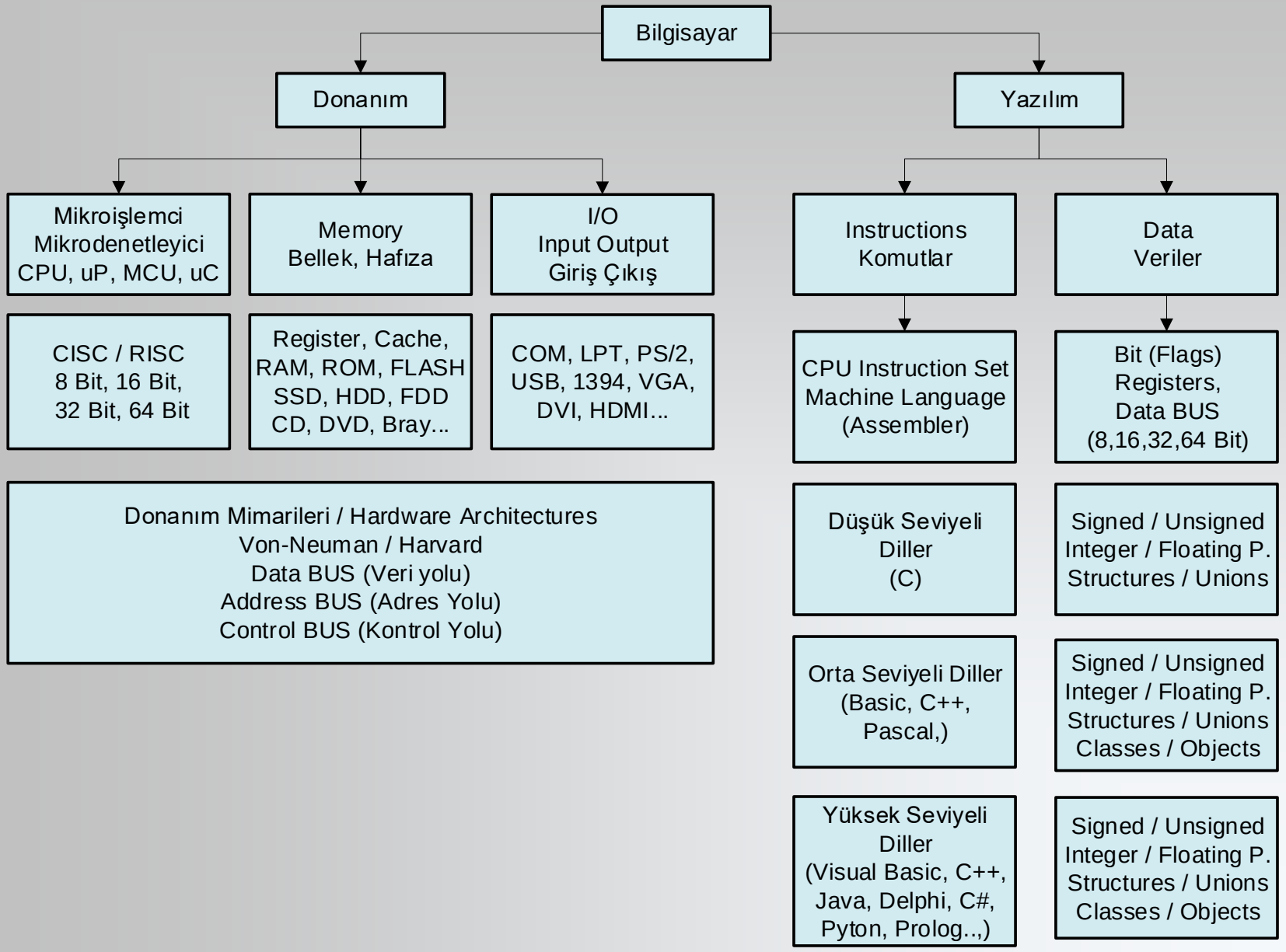
Veri Yapıları ve Algoritmalar

Giriş ve Temel Kavramlar

Dr. Öğr. Üyesi M. Ozan AKI

30.10.2020

Bilgisayar Yapısı



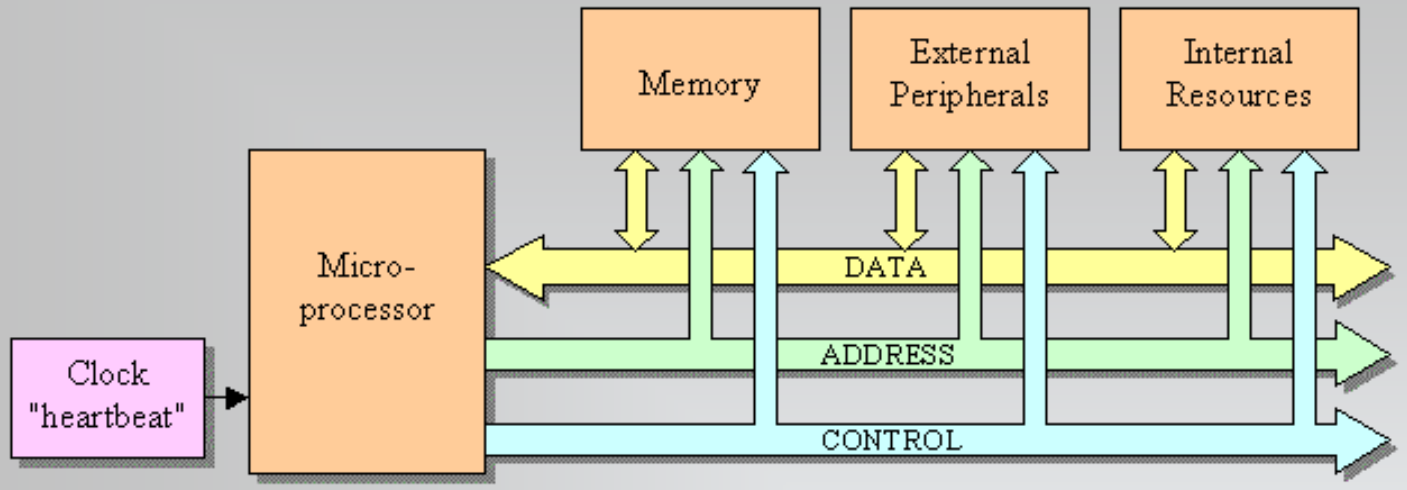
Donanım Mimarileri

Donanım Mimarileri: Mikroişlemci, Bellek ve Giriş/Çıkış birimlerinin arasındaki veri aktarımını sağlayan fiziksel yolların bağlantı tasarımları ve belleklerin organizasyonunu belirleyen tasarımlardır.

Günümüzde yaygın olarak kullanılan iki Mimari türü mevcuttur:

