

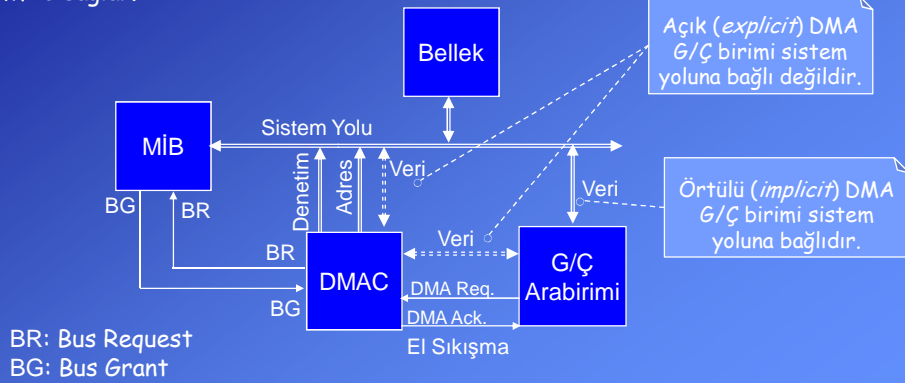
Doğrudan Bellek Erişimi (Direct Memory Access -DMA)

Yoğun (büyük bloklar halinde) veri aktarımı yapan G/Ç birimleri için kullanılır.

Yoklamalı ve kesmeli çalışmada veri aktarımını MİB yapmaktadır.

DMA yönteminde sistemde doğrudan bellek erişimi denetçisi (DMA Controller - DMAC) bulunur.

G/Ç birimi ile bellek arasındaki veri aktarımı MİB üzerinden geçmez. Aktarımı DMAC sağlar.



DMAC (DMA Controller) tipleri:

- Açık DMA (Explicit):** Bellek ile G/Ç arabirimi arasında aktarılan veriler DMAC üzerinden geçer. Önce DMAC veriyi G/Ç arabirminden (ya da bellekten) okur ve belleğe (ya da G/Ç arabirmine) yazar.
- Örtülü DMA (Implicit):** Aktarılan veriler DMAC üzerinden geçmez. DMAC sadece belleğe giden adres ve denetim işaretlerini çıkarır. Veri aktarımı bellek ile G/Ç arabirimi arasında doğrudan yapılır.

İki aktarım şekli mümkün:

- Blok aktarımı (Burst mode):** DMAC yolu aldıktan sonra daha önceden belirlenmiş olan blok boyu kadar veri aktarır ve bu sürede yolu elinde tutar.

Blok aktarımı bitince yol MİB'e verilir.

- Çevrim çalma (Cycle stealing):** DMAC yolu ister ve bir sözcük aktarır yolu MİB'e geri verir.

Hatırlatma: MİB, komut alma, operand alma ve eğer gerekli ise operand yazma çevrimlerinde belleğe erişir.

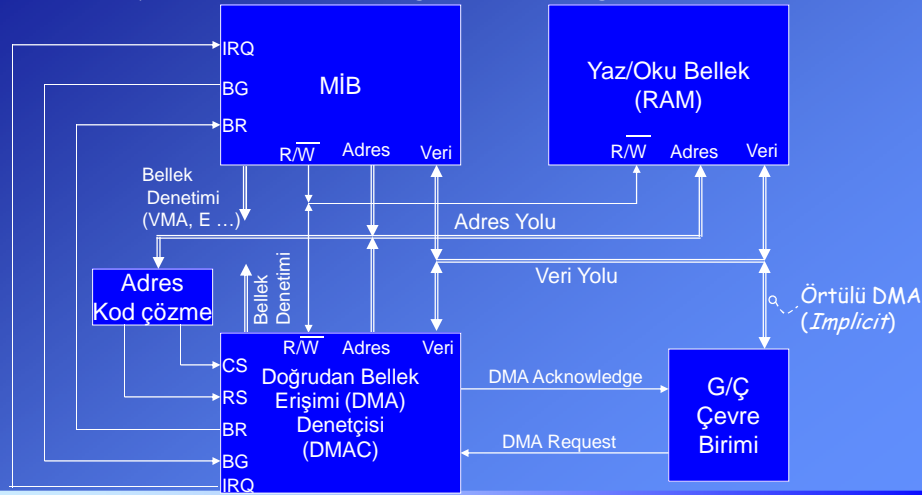
Bunun dışındaki komut çözme çevriminde ve iç saklayıcılar üzerindeki hesaplamalar sırasında MİB belleğe erişmez. Bu çevrimlerde DMAC aktarım yaparken MİB de kendi işine devam edebilir.

