

Giriş

5 Mayıs 2021 Çarşamba

08:02

Giriş

Kaynaklar

- Bu belge oluşturulurken <https://www.udemy.com/course/stm32f4-discovery-kart-ile-arm-dersleri/> linkteki eğitim kursu izlenirken alınan notlardan oluşmaktadır.

Giriş

- Gömülü yazılım, belirli bir **donanım üzerinde** belirli görevleri yerine getirmek üzere tasarlanmış, genellikle **gerçek zamanlı çalışan** özel amaçlı yazılımlardır. Bu yazılımlar; ev aletlerinden otomobillere, endüstriyel kontrol sistemlerinden tıbbi cihazlara kadar birçok cihazın içinde yer alır ve donanımın beyni olarak görev yapar.
- Gömülü sistemlerde, yazılım doğrudan **mikrodenetleyici (MCU)** veya **mikroişlemci (MPU)** gibi donanım bileşenleri üzerinde çalışır. Bu tür sistemlerde performans, bellek ve enerji tüketimi gibi sınırlamalar dikkate alınarak optimize edilmiş yazılımlar geliştirilir. Gömülü yazılım; donanımı doğrudan kontrol edebilen, kesme yönetimi yapabilen, haberleşme protokollerini işleyebilen ve genellikle işletim sistemi olmadan (**bare-metal**) ya da gerçek zamanlı işletim sistemi (**RTOS**) ile çalışan yapılardır.
- Gömülü yazılım geliştirme süreci; donanımı tanımayı, düşük seviyeli programlamayı, **hata ayıklamayı** ve sistem kaynaklarını **verimli kullanmayı** gerektirir. Bu alandaki yazılımcılar, yalnızca **yazılım** bilgisine değil, aynı zamanda **donanım** bilgisine de sahip olmalıdır.
- Bu kitapçıkta, gömülü yazılım dünyasına adım atmak isteyenler için temel kavramlardan başlayarak pratik uygulamalara kadar birçok konuyu ele alacağız. Amacımız, yazılım geliştiricilere hem teorik hem de uygulamalı bir bakış açısı kazandırmaktır.

Temel Kavramlar

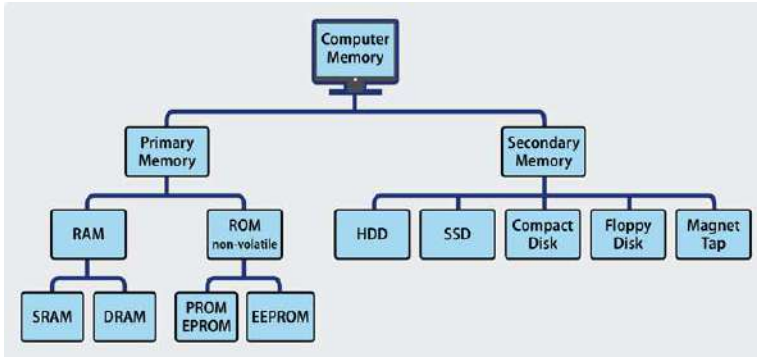
- [STM32 Donanım Soyutlama Katmanı \(HAL\) Yazısı](#) linkten donanım soyutlama katmanı hakkındaki yazıyı okuyabilirsiniz.
- [Gömülü Sistem, Mikroişlemci, Mikrodenetleyici Nedir?](#) ile [Mikroişlemci Hakkında Bilgilendirme](#) linkteki Gömülü Sistem, Mikroişlemci, Mikrodenetleyici nedir sorularına cevap veren yazıları okuyabilirsiniz.

Mikroişlemci

- Mikroişlemci yapısında bir **CPU** (Central Processing Unit), Ön bellek ve Input/Output (Giriş/Çıkış) birimleri bulunan devrelere **microprocessor** denir.
Bu üç temel unsur birbirlerine bus, **iletişim yolları** ile bağlıdır.
- Mikroişlemcinin beyni CPU'dur. Veri akışı ve veri işleme bu birim sayesinde gerçekleşir.
- Bu veri işleme genellikle CPU içerisinde yer alan ALU (Aritmetic Logic Unit)'da uygulanır. Bu birimde sayısal ve lojik işlemler yapılır. Tüm dijital elektronik işlemler CPU'ların en temel işlemleridir.
- CPU'ların içerisinde 8-16-32-64 bitlik registerler bulunmaktadır. Register, bilgilerin geçici sürede depolanmasını sağlarlar.
- CPU'lar, mikroişlemcinin hafızasındaki programları bulma, çağırma ve onları çalıştırma görevi görürler. Veri İşleme Adımları; Veriyi Getirmek (Fetch), Veriyi Çözmek (Decode), Veriyi İşlemek (Execute), Veriyi Hafızata, Geri Depolamak (Store)
- Merkezi işlem birimi, **üç** birimden oluşur.
 - Aritmetik Mantık Birimi (ALU)**, hafıza biriminden gelen verilerin işlenmesinde görev alır. Bu işlemlerise aritmetik olarak toplama, çıkarma, bölme ve çarpma. İkili sayı tabanındaki (binary) mantık işlemleri VE (AND), OR (VEYA) ve bit kaydırma işlemleridir.
 - Kayıtçılar (Registers)**, hafızadaki veriler ALU tarafından işlenirken kullanılan geçici ve kalıcı saklayıcılar. Registerler işlemcinin çekirdeğinde olduklarından verilere ulaşmak daha hızlı gerçekleşir.
 - Kontrol Birimi (CU)**, işlemcinin çalışmasını yönlendiren birimdir. İşlemci içerisindeki ve dışarısındaki birimlerin senkron şekilde çalışmasını sağlar.

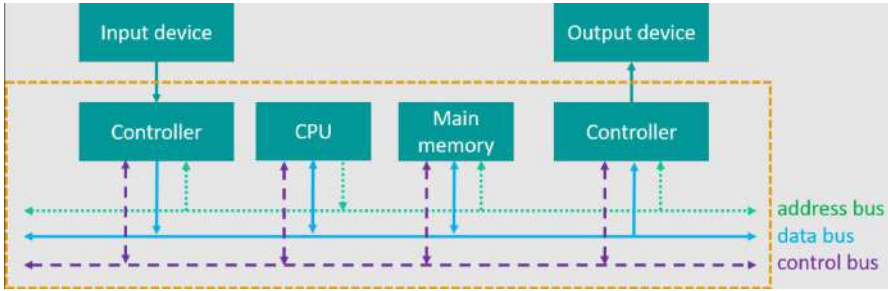
Bellek Yapıları





- RAM, ROM ve EEPROM, hafızanın temel birimleridir.
- Mikroişlemciye atılan veriler ilk olarak hafızaya gelir ve burada depolanır. CPU'ların doğrudan eriştiği birim bellektir. Bellekte iki tane birincil hafıza birimi vardır.
 - **RAM (Random Access Memory)**, mikroişlemcinin elektrik alması durumunda **geçici hafıza** olarak kullandığı birimdir. Elektrik kesildiği zaman bu veriler silinir ve bir daha kullanılmaz. RAM, diğer hafıza birimleri gibi verileri önceden verilen bir sırayla dizmez. Bu sebeple ismi rastgele erişim bellek olarak konulmuştur. RAM, **Dinamik** Rastgele Erişim Bellek ve **Statik** Rastgele Erişim Bellek olmak üzere **ikiye** ayrılır.
 - **ROM (Read Only Memory)**, **sadece okunabilir** bir bellektir. Elektrik kesildiğinde bu bellekteki veriler silinmez. ROM üzerindeki yazılmış fabrikasyon yazılımlar kullanıcılar tarafından değiştirilip, silinemez.
 - **EEPROM (Electrically Erasable Programmable Read-Only Memory)**, elektrik ile defalarca **yazılıp silinebilen** bellektir. Elektrik kesildiğinde bu bellekteki veriler silinmez. Flash belleklerde bir eeprom türüdür.

