1111	1111	→ 0x0FFF	} 2Kx8 RWM
0001	0000	0000	0000	→ 0x1000	
0001	0111	1111	1111	→ 0x17FF	



~~ör~~ 8Kx8 EPROM, 32Kx8 RAM, 8255 (8Kx8 RAM gibi düşünülecek) ve 16Kx8 RAM kullanılacak şekilde bir bellek organizasyonu yapınız.

⇒

0000	0000	0000	0000	→ 0x0000	} EPROM (8Kx8)
0001	1111	1111	1111	→ 0x1FFF	
0010	0000	0000	0000	→ 0x2000	
0011	1111	1111	1111	→ 0x3FFF	} RAM (32Kx8)
0100	0000	0000	0000	→ 0x4000	
0101	1111	1111	1111	→ 0x5FFF	
0110	0000	0000	0000	→ 0x6000	} RAM (16Kx8)
0111	1111	1111	1111	→ 0x7FFF	
1000	0000	0000	0000	→ 0x8000	
1001	1111	1111	1111	→ 0x9FFF	} 8255 (8Kx8)
1010	0000	0000	0000	→ 0xA000	
1011	1111	1111	1111	→ 0xBFFF	
1100	0000	0000	0000	→ 0xC000	} RAM (16Kx8)
1101	1111	1111	1111	→ 0xDFFF	
1110	0000	0000	0000	→ 0xE000	
1111	1111	1111	1111	→ 0xFFFF	



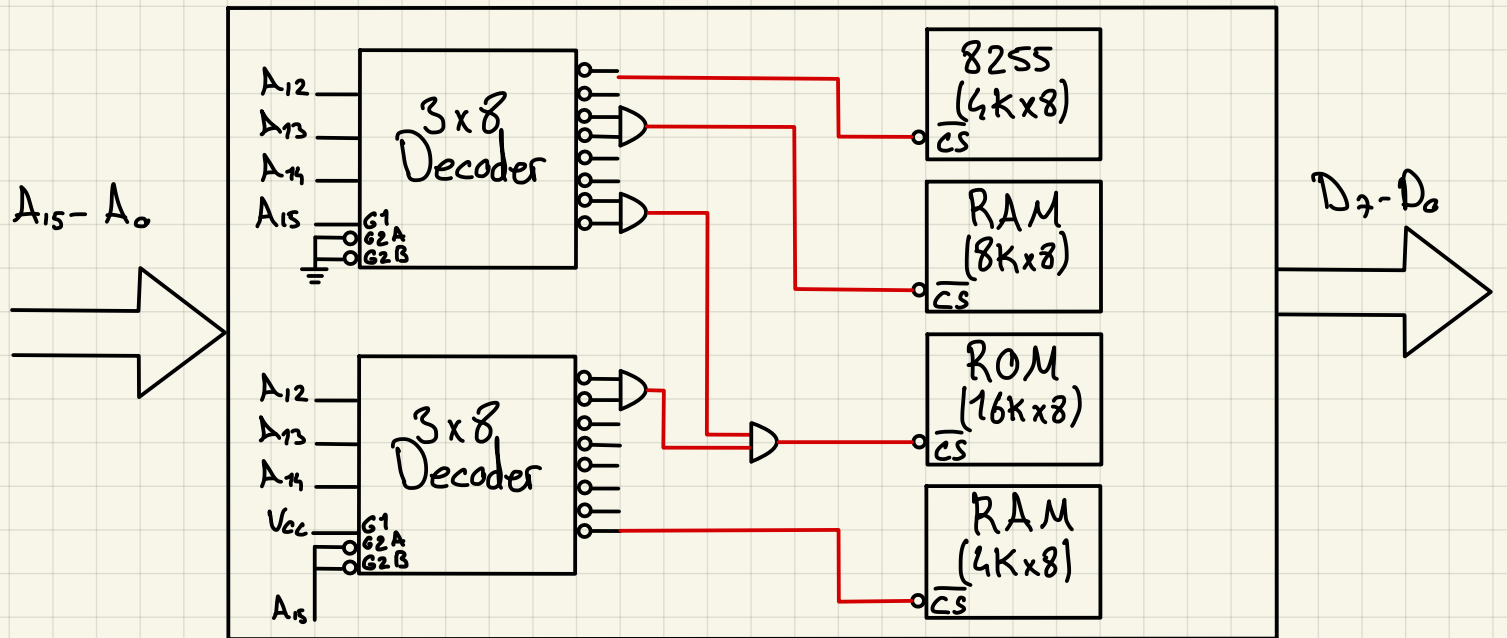
8085 mikroişlemcisine, bellek ve I/O birimlerine sahip bir mikroişlemcili sistemde;

- Bellek birimleri: 8255 arabirimi (4Kx8 gibi düşünün) 0000h adresinden, 8Kx8'lik RAM'in 2000h adresinden, 16Kx8'lik ROM'un 6000h adresinden, 4Kx8'lik RAM'in C000h adresinden itibaren,
- I/O birimleri: 48X8'lik I/O-1 birimi 20h adresinden, 48X8'lik I/O-2 birimi 70h adresinden, 88X8'lik I/O-3 birimi A0h adresinden, 8255 arabirimi F0h adresinden itibaren yerleştirilmiştir.