Örnek Bir MIB-DMAC Bağlantısı:

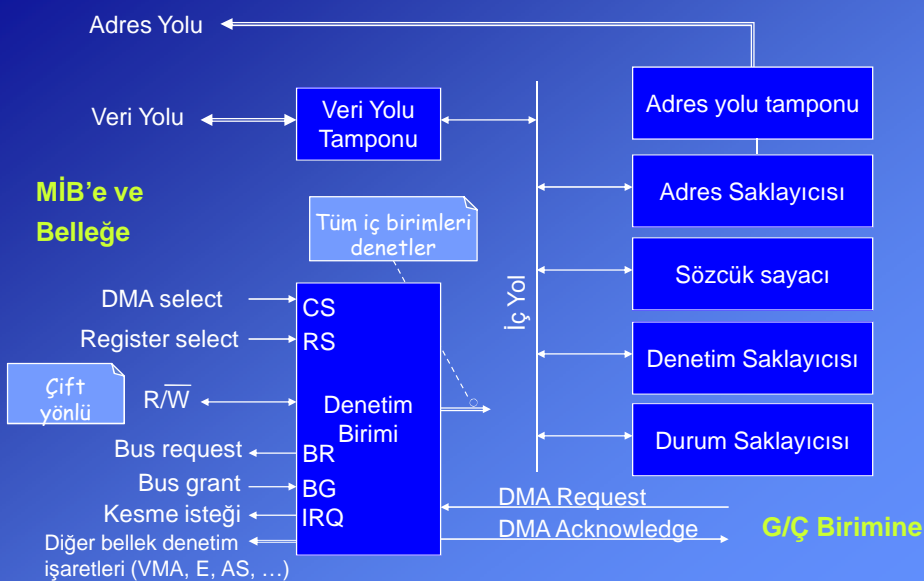
İletişim başlamadan önce DMAC MİB tarafından programlanır (ilgili saklayıcılarına gerekli bilgiler yüklenir).

Bu aşamada DMAC sistemde bir bellek ya da G/Ç birimi gibi davranır.

DMAC sistem yolunu aldıktan sonra belleği adresler (MİB gibi davranır).



Örnek Bir DMAC İç Yapısı:



DMA yönteminde veri aktarımının adımları:

1. DMA denetçisi (DMAC), MİB tarafından programlanır: *G/Ç* adresi, bellek adresi, oku/yaz, veri miktarı, çalışma şekli vs.
Bu aşamada DMAC MİB tarafından adreslenen bir bellek ya da *G/Ç* birimi gibi davranır.
2. *G/Ç* birimi DMAC'dan istekte bulunur (DMA Request).
3. DMAC, BR'yi etkin yaparak MİB'den sistem yolunu ister.
4. MİB yol çevrimini (komutu değil) tamamlar, sistem yolundan çekilir (3. konuma - yüksek empedans- geçer), yolu verdiğini BG ile bildirir.
5. Bu aşamada sistem yolunun sahibi DMA denetçisidir. Belleği (MİB'in yaptığı şekilde) uygun işaretlerle adreslemek denetçinin görevidir.
DMAC adres, R/W ve bellek erişimi için gerekli diğer denetim işaretlerini çıkartır. *G/Ç* birimini uyarır (DMA Acknowledge). *G/Ç* birimi veri aktarımını yapar (örtülü "implicit").
Açık DMA (*explicit*) yöntemde DMAC *G/Ç* birimini uyarır (DMA Acknowledge), veriyi *G/Ç* biriminden alır ve belleğe yazar.
Veya DMAC bellekten veriyi okur, *G/Ç* birimini uyarır (DMA Acknowledge), veriyi *G/Ç* birimine yazar.