Veri Yolları



- Giriş - Çıkış birimleri mikroişlemci ile dış dünyanın sinyaller aracılığı ile haberleştiği birimdir. Bu giriş ve çıkışlar; Giriş/ Çıkış portları, harici elektronik birimler, fiziksel cihazlar ve yazılımlar olabilir.
- CPU'daki veri akışının aktarılması, bellek ve Giriş/Çıkış birimlerinin bağlantılarını sağlayan üç çeşit bus vardır.
 - **Address Bus**, verinin okunacağı veya verinin yazılacağı bölgeyi belirten adres bilgilerinin taşınmasını sağlar. Tek yönlü bir veri yoludur.
 - **Data Bus**, CPU'dan bellek ve Giriş/Çıkış portlarına veya bu birimlerden CPU'ya çift yönlü bir hat vardır.
 - **Control Bus**, mikroişlemciye birimler arasında iletişimi sağlayan sinyalleri ileten, kontrol eden veri hattıdır. Her mikroişlemci farklı sayıda Control Bus'a sahiptir.

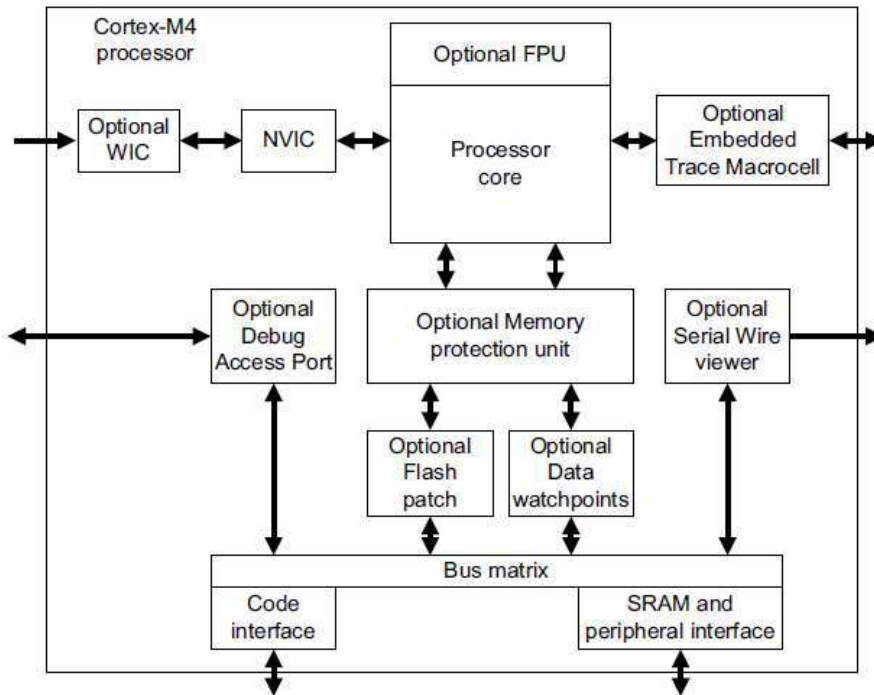
Mikrodenetleyici

- Mikroşlemcili bir sistemin içerisinde bulunması gereken temel bileşenlerden RAM, ROM, ALU, kontrol ünitesi ve I/O ünitesini tek bir çip içerisinde barındıran entegre devreye **microcontroller** denir.
- Mikrodenetleyici, dışarıdan gelen bir veriyi hafızasına alan, derleyen ve sonucunda çıktı elde eden bir bilgisayardır. Mikrodenetleyicilerin yapısında; CPU, RAM, ROM, I/O Portları, Seri ve Paralel Portlar, Zamanlayıcılar, ADC ve DAC çevre birimleri
- Mikrodenetleyiciler gerçek zamanlı işlemlerde oldukça başarılılardır.
- Mikrodenetleyiciler herhangi bir işi çok küçük boyutlarda ve daha düşük enerjide yaparlar.
- Mikroşlemcili ile kontrol edilecek bir sistemi kurmak için gerekli olan minimum donanımda CPU, RAM, I/O bulunmalıdır. Bunlar arasında veri alışverişini sağlamak için ise veri yolu, adres yolu ve kontrol yolu gereklidir. Birimler arasındaki iletişimi sağlayan bu yolları yerleştirmek içinde bir anakart gereklidir.
- Mikrodenetleyici ile kontrol edilecek sistemde ise yukarıda saydığımız birimler tek zaten mikrodenetleyici içerisinde bulunmaktadır. Bu da maliyetin daha düşük olacağı anlamına gelir.

- Mikrodenetleyiciler çok az sayıda ve karmaşık olmayan komutlarla programlanabilen sistemlerdir.
- Mikronetleyiciler hız bakımından mikroişlemcilerden daha hızlıdır. Güç tüketimi mikrodenetleyicilerde daha azdır. Fiyatları mikroişlemcilere göre daha uygundur.
- Mikroişlemcili sistemlerde harici donanım desteği gerekli iken, mikrodenetleyicilerde bu gereklilik çok azdır.

İşlemci ve Çekirdek Bileşenleri

- Cortex-M4 işlemcisi ARM tarafından tasarlanmış, gömülü sistemlerde çok yaygın kullanılan 32-bit RISC tabanlı bir mikrodenetleyici çekirdeğidir.
 - **Mimari:** ARMv7-M mimarisine dayanır.
 - **32-bit:** Veri yolu, adresleme ve register yapısı 32-bit'tir.
 - **Performans:** Tek çevrimlik çoğu talimat ile yüksek verimlilik sağlar.
 - **Donanım Çarpma/Bölme:** 32x32 → 64 bit çarpma ve hızlı bölme desteği vardır.
 - **DSP Özellikleri:** FIR/IIR filtreleri, FFT gibi sayısal sinyal işleme uygulamaları için özel SIMD (Single Instruction, Multiple Data) talimatları bulunur.
 - **Floating Point Unit (FPU):** Bazı modellerde donanım tabanlı kayan nokta birimi (single precision) entegre gelir.
 - **Düşük Güç:** Gömülü sistemler için optimize edilmiş düşük güç modları bulunur.
 - **Kesme Sistemi:** Nested Vectored Interrupt Controller (NVIC) ile çok hızlı ve düşük gecikmeli kesme yönetimi yapılır.



- Cortex çekirdeği, üreticiye özgü çevre birimlerinden bağımsız olarak her cihazda bulunan bazı **çekirdek çevre birimleri (core peripherals)** içerir. Bu birimler, işlemcinin kesme yönetimi, zamanlama, bellek koruması ve matematiksel hesaplama gibi sistem seviyesinde kritik görevlerini üstlenir.

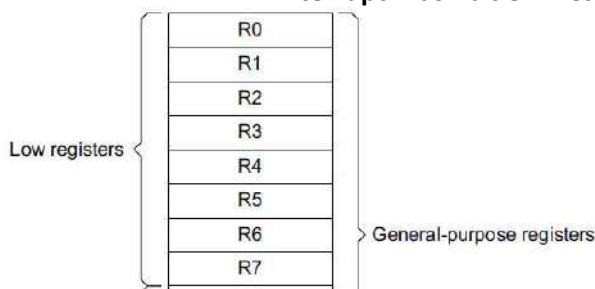
Address	Core peripheral
0xE000E008-0xE000E00F	SyStem Control Block
0xE000E010-0xE000E01F	System timer
0xE000E100-0xE000E4EF	Nested Vectored Interrupt Controller
0xE000ED00-0xE000ED3F	System Control Block
0xE000ED90-0xE000ED93	MPU Type Register
0xE000ED90-0xE000EDB8	Memory Protection Unit
0xE000EF00-0xE000EF03	Nested Vectored Interrupt Controller
0xE000EF30-0xE000EF44	Floating Point Unit