Bu verilere göre tüm sistem için kontrol devresini tasarlayarak işlemci, adres tutucu, bellek ve I/O'ların bağlantılarını çiziniz.

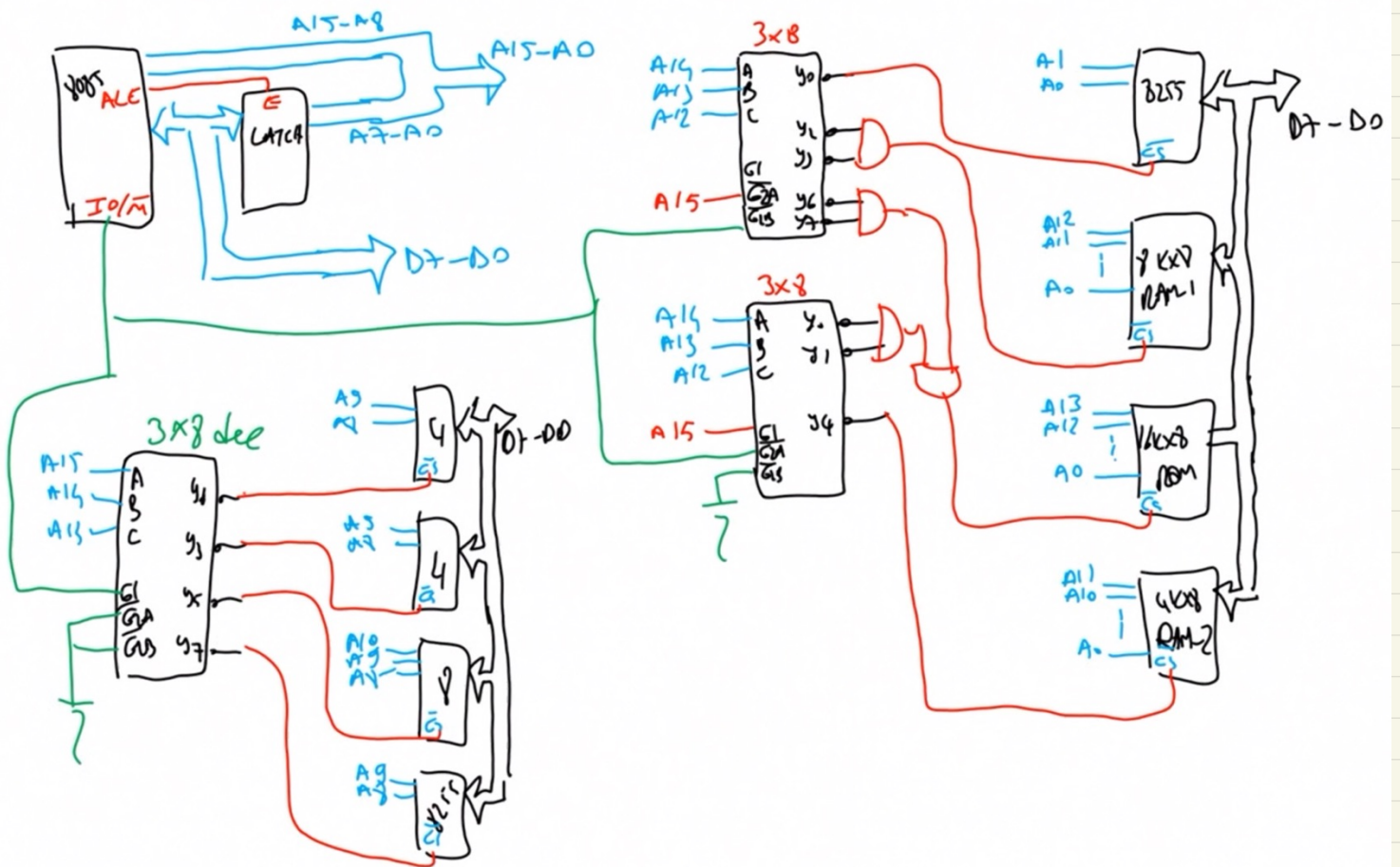
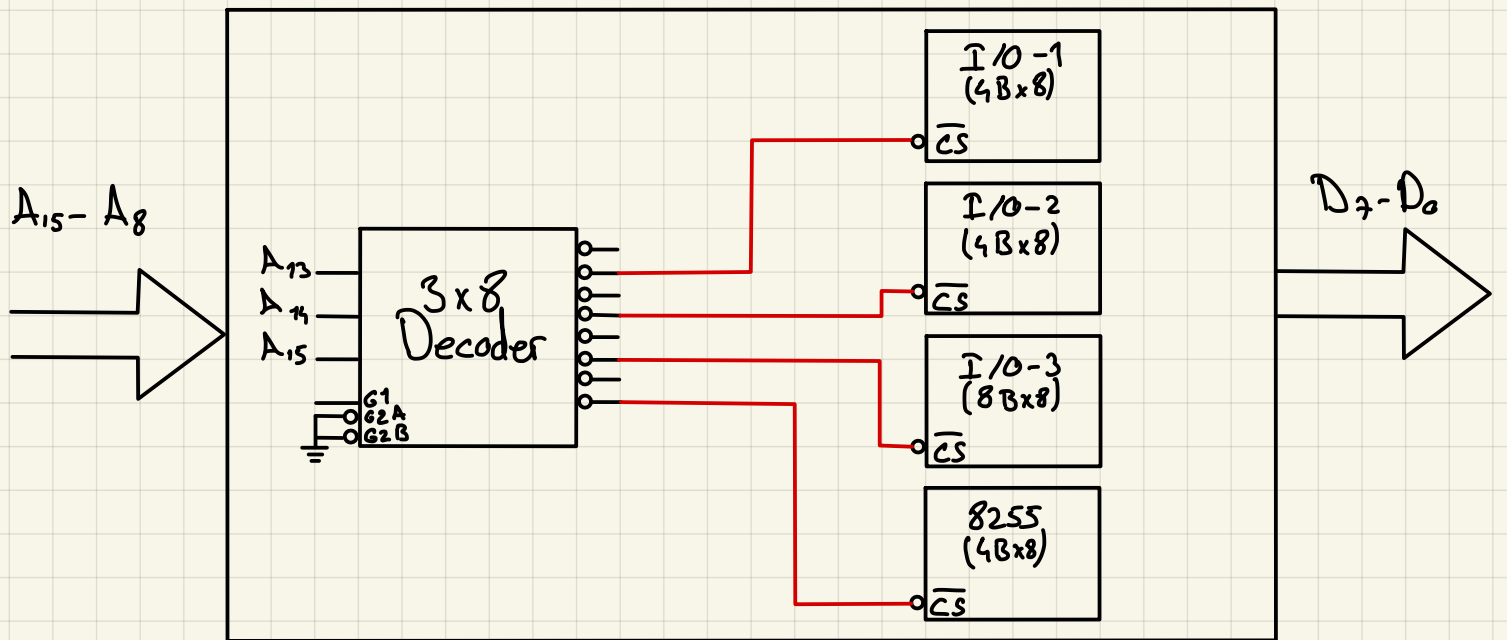
(Elimizde kod çözümü olarak 3x8'likler ve mantık kapıları (AND ve OR) mevcuttur. Bellek birimleri için yansız, I/O birimleri için yansız yapılabilmektedir.)