DMA yönteminde veri aktarımının adımları (devam):

6. Eğer *G/Ç* biriminden yeni istek geldiyse (DMA Request) , DMAC BR'yi etkin tutar.
Burst mode: Blok tamamlanıncaya kadar BR etkin kalır, yol bırakılmaz.
Cycle stealing: Bir çevrim MİB'e bırakılır, sonra yol tekrar istenir.
7. *G/Ç* biriminden istek (DMA Request) gelmiyorsa veya tüm blok tamamlandıysa (DMAC sözcük sayacı sıfırlanmıştır) DMAC sistem yolundan çekilir, yani sürdüğü denetim hatlarını 3ncü konuma çeker.
DMAC BR'yi etkisiz yapar. Sistem yolu tekrar MİB'in kontrolüne geçer.
8. Eğer kesme üretecek şekilde koşullandıysa, DMAC MİB'e kesme göndererek tamamlan iş hakkında rapor verebilir. MİB kesme hizmet programında DMAC'ın durum saklayıcılarını okuyarak aktarım hakkında bilgi alabilir.
DMA yönteminde kesmelerin yolun alınıp verilmesiyle ilgisi yoktur.

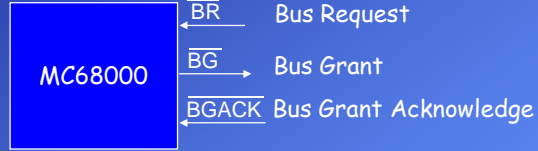
Üç Hatlı Doğrudan Bellek Erişimi

Önceki bölümde anlatılan iki hatlı (BR, BG) DMA erişiminden farklı olarak üç hat ile çalışan DMA denetçileri de vardır.

Birden fazla denetçinin bulunduğu sistemler için uygundur.

Örnek MC68000:

MC68000'in DMA hatları üç hatlı olarak hazırlanmıştır.



MC68000'in DMAC ile çalışmasının adımları:

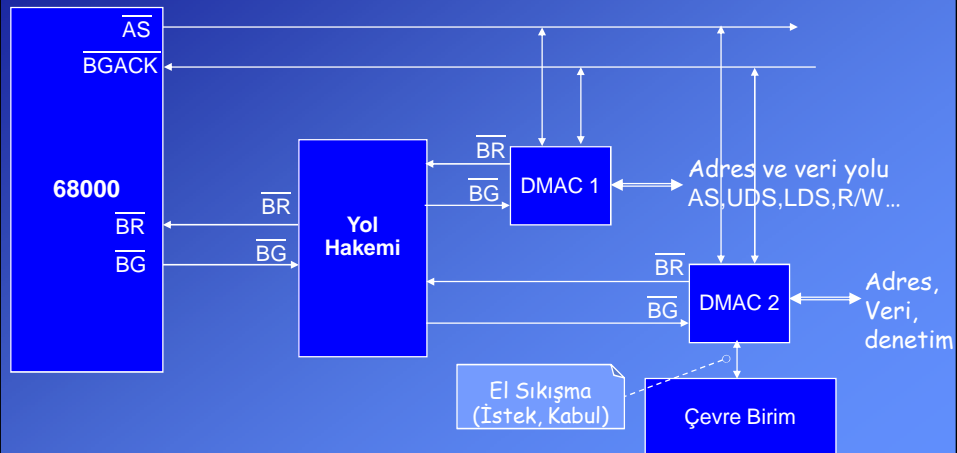
1. DMAC BR'yi etkin yaparak yolu ister.
2. 68000 hemen BG'yi etkin yapar. Bu yolu verdiği anlamına gelmez. Yolu vermeye hazır olduğunu, bir sonraki yol sahibinin belirlenebileceğini belirtir.
3. Eğer birden fazla DMAC varsa yol hakemi bir sonraki yol sahibini belirler. Bir sonraki yol sahibi yol çevriminin bitmesini bekler. AS, DTACK etkisiz olmalıdır. Ayrıca BGACK kontrol edilir: Yolu bir önceki sahibi yolu bıraktı mı?
4. Yolu sahibi BGACK işaretini etkin yapar ve yolu kullandığı sürece bu konumda tutar. BR etkisiz yapılır, böylece sıradaki birim istekte bulunabilir.
5. Veri aktarımı tamamlandıncaya BGACK etkisiz yapılır. BGACK girişi etkin olduğu sürece 68000 yol erişimi yapamaz.



Yol Hakemi (Bus Arbiter)

Sadece bir MİB ve bir DMA denetçisi olan sistemlerde sistem yolunun kullanılması konusunda DMA denetçisinin önceliği vardır.

Sistem yolunu kullanan birden fazla yol denetçisinin olduğu durumlarda bir yol hakemine (*bus arbiter*) gerek vardır.