- **Von Neumann** Mimarisi (John Von Neumann 1945)
- **Harvard** Mimarisi (Harvard Mark I, Harvard Univ. 1944)

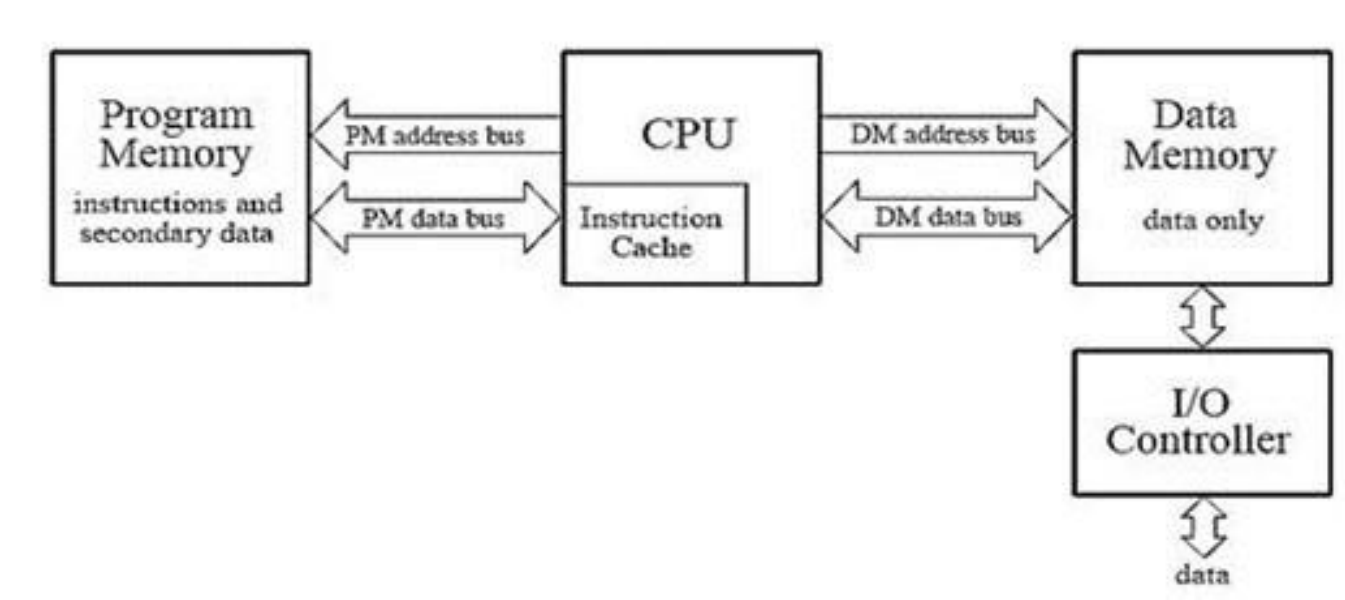
Von Neumann Mimarisi



Mikroişlemcinin çalışması için gerekli komutlar ve bu komutların işleyeceği veriler aynı fiziksel bellek üzerinde bulunurlar.

Bu veriler, «Adres Yolu» ile adreslenen bellek bölgesinden «Veri Yolu» aracılığı ile Mikroişlemci ile Bellek üniteleri arasında aktarılır.

