

Ödev 2: 8086 Bellek Haritalama Adres Çözümleme

İsa Koçan - 22011056

Video Linki: <https://youtu.be/R3xuwjkzWCo>

1 Tasarım Mantığı

Bu tasarımda **Separate Bank Decoder** yöntemi kullanılarak 64KB ROM ve 64KB RAM adreslemesi yapılmıştır. 8086 işlemcisi 16-bit veri yoluna sahip olduğu için, bellek sistemi Çift (Even) ve Tek (Odd) olmak üzere iki bank'a ayrılmıştır. Ayrıca Decoder Enable uçlarına M/IO ucu bağlanarak Isolated I/O Map yapısı sağlanmıştır.

1.1 Bank Seçimi

- **Even Decoder (Çift Bank):** Enable ucu A_0 hattına bağlanmıştır. İşlemci çift adrese erişmek istediğiinde ($A_0 = 0$) bu dekoder aktif olur.
- **Odd Decoder (Tek Bank):** Enable ucu \overline{BHE} hattına bağlanmıştır. İşlemci tek adrese erişmek istediğiinde ($\overline{BHE} = 0$) bu dekoder aktif olur.
- **16-Bit Erişim:** İşlemci hem $A_0 = 0$ hem de $\overline{BHE} = 0$ yaptığında, her iki dekoder aynı anda çalışarak 16-bit veri transferini sağlar.

1.2 Adres Çözümleme

Devrenin sade olması için adres bitleri dekoder girişlerine (A, B, C) stratejik olarak yerleştirilmiştir:

- **A_{15} Biti (Dekoder A Giriş):** ROM'un Alt (Low - 20000H) ve Üst (High - 28000H) bloklarını seçmek için kullanılmıştır.
- **A_{18} Biti (Dekoder C Giriş):** ROM (20000H) ve RAM (60000H) adreslerini ayırt etmek için kullanılmıştır.

2 Adres Çözümleme Tablosu

Tablo 1: Adres Çözümleme ve Bank Seçimi Tablosu

Adres Aralığı	A19	A18	A17	A16	A15	BHE	A0	Aktif Çıkış	Seçilen Blok	İşlem
ROM Bloğu (64KB) - (4 Parça)										
20000H - 27FFFFH (Low Blok)	0	0	1	0	0	1	0	Y2 (Even)	ROM00	8-bit Çift
	0	0	1	0	0	0	1	Y2 (Odd)	ROM01	8-bit Tek
	0	0	1	0	0	0	0	Y2 (İkisi)	ROM00 + ROM01	16-bit
28000H - 2FFFFFH (High Blok)	0	0	1	0	1	1	0	Y3 (Even)	ROM10	8-bit Çift
	0	0	1	0	1	0	1	Y3 (Odd)	ROM11	8-bit Tek
	0	0	1	0	1	0	0	Y3 (İkisi)	ROM10 + ROM11	16-bit
RAM Bloğu (64KB) - (2 Parça)										
60000H - 6FFFFH (Tüm RAM Alanı)	0	1	1	0	X	1	0	Y6 AND Y7	RAM00	8-bit Çift
	0	1	1	0	X	0	1	Y6 AND Y7	RAM01	8-bit Tek
	0	1	1	0	X	0	0	Y6 AND Y7	RAM00 + RAM01	16-bit

Not: RAM bloğu 64KB olduğu için adresin ilk yarısında ($A_{15} = 0$) Y6, ikinci yarısında ($A_{15} = 1$) Y7 aktif olmaktadır. Bu çıkışlar bir AND kapısı ile birleştirilerek tüm RAM alanına erişim sağlanmıştır.

3 Simülasyon Sonucu

ROM'dan okuduğumuz yordamlar RAM'e 1234h ve ABCDh yazma amacı taşıyor.

- Düşük anlamlı baytlar (34H ve CDH) → **RAM00** (Çift Bank),
- Yüksek anlamlı baytlar (12H ve ABH) → **RAM01** (Tek Bank)

Simülasyon çıktısında verilerin bu sırayla doğru bir şekilde yazıldığını görmekteyiz.

Memory Contents - RAM00		Memory Contents - RAM01		8086 Variables - U1	
Memory Contents - RAM00					
0000	34 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	CD 00 00 00
0038	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0070	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00A8	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00E0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0118	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0150	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0188	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
01C0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
Memory Contents - RAM00		Memory Contents - RAM01		8086 Variables - U1	
Memory Contents - RAM01					
0000	12 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	AB 00 00 00
0038	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0070	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00A8	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00E0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0118	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0150	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0188	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
01C0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00

Şekil 1: RAM Bellek İçeriği