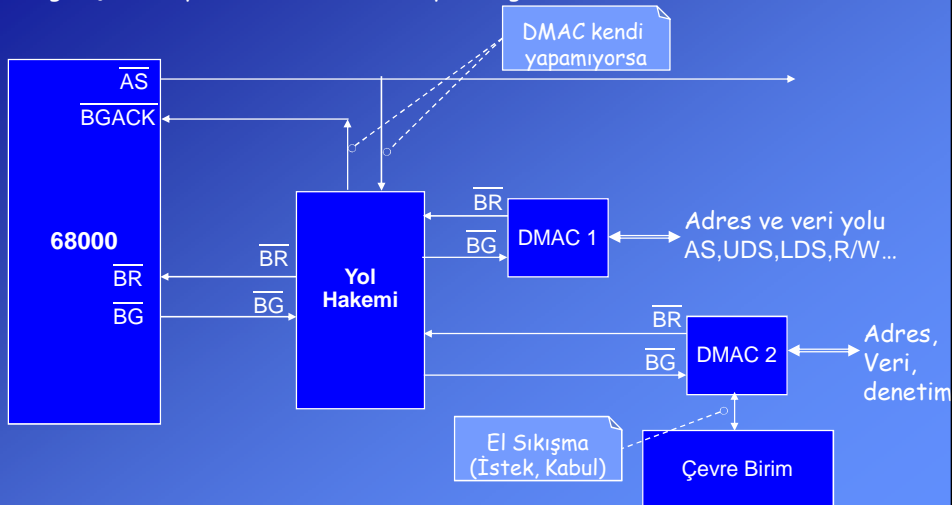


Sistemin Çalışması:**DMA denetçisi üç hatlı (BGACK) yapıyı kullanıyorsa**

1. DMA denetçisi isteğini yol hakemine BR ile iletir.
2. Yol hakemi isteği 68000'e (BR) iletir.
3. Yol hakemi 68000'den gelen kabul işaretini (BG) istekte bulunan en yüksek öncelikli DMA denetçisine iletir.
4. BG onayını alan DMA denetçisi AS ve BGACK işaretlerini gözleyerek bir önceki yol çevriminin tamamlanmasını ve yolun önceki sahibinin yolu bırakmasını bekler. AS bir önceki yol çevriminin bitip bitmediğini, BGACK ise bir önceki yol sahibinin yolu bırakıp bırakmadığını gösterir.
5. DMA denetçisi yolu aldıktan sonra BGACK işaretini kendisi etkin yaparak yolu ele geçirdiğini 68000'e ve diğer DMA denetçilerine bildirir. Yolu ele geçiren DMA denetçisi BR çıkışını etkisiz yaparak başka denetçilerin istekte bulunmasına olanak sağlar. Ancak aktarımı bitene kadar BGACK işaretini etkin tuttuğu için yol denetimi kendisinde kalır. Bundan sonra sistem yolunun sahibi DMA denetçisidir ve tüm bellek erişim işlerini yapmak (adresleme, AS, UDS, LDS, VMA, R/W ...) onun sorumluluğundadır.
6. Tüm DMA denetçileri aktarımlarını tamamladıklarında 68000'in BGACK girişi etkisiz olur. Bu durumda 68000 tekrar sistem yolunu kullanmaya başlar.

İki hat kullanan (BR,BG) Denetçilerin 68000'e bağlanması:

Yol denetimi için sadece iki hat kullanan (BR, BG) ve BGACK işlevlerine sahip olmayan DMA denetçilerin 68000 mikroişlemcisine bağlanması durumunda BGACK ile ilgili işlemler yol hakemi tarafından yerine getirilmelidir.