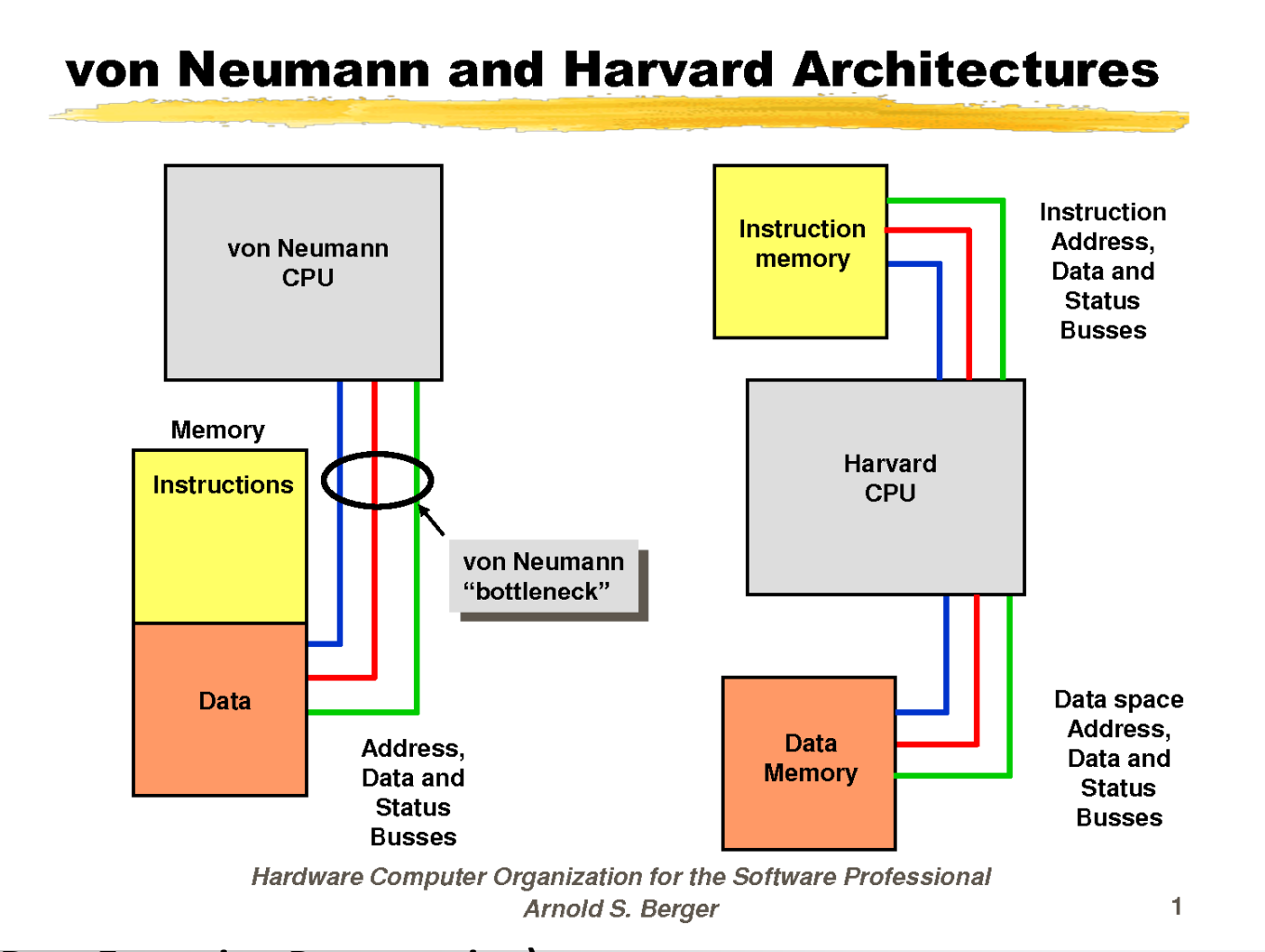
Harvard Mimarisi



Harvard Mimarisinde, Mikroişlemcinin çalıştıracağı komutlar (instructions) «Program Belleği»nde, bu komutların işleyeceği veriler ise «Veri Belleği»nde bulunur.

Bu bellekler fiziksek olarak ayrı bellekler olup, ayrı adres ve veri yollarından Mikroişlemciye bağlıdır. Veri belleğinde komut ya da program belleğinde veri bulunamaz.

Donanım Mimarileri



DEP (Data Execution Preventation)

Windows: Bilgisayarım -> Özellikler -> Gelişmiş Ayarlar -> Sistem özellikleri -> Advanced -> Performance -> Ayarlar -> Data Execution Prevention

Mikroişlemci (CPU, μP)

Temel lojik ve aritmetik işlemleri, bir dizi komutla sıralı bir şekilde gerçekleştirebilen sayısal tüm devrelerdir.

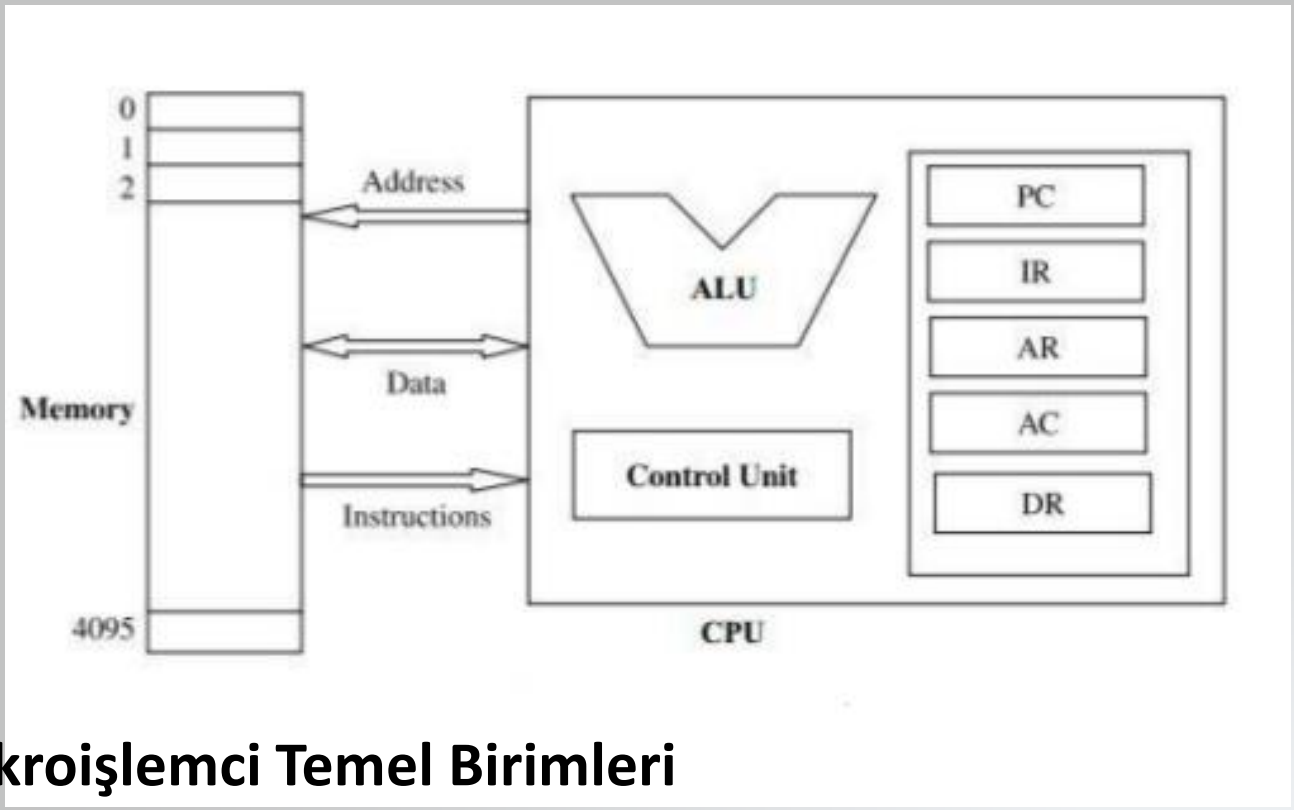
Çalışması gerekli komutları, komutların saklı olduğu bir bellekten okunması gerekir.