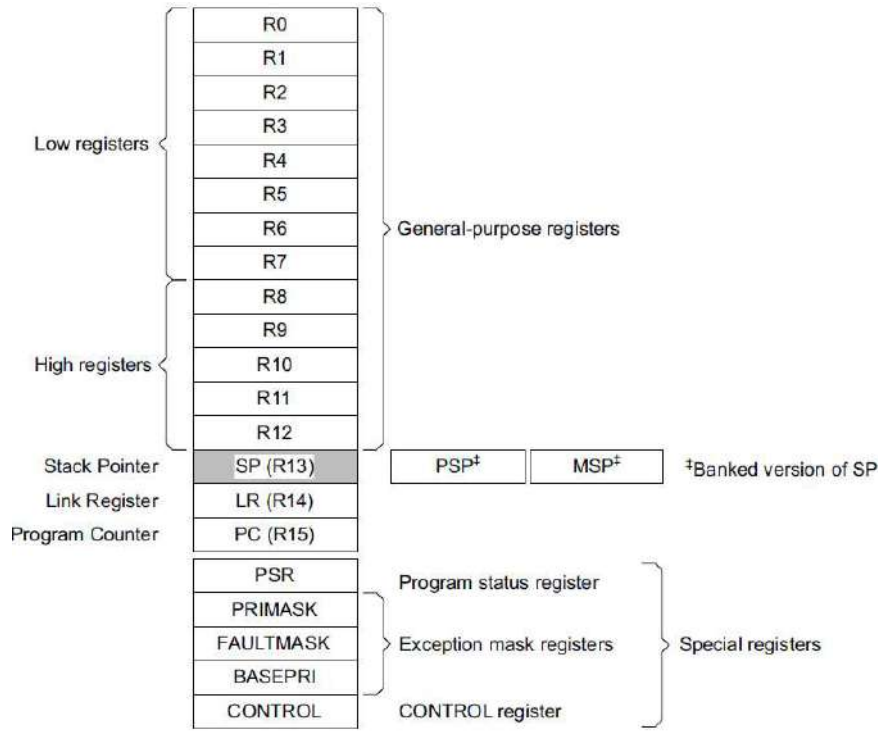
- **NVIC (Nested Vectored Interrupt Controller)**

- NVIC, düşük gecikmeli kesme yönetimini sağlar. Tüm kesmelerin önceliklendirilmesini ve hızlı şekilde işlenmesini mümkün kılar. Bu sayede gerçek zamanlı uygulamalarda hızlı tepki süresi elde edilir.
- **SCB (System Control Block)**
 - SCB, işlemciye ait sistem kontrolü ve yapılandırma işlemleri için programcı arayüzünü sunar. Exception yönetimi, hata raporlama ve bazı sistemsel ayarlar SCB üzerinden yapılır.
- **SysTick Timer**
 - SysTick, 24-bit geri sayım temelli bir sistem sayacıdır. Gerçek Zamanlı İşletim Sistemlerinde (RTOS) görev zamanlayıcısı olarak veya basit bir zaman ölçer olarak kullanılabilir.
- **MPU (Memory Protection Unit)**
 - MPU, bellek bölgelerinin erişim haklarını ve özelliklerini tanımlayarak sistemin güvenliğini ve güvenilirliğini artırır. Cortex çekirdeğinde sekiz farklı bellek bölgesi tanımlanabilir.
- **FPU (Floating-Point Unit)**
 - FPU, IEEE754 standardına uygun 32-bit tek duyarlıklı kayan nokta işlemlerini donanımsal olarak gerçekleştirir. Bu sayede sinyal işleme, kontrol algoritmaları ve matematiksel yoğunluğu yüksek uygulamalarda performans artışı sağlar.

Programmers Model

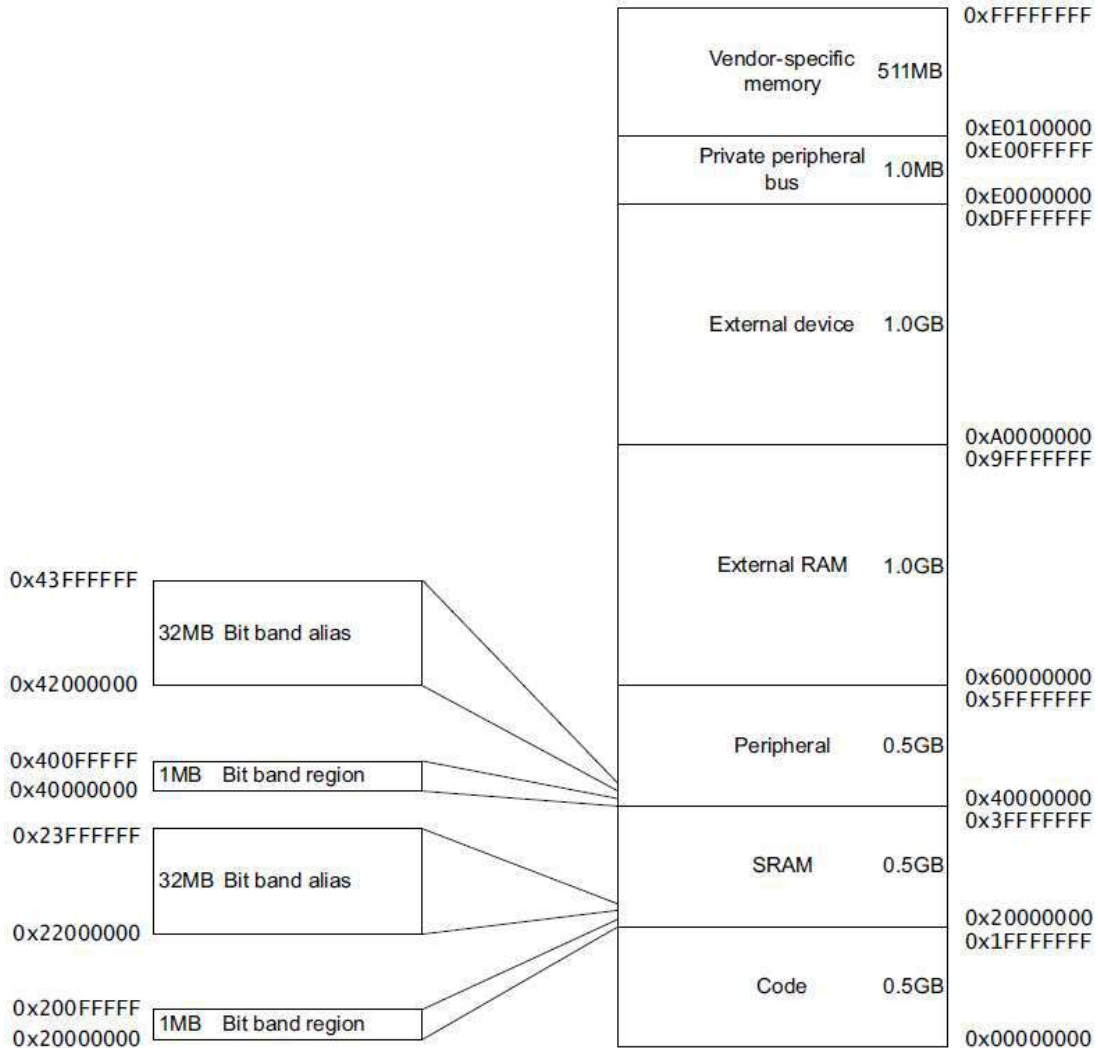
- STM32 mikrodenetleyicilerinde ARM Cortex-M mimarisi üzerinde çalışan **core registerları** bulunmaktadır. CPU core register'ları, işlemcinin çalışması için **en kritik** donanım alanıdır.
- Kodun çalışacağı adres (PC), stack yapısı (SP), fonksiyon dönüşleri (LR), işlem sonuç bayrakları (xPSR) ve genel veri saklama alanları (R0–R12) bu register'larda tutulur. Debugger aracılığıyla bu değerler görüntülenerek programın kontrol akışı ve bellek kullanımı hakkında bilgi edinilebilir.
 - **R0 – R12 (General-purpose registers)**, işlemci bunları geçici değer tutmak, aritmetik/lojik işlemler yapmak, fonksiyon parametrelerini taşımak gibi işler için kullanır. Derleyici de bunları yoğun kullanır.
 - 0xFFFFFFFF → muhtemelen debugger reset sonrası init olmamış (RAM içerikleri de belirsizdir).
 - **R13 (SP - Stack Pointer)**, Bellekteki **stack** (yığın) bölgesinin tepe adresini tutar. **MSP (Main Stack Pointer)** ve **PSP (Process Stack Pointer)** olmak üzere ikiye ayrılır. Sıfırlama sırasında işlemci MSP'ye 0x00000000 adresindeki değeri yükler.
 - 0x20000660 → Stack RAM içinde 0x20000660 adresinden aşağı doğru büyüyor. STM32 datasheet'inde başlangıç stack adresi vector table'ın ilk kelimesidir (bootloader ilk iş olarak SP'yi oraya yükler).
 - **R14 (LR - Link Register)**, fonksiyonlardan **dönüş adresini** tutar. Fonksiyon bittiğinde program LR'deki adrese geri döner. Sıfırlama sırasında işlemci LR değerini 0xFFFFFFFF olarak ayarlar.
 - 0xFFFFFFFF görünüyor → Geçerli bir return adresi yok gibi. Reset sonrası normal.
 - **R15 (PC - Program Counter)**, işlemcinin **bir sonraki çalıştıracağı komutun adresini** tutar. Kod yürütmenin akışını belirler. Atlamalar (branch, jump) bu register'ın değerini değiştirir. Sıfırlandığında, işlemci PC'ye 0x00000004 adresinde bulunan sıfırlama vektörünün değerini yükler.
 - 0x0800015C → Flash bellekteki (0x0800xxxx STM32'de Flash'tır) adresteki komutu çalıştırıyor. Yani bootloader kodun buradan devam ediyor.
 - **xPSR (Program Status Register)**
 - İşlemcinin **durumunu ve bayraklarını** tutar.
 - İçerisinde şunlar bulunur:
 - **N (Negative)**: Sonucun negatif olduğunu gösterir.
 - **Z (Zero)**: Sonucun sıfır olduğunu gösterir.
 - **C (Carry)**: Taşma/borç durumunu gösterir.
 - **V (Overflow)**: İşaretli taşma durumunu gösterir.
 - **Interrupt Mask bitleri**: Kesme izinlerini kontrol eder.