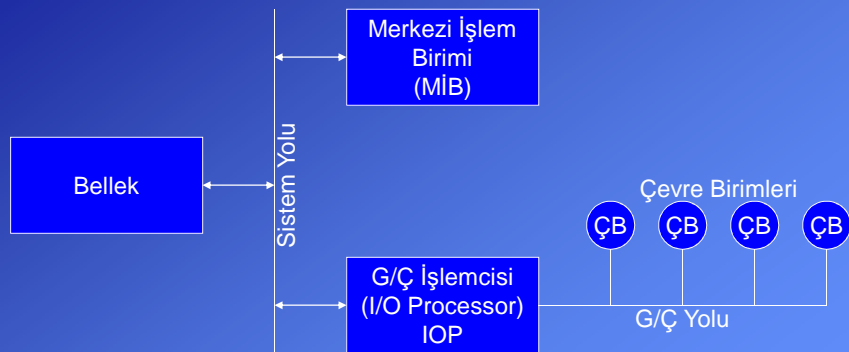


Sistemin çalışması:**DMA denetçisi sadece iki hat kullanıyorsa**

1. DMA denetçisi isteğini (BR) yol hakemine iletir.
2. Yol hakemi isteği 68000'e (BR) iletir.
3. 68000'den kabul işareti (BG) geldikten sonra yol hakemi AS işaretini gözleyerek bir önceki yol çevriminin tamamlanmasını ve (eğer varsa) yolun önceki sahibinin isteğinin (BR) sona ermesini bekler.
Bu yöntemde BGACK işaretini yol hakemi kendi sürdüğü için bu hattı yoklamasına gerek yoktur.
4. Yol hakemi istekte bulunan en yüksek öncelikli DMA denetçisine kabul işaretini (BG) iletilir.
5. Yol hakemi BGACK işaretini etkin yapar böylece yolun bir DMA denetçisi tarafından alındığını 68000'e bildirir.
6. Bu tür DMA denetçileri iletişim sürdüğü sürece istek çıkışlarını (BR) etkin tutarlar.
Bu durumda yol hakemi de BGACK işaretini etkin tutar.
7. İletişim işi sona eren DMA denetçisi istek çıkışını (BR) etkisiz yapar.
Eğer başka bir denetçiden gelen istek yoksa yol hakemi BGACK işaretini etkisiz yaparak 68000'in yolu almasını sağlar.

G/Ç İşlemcisi (I/O Processor)

G/Ç işlemleri için özelleşmiş (özel komutları) olan bir mikroişlemcidir. Kendi içinde mikroişlemci ve DMA denetçisine sahiptir. G/Ç ile ilgili işleri MİB yerine bu işlemci yapar.

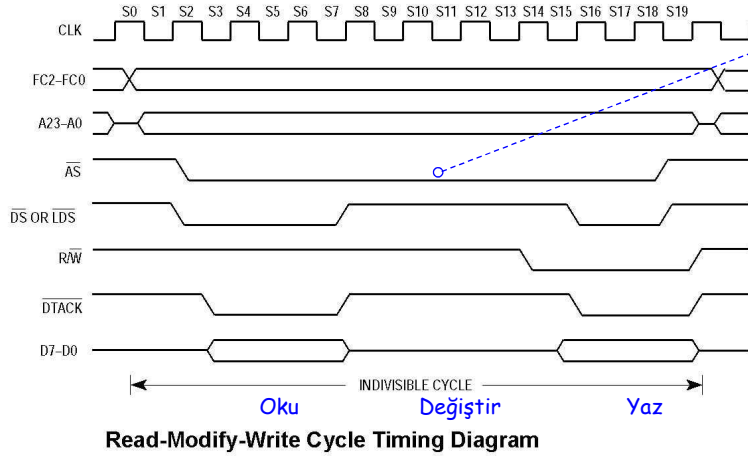


68000'de Bölünemez Yol Çevrimi

Semafor işlemlerinde Test-And-Set (TAS) komutu kullanılır.

Aynı yol çevriminde bellek gözü okunur, değerlendirilir ve güncellenir.

Read, modify, write cycle:



AS sürekli etkin olduğu için bir DMAC yolu alamaz.

Üç işlem aynı komut içinde yapıldığı için kesme ile de bölünemez.

TAS Test and set an operand

Yazım: TAS <ea>

İşlem: [CCR] ← tested([operand]); [destination(7)] ← 1

Operandı test eder: Verinin değerine göre Z ve N bayraklarını etkiler.

Operandın en yüksek anlamlı bitini 1'ler.

Bu işlemler bölünemez.

Semafor işlemleri için kullanılır.

Örnek1: TAS komutu olmadan kritik bölge denetimi (sakıncalı)

TEST	TST.B	BAYRAK	
	BMI	TEST	
KRITIK	OR.B	#\$80,BAYRAK	Semafor kapatılıyor
		Kritik bölge
		
SON	CLR.B	BAYRAK	Semafor açılıyor

Bu iki komut bölünebildiği için bu program sağlıklı olarak çalışmaz.

Örnek2: TAS komutu ile kritik bölge denetimi (doğru)

TEST	TAS	BAYRAK	Semafor test ediliyor ve gerekiyorsa kilitleniyor
	BMI	TEST	
KRITIK		Kritik bölge
		
SON	CLR.B	BAYRAK	Semafor açılıyor