Ayrıca komutların işleyeceği verileri almak ve işlemlerin sonuçlarını saklamak için bir veri belleği gereklidir

Tüm işlemler, mikroişlemciye saat darbelerini (pulse) sağlayan osilatörün her bir çevriminde gerçekleşir.

Saat darbelerinin bir saniyedeki sayısı (Hz), işlemlerin ne kadar hızlı yapılacağını belirler.

Mikroişlemci Temel Birimleri



Mikroişlemci Temel Birimleri

- ALU (Arithmetic-Logic Unit) (Aritmetik-lojik Birim)
- Registers (Kaydediciler)
- Control Unit (Kontrol Birimi)

Mikroişlemci ile ilgili Kavramlar

- Reset Vector

İlk Enerji verildiğinde ya da Resetlendiğinde Çalıştırılacak komutların başlangıç adresidir. Genelde 0x0000 adresidir.

- Fetch & Execution

Komutların bellekten sırasıyla alınır işlenmesidir. Bu işlem içinde ayrıca işemci zamanı harcanır. (Pipeline?)

- Flags (PSW – Program Status Word)

Çalışma durumu ve komutların çalışma sonuçları ile ilgili bilgilerin bit olarak tutulduğu kaydedicidir.

- Instruction Pointer (IP) / Program Counter (PC)

Komutları sırasıyla çalışmasını sağlayan sayaçtır. Bu sayaç dallanma komutları ile değiştirilerek komut işleyişinin sırası değiştirilir. (Kaç bit olmalı?)

CISC / RISC Komut Setleri

CISC (Complex Instruction Set Computer) Örn: 8086/8088

AAA, AAD, AAM, AAS, ADC, ADD, AND, CALL, CBW, CLC, CLD, CLI, CMC, CMP, CMPSB, CMPSW, CWD, DAA, DAS, DEC, DIV, ESC, HLT, IDIV, IMUL, IN, INC, INT, INTO, IRET, JA, JAE, JB, JBE, JC, JCXZ, JE, JG, JGE, JL, JLE, JNA, JNAE, JNB, JNBE, JNC, JNE, JNG, JNGE, JNL, JNLE, JNO, JNP, JNS, JNZ, JO, JP, JPE, JPO, JS, JZ, JMP, LAHF, LDS, LEA, LES, LOCK, LODSB, LODSW, LOOP, MOV, MOVSB, MOVSW, MUL, NEG, NOP, NOT, OR, OUT, POP, POPF, PUSH, PUSHF, RCL, RCR, REP, REPE, REPNE, REPNZ, REPZ, RET, RETN, RETF, ROL, ROR, SAHF, SAL, SAR, SBB, SCASB, SCASW, SHL, SHR, STC, STD, STI, STOSB, STOSW, SUB, TEST, WAIT, XCHG, XLAT, XOR

RISC (Reduced Instruction Set Computer) Örn: PIC16F84A

ADDWF, ANDWF, CLRF, CLRW, COMF, DECF, DECFSZ, INCF, INCFSZ, IORWF, MOVF, MOVWF, NOP, RLF, RRF, SUBWF, SWAPF, XORWF, BCF, BSF, BTFSC, BTFSS, ADDLW, ANDLW, CALL, CLRWDT, GOTO, IORLW, MOVLW, RETFIE, RETLW, RETURN, SLEEP, SUBLW, XORLW

CISC vs RISC

A ve B Bellek adreslerindeki iki sayıyı karşılaştıralım ve büyük olan sayıyı C adresine yazalım