Memory Model

- İşlemcinin, **maksimum 4 GB adreslenebilir bellek** sağlayan sabit bir varsayılan bellek haritası vardır.



Exception Model

- Cortex-M çekirdeğinde tüm exception'lar **öncelik sırasına** göre numaralandırılır. Negatif numaralar sistem exception'larını, sıfır ve üzeri numaralar ise harici kesmeleri (IRQ) temsil eder.
 - Reset**, en yüksek önceliğe sahiptir. Sistem açıldığında veya reset durumunda çalışır.

- **NMI (Non-Maskable Interrupt)**, maskelenemeyen, çok kritik olaylar için ayrılmış kesme.
- **Fault Exceptions**, hatalar için tetiklenen exception'lar (HardFault, MemManage, BusFault, UsageFault)
 - **HardFault**, daha düşük seviyeli fault'ların ele alınamadığı durumlarda devreye girer.
 - **MemManage**, bellek koruması ihlali olduğunda tetiklenir. MPU (Memory Protection Unit) ile ilişkilidir.
 - **BusFault**, veri yolundaki hatalarda tetiklenir
 - **UsageFault**, geçersiz komut, yanlış hizalanmış erişim gibi CPU kullanım hatalarında tetiklenir
- **System Exceptions** → Çekirdeğe özgü özel kesmeler (ör. SVCcall, SysTick, PendSV)
 - SVCcall (Supervisor Call), genellikle işletim sistemi çağrılarında kullanılır.
 - PendSV (Ertelenmiş kesme), özellikle RTOS'larda görev değiştirme için kullanılır.
 - SysTick, genellikle RTOS tick üretiminde veya zamanlama amaçlı kullanılır.
- **External Interrupts (IRQs)** → Mikrodenetleyiciye bağlı çevre birimlerden gelen normal kesmeler.
- Cortex-M mimarisinde **exception'lar** farklı gruplara ayrılır ve her biri kendi handler fonksiyonuyla karşılanır.
 - **Interrupt Service Routines (ISRs)**: IRQ kesmeleri ISR'ler tarafından işlenen exception'lardır.
 - **Fault Handlers**: HardFault, MemManage Fault, UsageFault ve BusFault, fault handler tarafından işlenen fault exceptions.
 - **System Handlers**: NMI, SVCcall, PendSV, SysTick ve hata istisnalarının tamamı system exceptions ve system handler tarafından işlenir.

Exception number ^a	IRQ number ^a	Exception type	Priority	Vector address or offset ^b	Activation
1	-	Reset	-3, the highest	0x00000004	Asynchronous
2	-14	NMI	-2	0x00000008	Asynchronous
3	-13	HardFault	-1	0x0000000C	-
4	-12	MemManage	Configurable ^c	0x00000010	Synchronous
5	-11	BusFault	Configurable ^c	0x00000014	Synchronous when precise, asynchronous when imprecise
6	-10	UsageFault	Configurable ^c	0x00000018	Synchronous
7-10	-	Reserved	-	-	-
11	-5	SVCcall	Configurable ^c	0x0000002C	Synchronous
12-13	-	Reserved	-	-	-
14	-2	PendSV	Configurable ^c	0x00000038	Asynchronous
15	-1	SysTick	Configurable ^c	0x0000003C	Asynchronous
16	0	Interrupt (IRQ)	Configurable ^d	0x00000040 ^e	Asynchronous

- Vektör tablosu, **stack pointer'ın başlangıç değerini** ve tüm exception handlers için **başlangıç adreslerini** (exception vectors olarak da adlandırılır) içerir.
- Sistem resetlendiğinde, vektör tablosu varsayılan olarak **0x00000000** adresinde bulunur. Ancak yetkili yazılım, **VTOR** kaydını kullanarak bu tabloyu başka bir bellek bölgesine taşıyabilir. Bu taşıma, tablonun başlangıç adresini **0x00000080 ile 0x3FFFFFF80** aralığında herhangi bir konuma ayarlamayı mümkün kılar.

Exception number	IRQ number	Offset	Vector
16+n	n	0x0040+4n	IRQn
18	2	0x004C	IRQ2
17	1	0x0048	IRQ1
16	0	0x0044	IRQ0
15	-1	0x003C	Systick
14	-2	0x0038	PendSV
13			Reserved
12			Reserved for Debug
11			SVCcall

Exception number	IRQ number	Offset	Vector
16+n	n	0x0040+4n	IRQn
·	·	·	·
·	·	·	·
·	·	·	·
18	2	0x004C	IRQ2
17	1	0x0048	IRQ1
16	0	0x0044	IRQ0
15	-1	0x0040	Systick
14	-2	0x003C	PendSV
13		0x0038	Reserved
12			Reserved for Debug
11	-5	0x002C	SVCall
10			Reserved
9			
8			
7			
6	-10	0x0018	Usage fault
5	-11	0x0014	Bus fault
4	-12	0x0010	Memory management fault
3	-13	0x000C	Hard fault
2	-14	0x0008	NMI
1		0x0004	Reset
		0x0000	Initial SP value

