



STM32 MEMORY LAYOUT

Mehmet Faruk Gül - 031790044

STM32 Nedir?



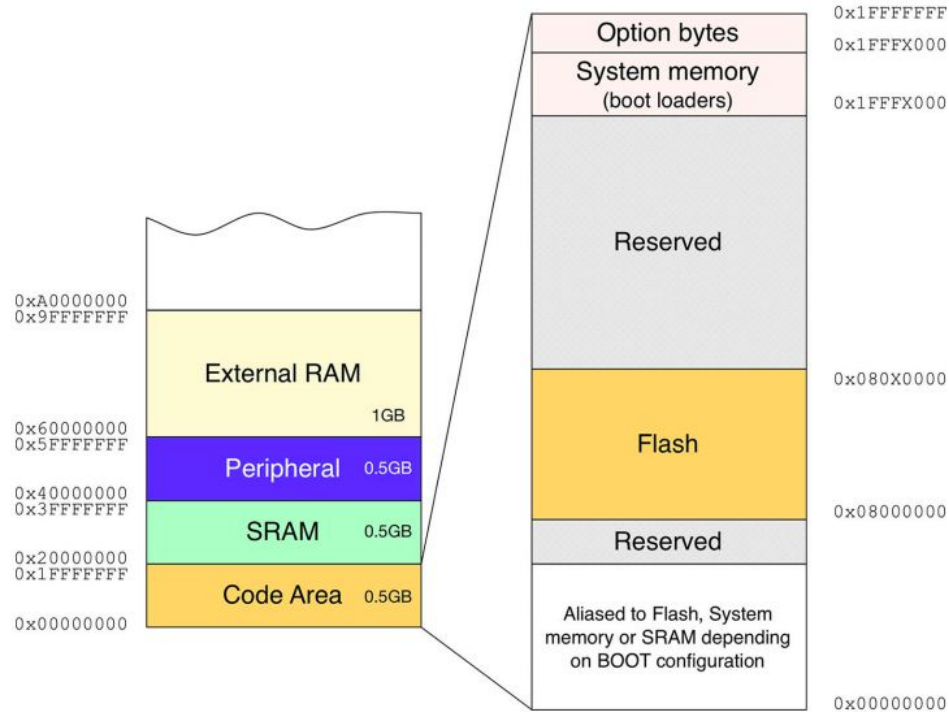
- STM32, belirli bir mikrodenetleyiciye verilen bir isim değildir. STMicroelectronics'in ARM Cortex-M tabanlı mikrodenetleyici ailesine verdiği isimdir. STM32 ailesi birden fazla gruba bölünebilir ve bu da serilere bölünebilir.
- Yeni başlayanlar için diğer serilere göre daha uygun iki seri var. STM32F1 serisi ve STM32F4 serisi. Sebebi her iki seri de uzun süredir piyasada ve hobiler arasında hala popüler. Bu, bu serideki cihazlar için çok sayıda kaynak olduğu anlamına gelir. Ayrıca STM32F1 serisindeki cihazlar diğer birçok parça kadar karmaşık değildir.

IDE'ler



- IDE, Entegre Geliştirme Ortamı anlamına gelir. Bu genellikle, bir IDE kurulduktan sonra başka herhangi bir işlem gerekmeden kullanılmaya hazır olduğu anlamına gelir. IDE'ler, derleyiciler ve sürücüler gibi gerekli tüm yazılımları kurulduktan sonra yükler.
- STM32 için kullanabileceğiniz IDE'ler:
 - Arm Keil MDK - STM32G0, STM32F0 ve STM32L0 serisi için ücretsiz (Windows)
 - PlatformIO IDE - Ücretsiz (Windows, Linux, macOS)
 - STM32CubeIDE - Ücretsiz (Windows, Linux, macOS)
 - Segger Embedded Studio - Ticari olmayan kullanım için ücretsiz (Windows, Linux, macOS)
 - SW4STM32 - Ücretsiz (Windows, Linux, macOS)

STM32 Memory Layout Örnek Görsel



STM32 Memory Layout

- Bağlayıcı komut dosyası, tüm program bölümlerini (kod ve veri) ayarlamak için derleyici tarafından kullanılan bellek düzenini tanımlar. Cortex-M3, aşağıdaki resimde gösterildiği gibi iyi tanımlanmış bir bellek (SRAM) düzenini takip eder.