CISC Komut Seti

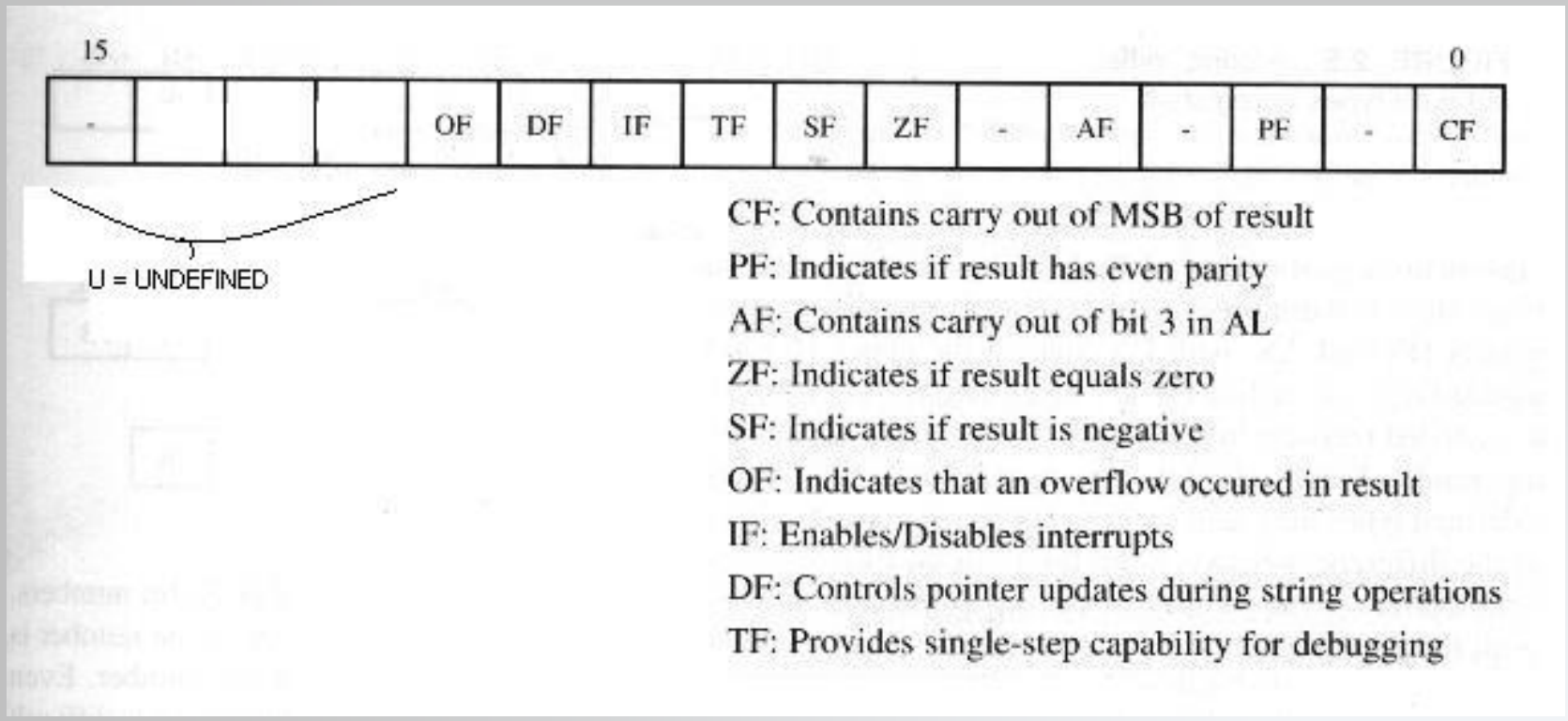
```
CMP A,B
JB X
MOV AX,[A]
JMP Z
X:
MOV AX,[B]
Z:
MOV [C],AX
```