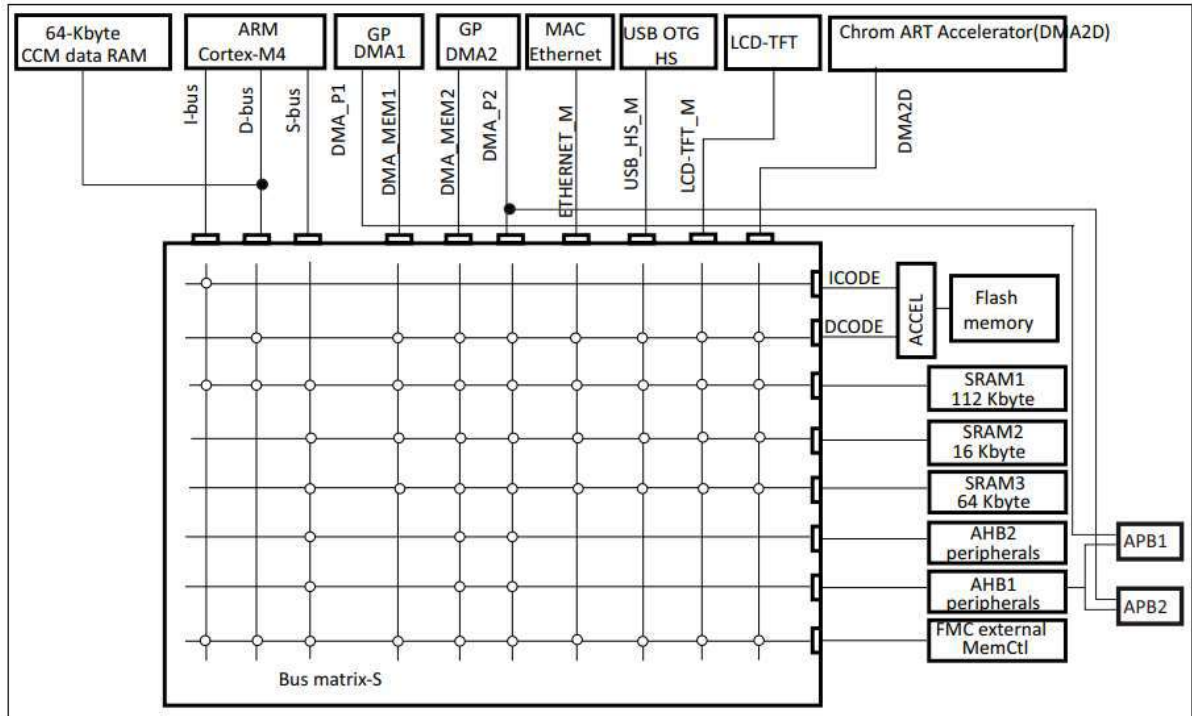
Bellek ve Veri Yolu Mimarisi

- Bellek mimarisi, sistemin bellek kaynaklarının nasıl yapılandırıldığını ve yönetildiğini belirler.
 - ROM (Read-Only Memory)**
 - Kalıcı veriler ve **program kodu** saklanır.
 - Mikrodenetleyicinin yeniden başlatılmasında bile veriler silinmez.
 - Flash Memory**
 - Program ve veri saklama için kullanılan yeniden programlanabilir ROM türüdür.
 - Genellikle firmware ve yazılım güncellemeleri burada saklanır.
 - SRAM (Static RAM)**
 - Geçici veri depolama için kullanılır.
 - Hızlı erişim süresine sahiptir ve genellikle çalışma sırasında kullanılan veri ve değişkenler burada saklanır.
 - EEPROM (Electrically Erasable Programmable ROM)**
 - Küçük veri parçalarını kalıcı olarak saklamak için kullanılır.
 - Elektriksel olarak silinebilir ve yeniden programlanabilir.
 - CCM (Core Coupled Memory)**
 - ARM Cortex-M4 işlemcisine yakın, düşük gecikmeli bellek alanıdır.
 - Yüksek performans gerektiren işlemler için kullanılır.
- Veri yolu mimarisi, farklı bileşenlerin birbirleriyle nasıl iletişim kurduğunu ve veri transferi yaptığını belirler.
 - System Bus (Sistem Veri Yolu)**
 - CPU, bellek ve çevresel birimler arasındaki ana iletişim hattıdır.
 - ARM Cortex-M4 gibi işlemciler genellikle AHB (Advanced High-performance Bus) veya APB (Advanced Peripheral Bus) kullanır.
 - AHB (Advanced High Performance Bus)**
 - Yüksek hızlı veri transferi sağlar.
 - Genellikle CPU, bellek ve yüksek hızlı çevresel birimler arasında kullanılır.
 - APB (Advanced Peripheral Bus)**
 - Daha düşük hız gerektiren çevresel birimler için kullanılır.
 - Daha az karmaşıktır ve daha az güç tüketir.
 - DMA (Direct Memory Access)**

- CPU müdahalesi olmadan veri transferi yapar.
- Bellekler ve çevresel birimler arasında hızlı veri transferi sağlar.
- **Bus Matrix:**
 - Birden fazla master ve slave bileşenin aynı anda veri transferi yapmasını sağlar.
 - STM32F407 şemasında görüldüğü gibi, bus matrix-S veri yollarını birbirine bağlar ve verimli veri transferi sağlar.
 - **I-bus, D-bus ve S-bus** veri yolları ile bellek ve çevresel birimlere bağlanır.
 - I-bus **talimatlar** için, D-bus **veriler** için ve S-bus **sistem veri yolu** için kullanılır.

Sistem Mimarisi

- STM32F407 mikrodenetleyicisinin sistem mimarisi, 32-bit çok katmanlı AHB veri yolu matrisi üzerine kuruludur. Bu matriste sekiz master ve yedi slave bileşen bulunur:
 - Master bileşenler, veri yolu üzerinde kontrol yetkisine sahiptir ve veri transferini başlatabilir. İşlemci veya DMA gibi master bileşenler, veri yolu üzerinden slave bileşenlere erişim talimatları göndererek veri okuyabilir veya yazabilir.
 - Cortex®-M4 with FPU core I-bus, D-bus and S-bus
 - DMA1 memory bus
 - DMA2 peripheral bus
 - Ethernet DMA bus
 - USB OTG HS DMA bus
 - Slave bileşenler, genellikle veri depolama veya belirli bir işlevi yerine getirme rolüne sahiptir. Bunlar, kendi başlarına aktif olarak veri yolu üzerinde veri transferi başlatabilmezler. Bu yüzden veri alışverişi yapabilmek için bir master bileşenin aktivasyonuna ve yönlendirmesine ihtiyaç duyarlar.
 - Internal Flash memory ICode bus
 - Internal Flash memory DCode bus
 - Main internal SRAM1 (112 KB)
 - Auxiliary internal SRAM2 (16 KB)
 - AHB1 peripherals including AHB to APB bridges and APB peripherals
 - AHB2 peripherals
 - FSMC