Address	Segment Type
0x20000000	Data
	BSS Segment
↓	HEAP
↑	STACK

STM32 Memory Layout



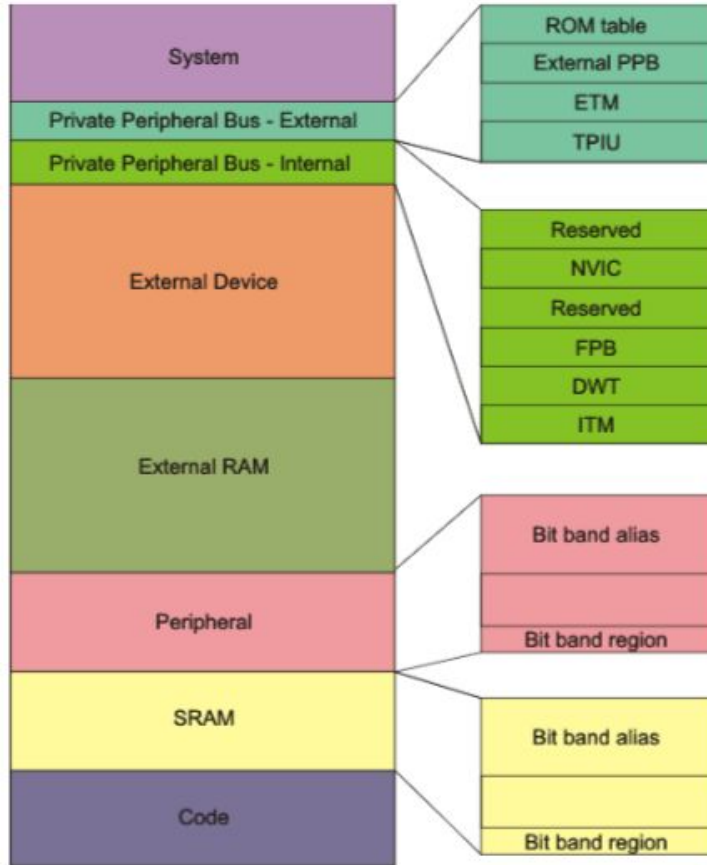
- Veri segmenti, uygulama kodu tarafından başlatılan global ve statik değişkenleri depolar.
- BSS segmenti, uygulama kodu tarafından başlatılmayan global ve statik değişkeni depolar. Derleyici bu verileri 0 olarak başlatır.
- HEAP, BSS segmentinden sonra tahsis edilir ve yukarıdaki resimde gösterilen yönde büyür.
- STACK, SRAM bölgesinin sonunda tahsis edilir ve başlangıç adresi SRAM'ın bitiş adresine eşittir. HEAP yönünün tersidir.

Cortex-M3 ve M4 Memory Organization



- Cortex-M3 ve Cortex-M4 önceden belirlenmiş bir hafıza yapısına sahiptir. Bu, kesme denetleyicisi ve hata ayıklama bileşenleri gibi yerleşmiş halde olan çevre birimlerine bellek erişim talimatlarıyla erişilmesini sağlar. Bu sayede çoğu sistem özelliğine program kodunda erişilebilmektedir. Önceden tanımlanmış bellek haritası ayrıca Cortex-M3 işlemcinin çip üzerinde sistem (SoC) tasarımlarında hız ve entegrasyon kolaylığı için yüksek düzeyde optimize edilmesini sağlar.
- Genel olarak, 4 GB bellek alanı, aşağıdaki resimde gösterildiği gibi aralıklara bölünebilir. Cortex-M3 tasarımı, bu bellek kullanımı için optimize edilmiş bir dahili veri yolu altyapısına sahiptir.

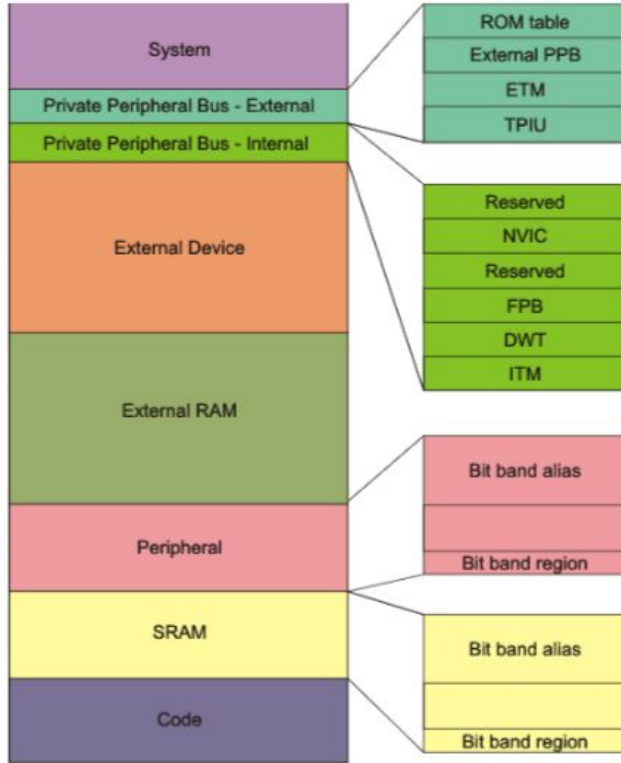
ARM belleğinin grafiksel gösterimi aşağıdaki resimde gösterilmektedir:



ARM Cortex-M3 belleği aşağıdaki bölgelere ayrılmıştır:

- Sistem
- Özel Çevresel Veri Yolu - Harici - Şunlara erişim sağlar:
 1. İzleme Bağlantı Noktası Arabirim Birimi (TPIU),
 2. Gömülü İz Makrohücre (ETM),
 3. ROM tablosu,
 4. PPB bellek haritasının uygulamaya özel alanları.
- Özel Çevresel Veri Yolu - Harici - Şunlara erişim sağlar:
 1. Enstrümantasyon İzleme Makro Hücre (ITM),
 2. Veri İzleme Noktası ve İzleme (DWT),
 3. Flashpatch ve Breakpoint (FPB),
 4. MPU ve İç İç Vektörlü Kesme Denetleyicisi (NVIC) dahil olmak üzere Sistem Kontrol Alanı (SCS).
- Harici Aygıt - Bu bölge, harici aygıt belleği için kullanılır.
- Harici RAM - Bu bölge veri için kullanılır.

Devam



- Çevre Birimi - Bu bölge, bit bandı ve bit bandı takma ad alanlarını içerir.
 1. Çevresel Bit-bant takma adı - Bu bellek aralığına doğrudan erişimler, çevresel bellek erişimleri gibi davranır, ancak bu bölge aynı zamanda bit-bant takma adıyla bit adreslenebilir.
 2. Çevresel bit bandı bölgesi - Bu bölgeye veri erişimleri, bit bandı bölgesine yeniden eşlenir. Okuma-değiştir-yaz şeklinde bir yazma işlemi gerçekleştirilir.
- SRAM - Bu yürütülebilir bölge, veri depolama içindir. Kod burada da saklanabilir. Bu bölge, bit bandı ve bit bandı takma ad alanlarını içerir.
 1. SRAM Bit-bant takma adı - Bu bellek aralığına doğrudan erişimler, SRAM bellek erişimleri gibi davranır, ancak bu bölge aynı zamanda bit-bant takma adıyla bit adreslenebilir.
 2. SRAM bit-band bölgesi - Bu bölgeye veri erişimleri, bit band bölgesine yeniden eşlenir. Okuma-değiştir-yaz şeklinde bir yazma işlemi gerçekleştirilir.
- Kod - Bu yürütülebilir bölge program kodu içindir. Veriler burada da saklanabilir.