RISC Komut Seti

```
MOVF B,W
SUBWF A,W
BTFSS STATUS,C
GOTO X
MOVF A,W
GOTO Z
X:
MOVF B,W
Z:
MOVWF C
```

Flags – Durum Bayrakları

8086 Program Status Word (PSW) Flag Register

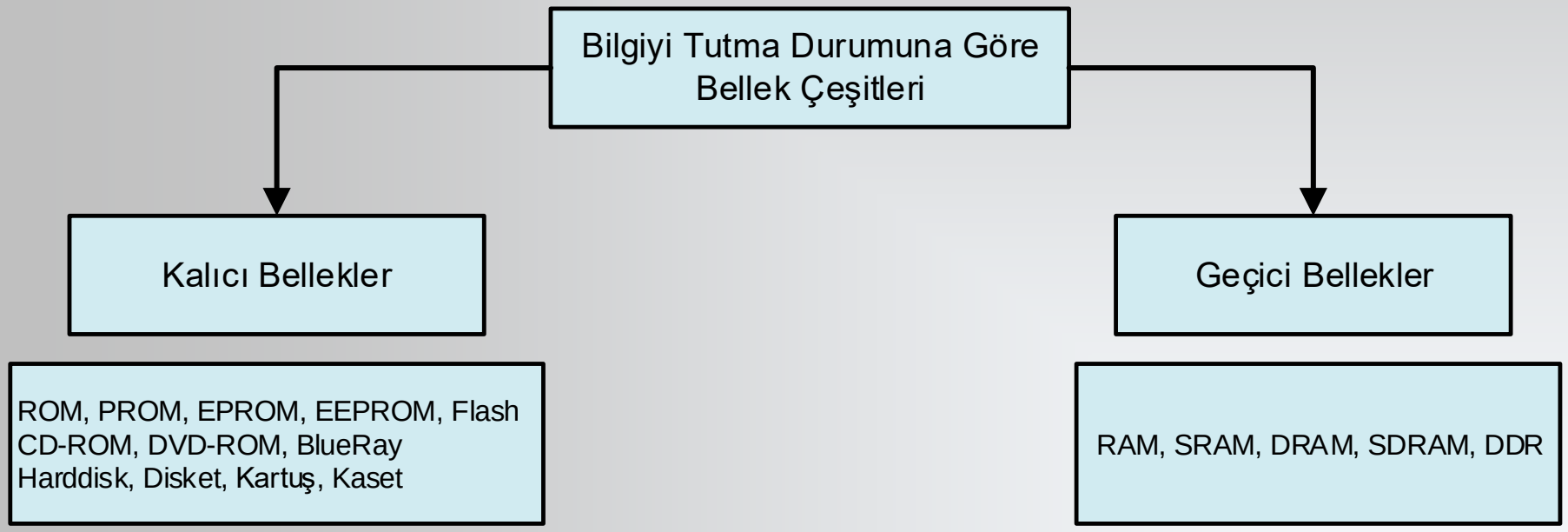


Bellek – Hafıza - Memory

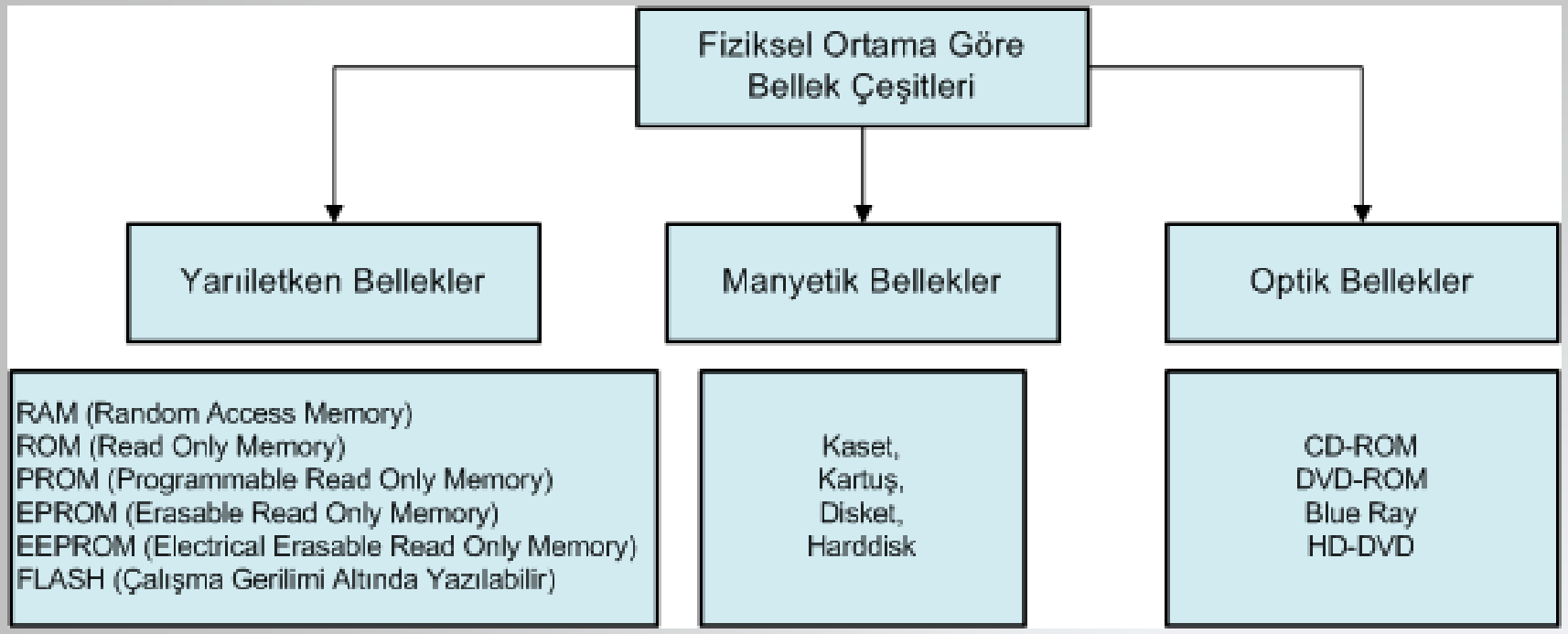
Bellek

Üzerinde bilgi saklanabilen tüm birimler «Bellek» olarak adlandırılır.