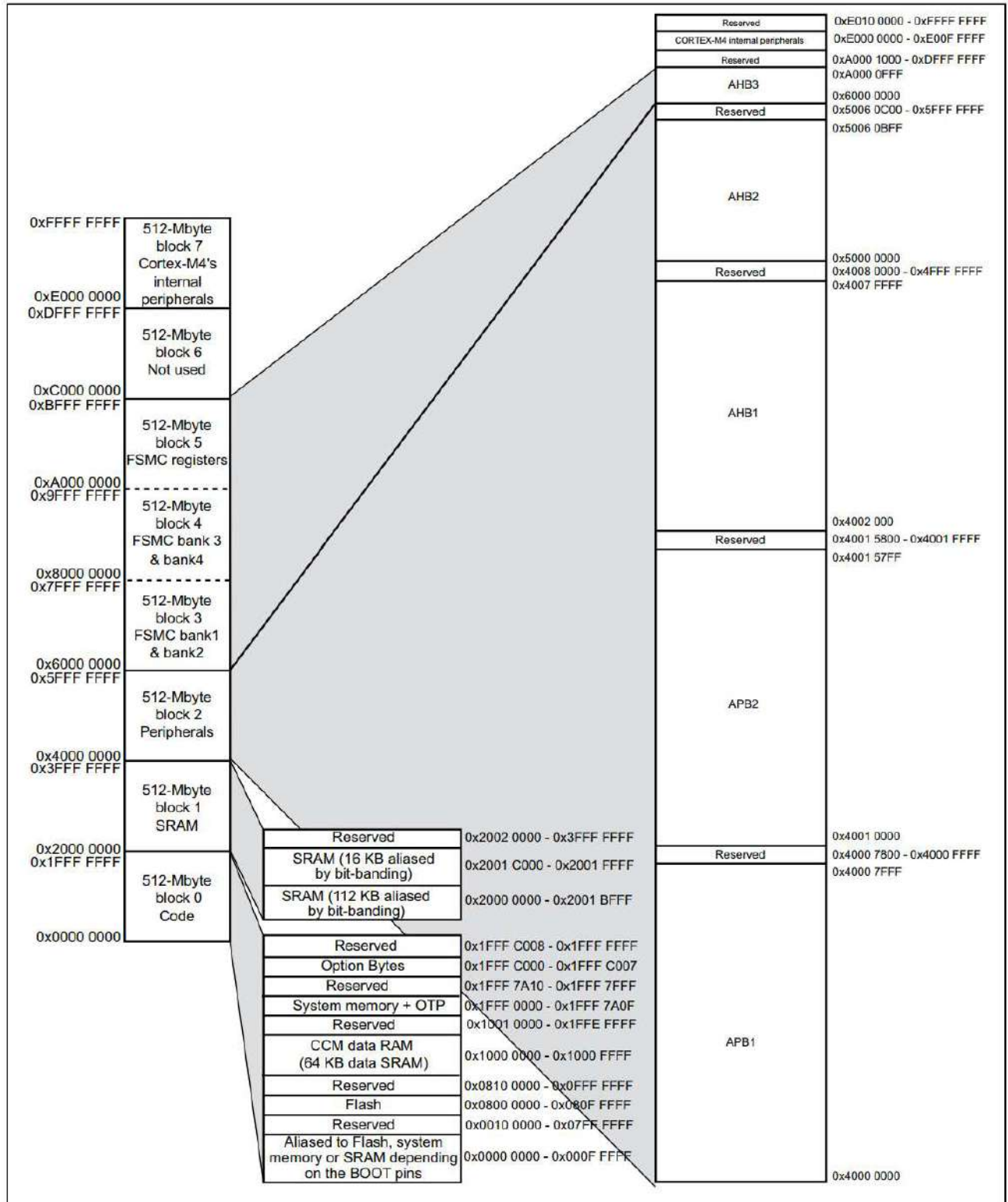


Bellek Organizasyonu

- Program belleği, veri belleği, kayıtlar ve I/O portları, aynı doğrusal 4 GB adres alanı içinde düzenlenmiştir.
- İşlemcimiz 32 bit veri yoluna sahiptir. Bu, 2^{32} işlemiyle hesaplanan 4.294.967.296 (4 GB) adresleyebilme kapasitesine sahip olduğu anlamına gelir. Bu sayıyı ondalık (decimal) sistemden onaltılık (hexadecimal) sisteme çevirdiğimizde **0x100000000** sonucunu elde ederiz. Buradan 1 çıkardığımızda ise **0xFFFFFFF** değerine ulaşırız. Adres uzayı **0x00000000** adresinden başlar ve **0xFFFFFFF** adresine kadar devam eder. Unutulmamalıdır ki, hex formatında her bir rakam 4 bitlik bir değere karşılık gelir.

- Bellekteki baytlar **little endian** formatında kodlanmıştır. En düşük numaralı bayt, kelimenin **en az anlamlı baytı** olarak kabul edilir; en yüksek numaralı bayt ise kelimenin **en anlamlı baytı**dır.
- Adreslenebilir bellek alanı, her biri 512 MB olan 8 ana bloğa bölünmüştür.
On-chip bellekler ve çevresel birimlere tahsis edilmemiş tüm bellek alanları **Reserved** olarak kabul edilir.

Bellek Haritası



Boundary address	Peripheral	Bus
0xE000 0000 - 0xE00F FFFF	Cortex-M4 internal peripherals	Cortex M4
0xA000 0000 - 0xA000 0FFF	FSMC control register	AHB3
0x5006 0800 - 0x5006 0BFF	RNG	
0x5006 0400 - 0x5006 07FF	HASH	

0x5006 0800 - 0x5006 0BFF	RNG	AHB2
0x5006 0400 - 0x5006 07FF	HASH	
0x5006 0000 - 0x5006 03FF	CRYP	
0x5005 0000 - 0x5005 03FF	DCMI	
0x5000 0000 - 0x5003 FFFF	USB OTG FS	
0x4004 0000 - 0x4007 FFFF	USB OTG HS	AHB1
0x4002 B000 - 0x4002 BBFF	DMA2D	
0x4002 8000 - 0x4002 93FF	ETHERNET MAC	
0x4002 6400 - 0x4002 67FF	DMA2	
0x4002 6000 - 0x4002 63FF	DMA1	
0x4002 4000 - 0x4002 4FFF	BKPSRAM	
0x4002 3C00 - 0x4002 3FFF	Flash interface register	
0x4002 3800 - 0x4002 3BFF	RCC	
0x4002 3000 - 0x4002 33FF	CRC	
0x4002 2800 - 0x4002 2BFF	GPIOK	
0x4002 2400 - 0x4002 27FF	GPIOK	
0x4002 2000 - 0x4002 23FF	GPIOK	
0x4002 1C00 - 0x4002 1FFF	GPIOK	
0x4002 1800 - 0x4002 1BFF	GPIOK	
0x4002 1400 - 0x4002 17FF	GPIOK	
0x4002 1000 - 0x4002 13FF	GPIOK	
0x4002 0C00 - 0x4002 0FFF	GPIOD	
0x4002 0800 - 0x4002 0BFF	GPIOC	
0x4002 0400 - 0x4002 07FF	GPIOB	
0x4002 0000 - 0x4002 03FF	GPIOA	
0x4001 6800 - 0x4001 6BFF	LCD-TFT	APB2
0x4001 5800 - 0x4001 5BFF	SAI1	APB2
0x4001 5400 - 0x4001 57FF	SPI6	
0x4001 5000 - 0x4001 53FF	SPI5	APB2
0x4001 4800 - 0x4001 4BFF	TIM11	
0x4001 4400 - 0x4001 47FF	TIM10	
0x4001 4000 - 0x4001 43FF	TIM9	
0x4001 3C00 - 0x4001 3FFF	EXTI	APB2
0x4001 3800 - 0x4001 3BFF	SYSCFG	
0x4001 3400 - 0x4001 37FF	SPI4	APB2
0x4001 3000 - 0x4001 33FF	SPI1	APB2
0x4001 2C00 - 0x4001 2FFF	SDIO	
0x4001 2000 - 0x4001 23FF	ADC1 - ADC2 - ADC3	
0x4001 1400 - 0x4001 17FF	USART6	
0x4001 1000 - 0x4001 13FF	USART1	
0x4001 0400 - 0x4001 07FF	TIM8	
0x4001 0000 - 0x4001 03FF	TIM1	
0x4000 7C00 - 0x4000 7FFF	UART8	APB1
0x4000 7800 - 0x4000 7BFF	UART7	
0x4000 7400 - 0x4000 77FF	DAC	
0x4000 7000 - 0x4000 73FF	PWR	
0x4000 6800 - 0x4000 6BFF	CAN2	
0x4000 6400 - 0x4000 67FF	CAN1	
0x4000 5C00 - 0x4000 5FFF	I2C3	

0x4000 6400 - 0x4000 67FF	CAN1	APB1
0x4000 5C00 - 0x4000 5FFF	I2C3	
0x4000 5800 - 0x4000 5BFF	I2C2	
0x4000 5400 - 0x4000 57FF	I2C1	
0x4000 5000 - 0x4000 53FF	UART5	
0x4000 4C00 - 0x4000 4FFF	UART4	
0x4000 4800 - 0x4000 4BFF	USART3	
0x4000 4400 - 0x4000 47FF	USART2	
0x4000 4000 - 0x4000 43FF	I2S3ext	
0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S3	
0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2	
0x4000 3400 - 0x4000 37FF	I2S2ext	
0x4000 3000 - 0x4000 33FF	IWDG	
0x4000 2C00 - 0x4000 2FFF	WWDG	
0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers	
0x4000 2000 - 0x4000 23FF	TIM14	
0x4000 1C00 - 0x4000 1FFF	TIM13	
0x4000 1800 - 0x4000 1BFF	TIM12	
0x4000 1400 - 0x4000 17FF	TIM7	
0x4000 1000 - 0x4000 13FF	TIM6	
0x4000 0C00 - 0x4000 0FFF	TIM5	
0x4000 0800 - 0x4000 0BFF	TIM4	
0x4000 0400 - 0x4000 07FF	TIM3	
0x4000 0000 - 0x4000 03FF	TIM2	