⇒ 0000	0000	0000	0000 → 0x0000	8255 (4Kx8)	0x0000	8255 (4Kx8)
0000	1111	1111	1111 → 0x0FFF		0x0FFF	
0001	0000	0000	0000 → 0x1000	Boş	0x1000	Boş
0001	1111	1111	1111 → 0x1FFF	RAM (8Kx8)	0x1FFF	
0010	0000	0000	0000 → 0x2000		0x2000	
				RAM (8Kx8)		
0010	1111	1111	1111 → 0x2FFF			
0011	0000	0000	0000 → 0x3000	Boş	0x3FFF	
					0x4000	
0011	1111	1111	1111 → 0x3FFF	Boş		
0100	0000	0000	0000 → 0x4000			
				ROM (16Kx8)	0x5FFF	
0101	1111	1111	1111 → 0x5FFF		0x6000	
0110	0000	0000	0000 → 0x6000	ROM (16Kx8)		
0110	1111	1111	1111 → 0x6FFF	ROM (16Kx8)		
0111	0000	0000	0000 → 0x7000			
				ROM (16Kx8)		
0111	1111	1111	1111 → 0x7FFF			
1000	0000	0000	0000 → 0x8000	Boş	0x9FFF	
					0xA000	
1000	1111	1111	1111 → 0x8FFF	Boş		
1001	0000	0000	0000 → 0x9000			
				RAM (4Kx8)	0xBFFF	
1001	1111	1111	1111 → 0x9FFF		0xC000	
1010	0000	0000	0000 → 0xA000	Boş	0xCFFF	
					0xD000	
1011	1111	1111	1111 → 0xBFFF	RAM (4Kx8)		
1100	0000	0000	0000 → 0xC000			
				Boş		
1100	1111	1111	1111 → 0xCFFF			
1101	0000	0000	0000 → 0xD000	Boş		
1111	1111	1111	1111 → 0xFFFF		0xFFFF	