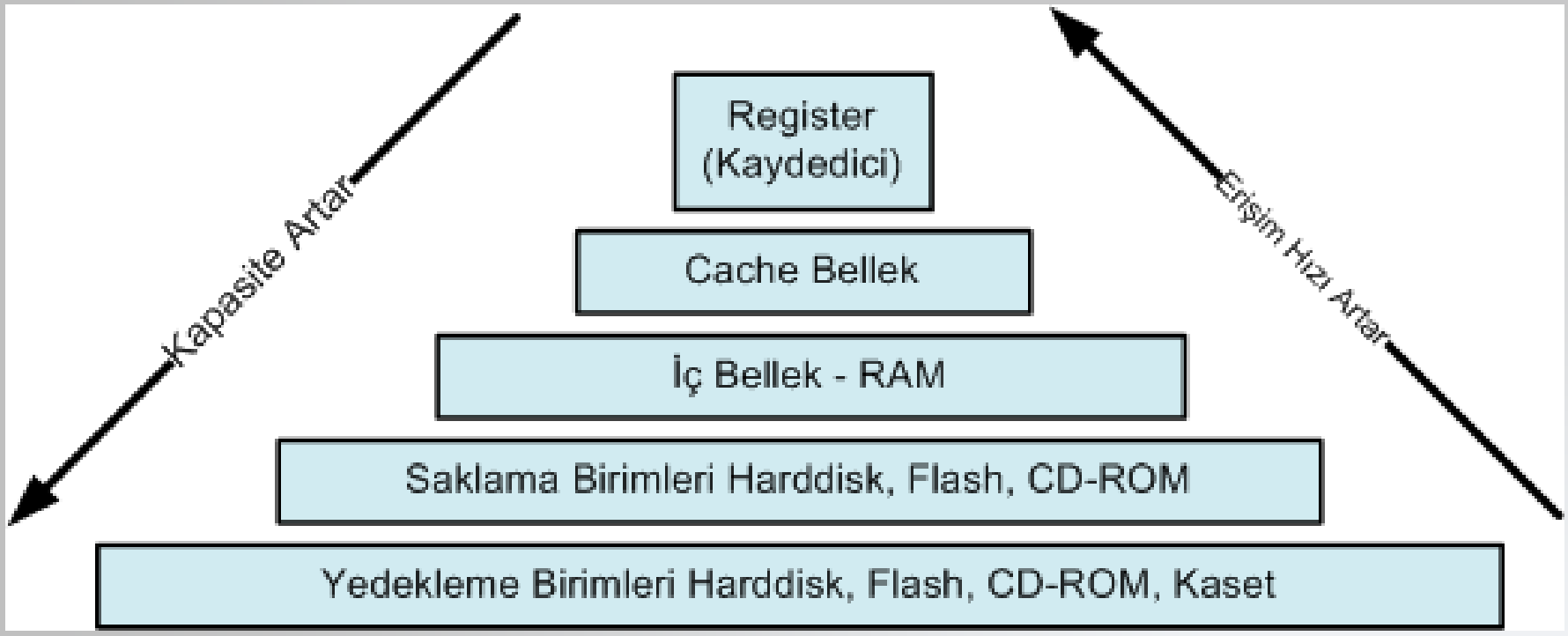
Bilgiyi Tutma Durumuna Göre Bellekler Çeşitleri



Fiziksel Ortamına Göre Bellek Çeşitleri



Erişim Durumuna Göre Bellek Çeşitleri



Analoji: Depo -> Toptancı -> Bakkal -> Buzdolabı -> Masadaki Tabak

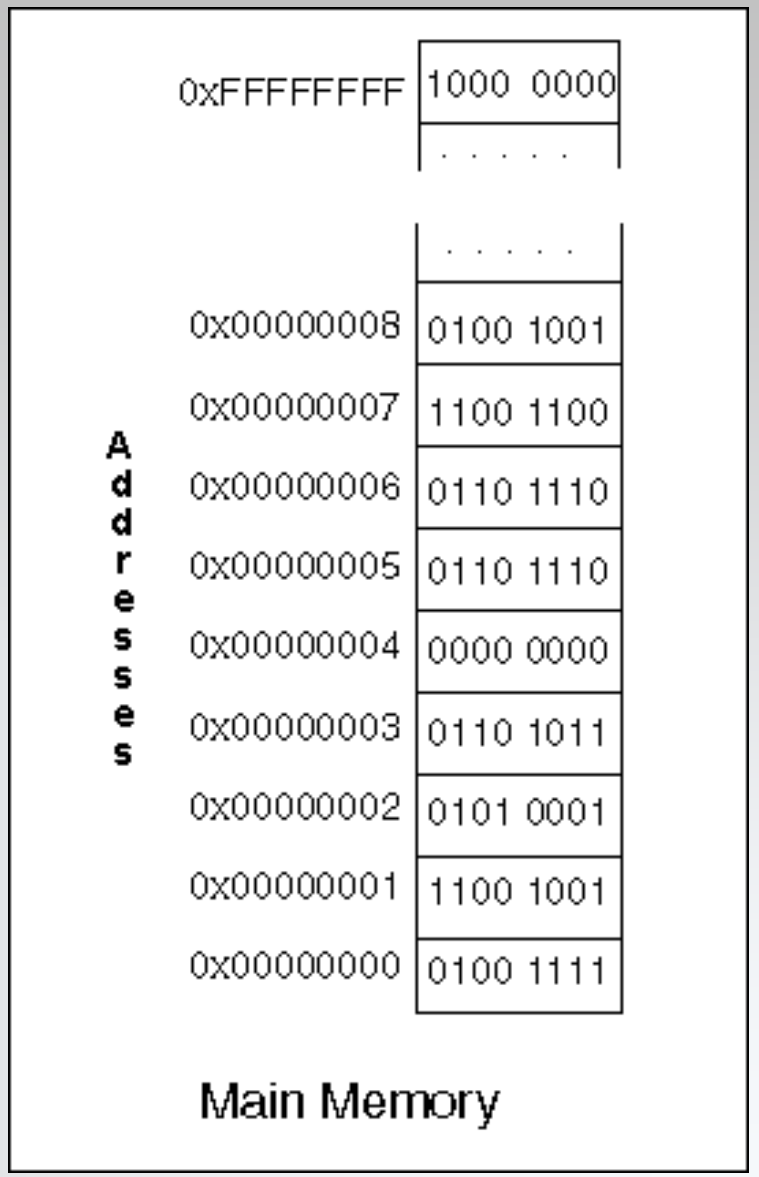
Belleğin Adreslenmesi

Register

Doğrudan kaydedici adı ile kullanılırlar, bellek adresleri söz konusu değildir. Kaydedicilerin bellek adresi yoktur, «Instruction Word» içerisinde kodlanmıştır.