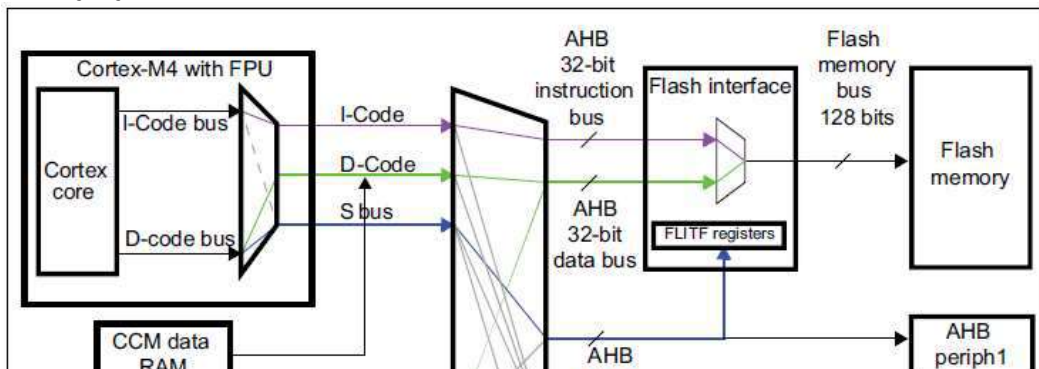
Boot Konfigurasyonu

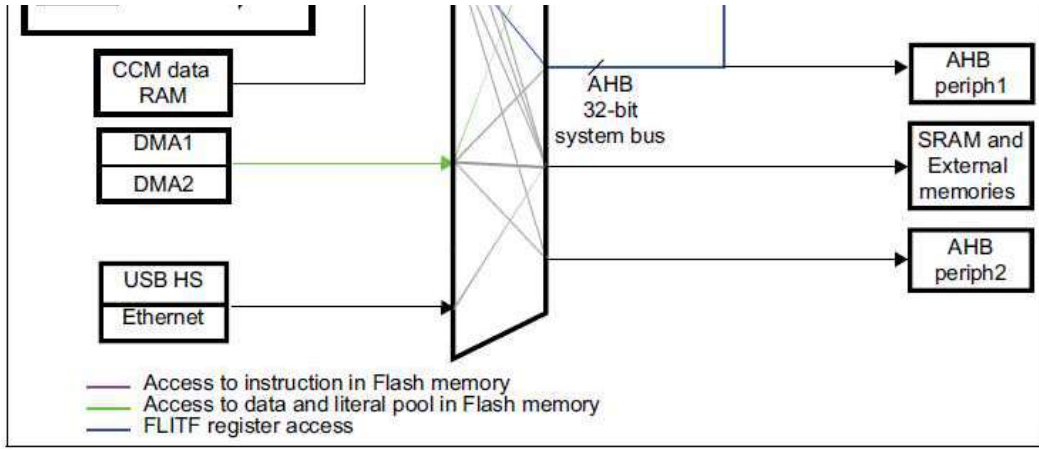
- STM32'de BOOT pinleri ile farklı bellek bölgeleri **0x0000 0000** adresine haritalanabilir. Aslında işlemcinin gördüğü 0x0000 0000 adresi **donanım tarafından seçilen bellek bölgesine yansıtılır**.
 - Main Flash (0x0800 0000)** → Normal uygulama başlatma
 - System Memory (0x1FFF xxxx)** → ST'nin dahili bootloader'ı
 - SRAM (0x2000 0000)** → RAM'den çalıştırma

Boot mode selection pins		Boot mode	Aliasing
BOOT1	BOOT0		
x	0	Main Flash memory	Main Flash memory is selected as the boot space
0	1	System memory	System memory is selected as the boot space
1	1	Embedded SRAM	Embedded SRAM is selected as the boot space

Flash Memory

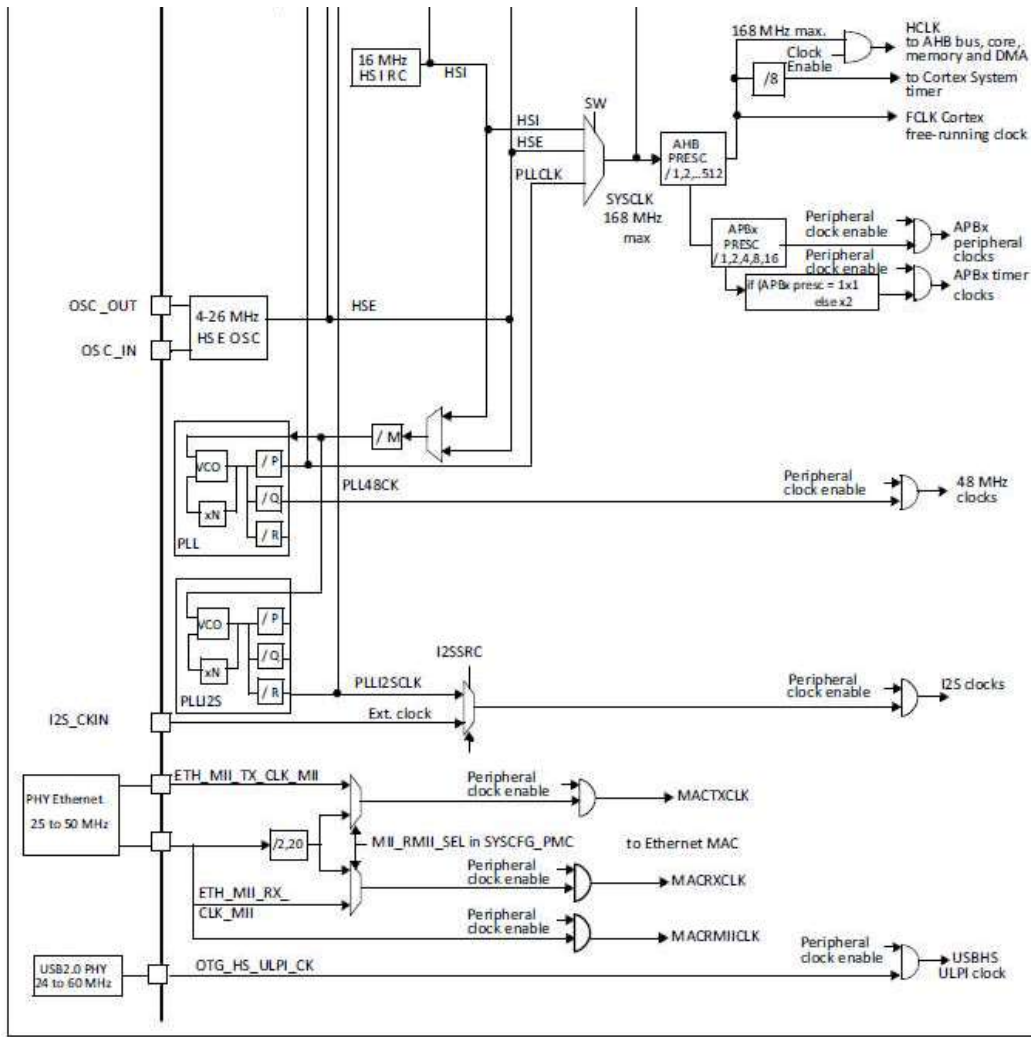
- Cortex-M çekirdeğinin aynı anda hem komut hem veri hem de sistem erişimlerini paralel olarak yapabilmesini sağlar. Flash interface 128-bit geniş erişimle Flash'tan bloklar alır, bunları cache'ler ve CPU'ya 32-bit olarak iletir. Böylece Flash'ın görece yavaş hızına rağmen işlemci yüksek performansta çalışır.