Hafıza Haritaları

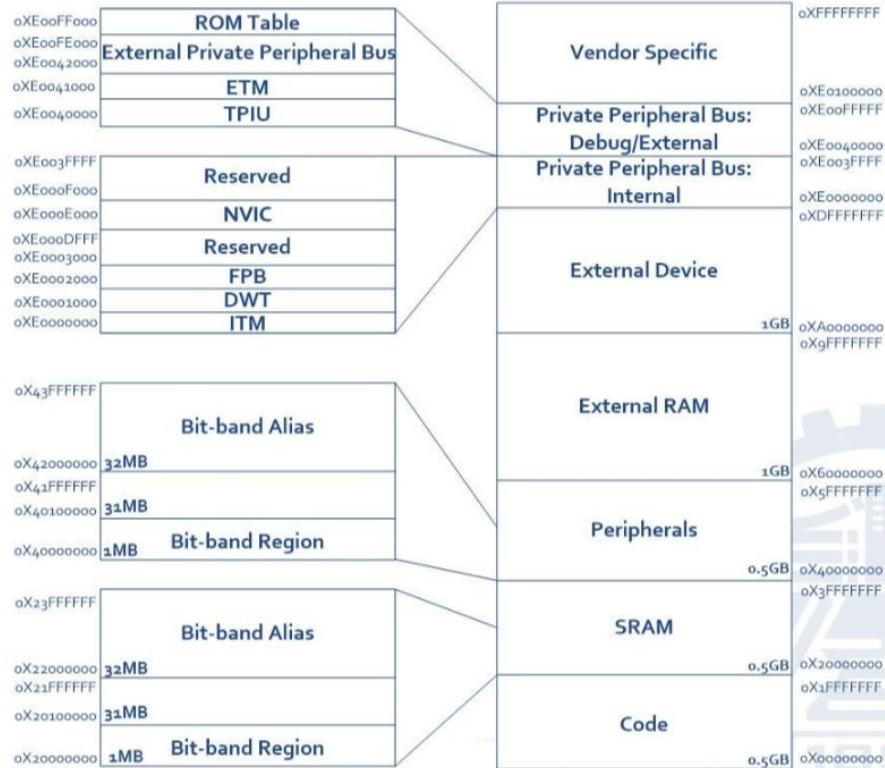


Cortex-M3 işlemcinin sabit bir bellek haritası vardır. Bellek konumlarından bazıları, hata ayıklama bileşenleri gibi özel çevre birimleri için ayrılmıştır.

1. Getirme Yama ve BreakPoint Birimi (FPB)
2. Veri İzleme Noktası ve İzleme Birimi (DWT)
3. Enstrümantasyon İz Makrocell (ITM)
4. Gömülü İz Makrosel (ETM)
5. İzleme Bağlantı Noktası Arayüz Birimi (TPIU)
6. ROM Tablosu

Cortex-M3 işlemci toplam 4 GB adres alanına sahiptir.

Hafıza Haritaları Görsel

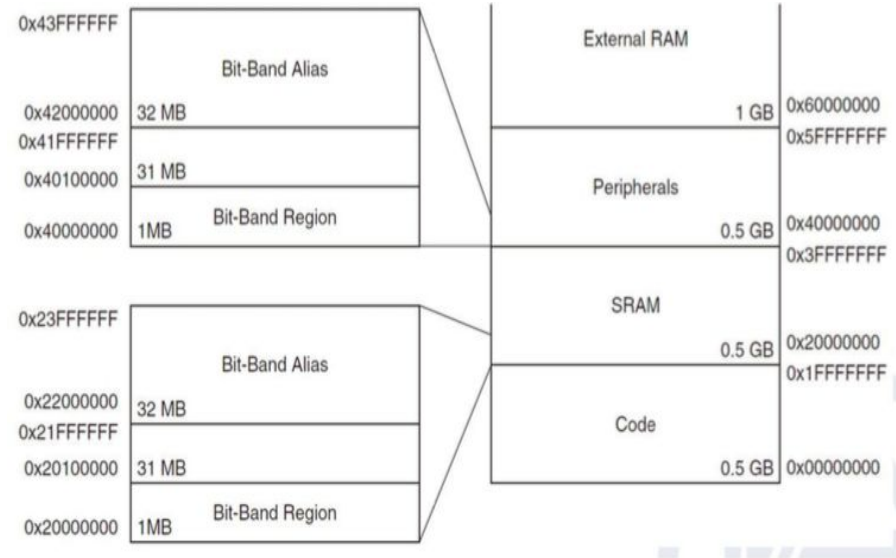


Bit-Bant İşlemleri

Bit-bant işlem desteği, tek bir veri bitine erişmek için tek bir yükleme/depolama (okuma/yazma) işlemine izin verir.

Bit bant bölgeleri:

1. SRAM bölgesinin ilk 1 MB'ı
2. Çevre bölgesinin ilk 1 MB'ı



Bit-Bant İşlemleri

**Remapping of Bit-Band Addresses
in SRAM Region**

Bit-Band Region	Aliased Equivalent
0x20000000 bit[0]	0x22000000 bit[0]
0x20000000 bit[1]	0x22000004 bit[0]
0x20000000 bit[2]	0x22000008 bit[0]
...	...
0x20000000 bit[31]	0x2200007C bit[0]
0x20000004 bit[0]	0x22000080 bit[0]
...	...
0x20000004 bit[31]	0x220000FC bit[0]
...	...
0x200FFFFC bit[31]	0x23FFFFFC bit[0]

**Remapping of Bit-Band Addresses
in Peripheral Memory Region**

Bit-Band Region	Aliased Equivalent
0x40000000 bit[0]	0x42000000 bit[0]
0x40000000 bit[1]	0x42000004 bit[0]
0x40000000 bit[2]	0x42000008 bit[0]
...	...
0x40000000 bit[31]	0x4200007C bit[0]
0x40000004 bit[0]	0x42000080 bit[0]
...	...
0x40000004 bit[31]	0x420000FC bit[0]
...	...
0x400FFFFC bit[31]	0x43FFFFFC bit[0]

Teşekkür ederim

Mehmet Faruk Gül - 031790044