0000	0000	→ 0x00	BoF
...	
0001	1111	→ 0x1F	BoF
0010	0000	→ 0x20	
...	I/O-1 (1)
0010	0011	→ 0x23	
0010	0100	→ 0x24	BoF
...	
0110	1111	→ 0x6F	BoF
0111	0000	→ 0x70	
...	I/O-2 (3)
0111	0011	→ 0x73	
0111	0100	→ 0x74	BoF
...	
1001	1111	→ 0x9F	BoF
1010	0000	→ 0xA0	
...	I/O-3 (5)
1010	0011	→ 0xA3	
1010	0100	→ 0xA4	BoF
...	
1010	0111	→ 0xA7	BoF
1010	1000	→ 0xA8	
...	BoF
1110	1111	→ 0xEF	
1111	0000	→ 0xF0	8255 (7)
...	
1111	0011	→ 0xF3	BoF
1111	0100	→ 0xF4	
...	BoF
1111	1111	→ 0xFF	

0x00	BoF
0x1F 0x20	I/O-1
0x23 0x24	BoF
0x6F 0x70	I/O-2
0x73 0x74	BoF
0x9F 0xA0	I/O-3
0xA3 0xA8	BoF
0xEF 0xF0	8255
0xF3 0xF4	BoF
0xFF	BoF



(FF)H

— Bellek Boyutları —

Dahili Kod Belleği: 4 KiB

Harici Kod Belleği: 64 KiB'a kadar genişletilebilir.

Dahili Veri Belleği: 256 Bayt (128+128)

Harici Veri Belleği: 64 KiB'a kadar genişletilebilir.

SFR: 128 Bayt

— Özel SFR Kaydedicileri —

P0 (80)H

P1 (90)H

P2 (A0)H

P3 (B0)H

--- (D0)H

PSW (D0)H

ACC (E0)H

B (F0)H

--- (81)H

SP (81)H

DPH (82)H

DPL (83)H

--- (A8)H

IE (A8)H

IP (B8)H

--- (88)H

TCON (88)H

TMOD (89)H

TL0 (8A)H

TL1 (8B)H

TH0 (8C)H

TH1 (8D)H

--- (98)H

SCON (98)H

SBUF (99)H

Bit adreslenebilir değil!

Bit adreslenebilir değil!

Bit adreslenebilir değil!

Bit adreslenebilir değil!

Bit adreslenebilir değil!

Bit adreslenebilir değil!

Bit adreslenebilir değil!

Bit adreslenebilir değil!

Bit adreslenebilir değil!

Bit adreslenebilir değil!

— Notlar —

- 1) PSW kaydedicisinin RS1 ve RS0 bitleri ile istenilen kaydedici bankası seçilir.
- 2) Dolaylı adresleme için sadece R0 ve R1 kaydedicileri kullanılabilir.
- 3) Eğer SFR alanındaki kaydedicinin adresi 0 veya 8 ile bitiyorsa o kaydedici bit adreslenebilirdir.

— Adresleme Yöntemleri —

- 1) Kaydedici Adresleme
- 2) Doğrudan (Direkt) Adresleme
- 3) Dolaylı Adresleme
- 4) İvedi Adresleme
- 5) Bağlı (Göreceli) Adresleme
- 6) Mutlak Adresleme
- 7) Uzun Adresleme
- 8) İndisli Adresleme

— Yönergeler —

- ORG
- END
- DB
- EQU

— Sayı Formatları —

- Onlu : 55 - 55D
- İkili : %1010 - 1010B
- Sekizli : @33 - 33O
- Onaltılı: \$15 - 15H

(80)H

(7F)H

(30)H

(2F)H

(20)H

(1F)H

(18)H

(17)H

(10)H

(0F)H

(08)H

(07)H

(00)H

PSW

CY	AC	F0	RS1	RS0	OV	-	P
7	6	5	4	3	2	1	0