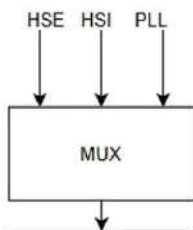


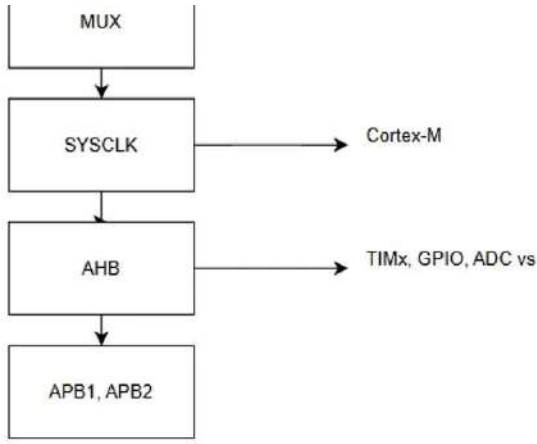
- Mikrodenetleyicinin hafıza organizasyonu aşağıda detaylı bir şekilde açıklanmaktadır.
 - **Flash Bellek**, uygulama kodumuzu saklamak ve programın verilerini okumak için kullanılır. Ayrıca vektör tablolarını saklamak için de kullanılır.
 - Boyut: 1 Mbyte
 - Başlangıç: 0x0800 0000
 - Bitiş: 0x080F FFFF
 - İçine yazılan veriler kalıcıdır. (Non Volatile)
 - **SRAM**, mikrodenetleyicinin rastgele erişim belleği sayısı 2'dir.
 - **SRAM1**, uygulamamızdaki global verileri ve statik değişkenlerimizi depolamak için kullanılır. Ayrıca **stack** ve **heap** amacıyla kullanılır. Bu bellekten kod çalıştırılabilir. Yani burada herhangi bir talimat sakladıysanız, işlemcinin bunu yürütebilmesi için SRAM'e geçebileceği anlamına gelir.
 - Boyut: 112 Kbyte
 - Başlangıç: 0x2000 0000
 - Bitiş: 0x2001 BFFF
 - İçine yazılan veriler kalıcı değildir (Volatile)
 - **SRAM2**, boyut olarak SRAM1'den daha küçüktür ve bellek haritasında SRAM1'den hemen sonra gelir. Uygulamalarımızda global verileri ve statik değişkenleri depolamak için kullanılır. Ayrıca heap ve stack oluşturabiliriz. Yine bu bellekten de kod çağrılabilir.
 - Boyut: 16 Kbyte
 - Başlangıç: 0x2001 0000
 - Bitiş: 0x2001 FFFF
 - İçine yazılan veriler kalıcı değildir. (Volatile)
 - **ROM** (Read Only Memory) belleğine sahiptir. ST dokümanlarında bu bellek **System Memory** olarak da adlandırılmaktadır. Read Only Memory olduğu için sadece okunabilir bir bellektir ve herhangi bir veri bu hafıza alanına yazılamaz. Bu bellek, ST tarafından oluşturulan bootloader'ın saklandığı bellek bölgesidir. Yani STM ürünlerinin önyükleyicileri ROM'da saklanır ve tek seferlik programlanabilir hafıza olarak adlandırılan veriye sahiptir
 - Boyut: 30 Kbyte
 - Başlangıç: 0x1FFF 0000
 - Bitiş: 0x1FFF 77FF
 - **OTP** belleği de yalnızca bir kez programlanabilir ve bu bilgi genellikle bir ürünün ürün numarası veya seri numarasını kullanıcı tarafından saklamak için kullanılır.
 - Boyut: 528 bayt
 - Başlangıç: 0x1FFF 7800
 - Bitiş: 0x1FFF 7A0F
 - **Option Bytes** Memory bulunmaktadır. Genel kullanım için değildir. Bu baytların flash belleğe erişimi kontrol eden bazı kaynakları vardır.
 - Boyut: 8 Kbyte
 - Başlangıç: 0x1FFF C000
 - Bitiş: 0x1FFF C00F

Block	Name	Block base addresses	Size
	Sector 0	0x0800 0000 - 0x0800 3FFF	16 Kbytes
	Sector 1	0x0800 4000 - 0x0800 7FFF	16 Kbytes



- AHB (Advanced High-performance Bus) Clock, STM32 mikrodenetleyicilerde saat sistemi, SYSCLK olarak adlandırılan sistem saatinden türetilen AHB ve APB veriyolları üzerinden çevre birimlere dağıtılır. SYSCLK, öncelikle AHB adı verilen yüksek hızlı veri yoluna aktarılır. RAM, Flash bellek, DMA ve GPIO gibi yüksek hız gereksinimi olan birimlerin saat kaynağını sağlar. Bu saat, prescaler kullanılarak gerektiğinde yavaşlatılabilir; örneğin 1, 2, 4, 8 gibi böölücülerle çalışabilir. AHB üzerinden türeyen iki önemli alt saat yolu ise APB1 ve APB2'dir.
 - APB1 (Advanced Peripheral Bus 1), APB1, daha düşük hızda çalışması gereken çevre birimlerine hizmet verir. Bu bus üzerinden USART2 ve USART3 gibi seri haberleşme birimleri, I2C arayüzleri, CAN modülü, DAC ve zamanlayıcılar (TIM2-TIM7 gibi) çalıştırılır. Genellikle bu veri yolu için tanımlanan maksimum frekans STM32F4 serisinde 42 MHz'tir.
 - APB2 (Advanced Peripheral Bus 2), APB2 veri yolu daha yüksek hızlara uygun çevre birimlerini besler. Bu bus üzerinden USART1, SPI1, ADC ve TIM1 gibi çevre birimleri saat sinyali alır. STM32F4 serisinde APB2'nin maksimum frekansı genellikle 84 MHz'e kadar çıkabilir. Bu yapı sayesinde farklı hız gereksinimleri olan çevre birimleri, sistem performansı ve enerji verimliliği gözetilerek uygun saat kaynaklarıyla çalıştırılabilir.
- STM32 mikrodenetleyicilerde clock yapılandırma süreci, öncelikle sistemin hangi ana saat kaynağını kullanacağına karar verilmesiyle başlar; bu kaynak dahili HSI veya harici HSE osilatörü olabilir. Seçilen bu saat kaynağı, ihtiyaç duyulan daha yüksek frekanslar için PLL (Phase-Locked Loop) devresi kullanılarak çarpılabilir. PLL kullanımı sayesinde sistem saati (SYSCLK) istenilen frekansa yükseltilmiş olur. Belirlenen SYSCLK daha sonra AHB veri yoluna aktarılır ve buradan AHB'ye bağlı çevre birimlerine saat sinyali sağlanır. AHB üzerinden türetilen APB1 ve APB2 veri yolları aracılığıyla ise düşük ve yüksek hızlı çevre birimlerine uygun saat sinyalleri dağıtılarak sistem genelinde senkronizasyon sağlanır.





- APB1 frekansı düşük tutulur çünkü düşük hızlı çevre birimleri buraya bağlıdır.
- RCC (Reset and Clock Control) register'ları üzerinden bu ayarlar yapılır.

Çevresel Saat Etkinleştirme Kayıtları

- Her bir çevresel birimin saati, ilgili register kayıtlarındaki etkinleştirme biti ile etkinleştirilebilir. Çevresel saat **aktif değilken**, çevresel birimin kayıtlarına **okuma** veya **yazma** erişimleri **desteklenmez**.
- **Etkinleştirme bitinin**, çevresel birim için **parazitsiz glitch-free** bir saat oluşturmak üzere bir senkronizasyon mekanizması vardır. Etkinleştirme biti set edildikten sonra, saatin aktif hale gelmesi için **2 saat döngüsü (clock cycle)** gecikme vardır. Bu durum, yazılım tarafından dikkate alınmalıdır.

Birim Yapısı