Bellek

Bellek adresi ile kullanılırlar.
Her adres bir bellek hücresine işaret eder.
Her hücre, veriyolu bit sayısı kadar veri saklar



Yazılım & Program

Yazılım (Software)

Yazılım, bir işin gereklerini tümüyle karşılamak amacıyla tasarlanmış ve birçok alt bileşenden ve alt programdan oluşan bir sistemdir.

Program (Programm)

Program, bir algoritmanın belirli bir programlama dilinde kodlanmasını kapsar. Sadece spesifik bir amaca yönelik olup, bir yazılımın parçası olabilirler.

Kaynak Kod & Derleyici & Bağlayıcı

Makine Dili (Machine Language)

Komut setlerini oluşturan ikili sayılardır. Doğrudan mikroişlemciye girilebilir.

Birleştirici Dil (Assembly)

İnsan tarafından okunabilir ve anlaşılabilir OPCODE lar makine diline çevrilir.

Kaynak Kod (Source Code)

Yüksek seviyeli dillerin kaynak kodlarını makine diline çevirerek object file olarak kaydeder.

Compiler (Derleyici)

Yüksek seviyeli dillerin kaynak kodlarını makine diline çevirerek object file olarak kaydeder.

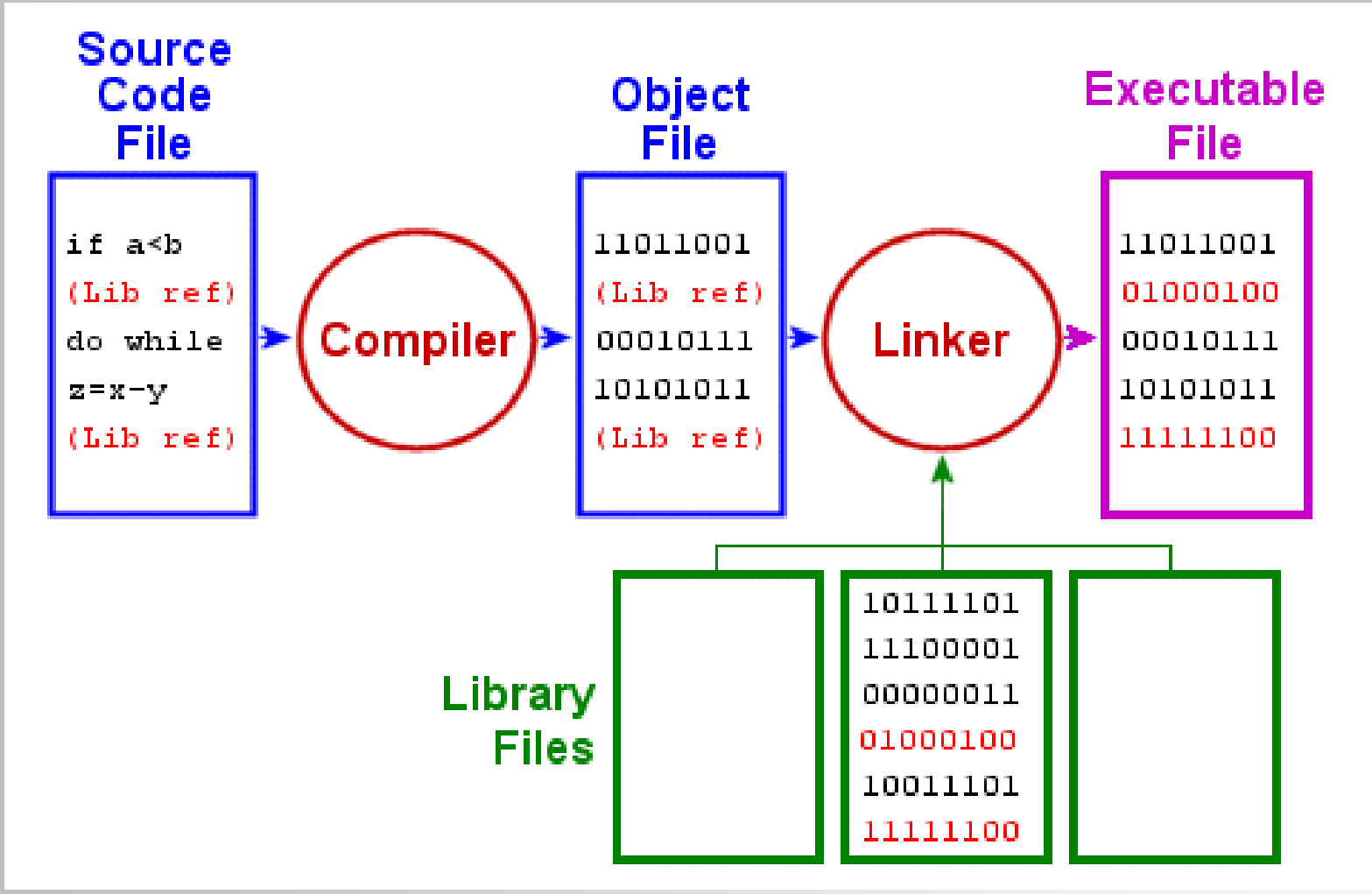
Cross Compiler (Çapraz Derleyici)

Yüksek seviyeli dillerin kaynak kodlarını üzerinde çalıştığı makineden farklı bir işlemcinin makine diline çevirerek object file olarak kaydeder.

Linker (Bağlayıcı)

Daha önce derlenmiş olan object dosyaları birleştirerek çalıştırılabilir (COM, EXE, DLL, BIN, HEX) dosyayı elde eder ve kaydeder.

Kaynak Kod & Derleyici & Bağlayıcı



Makine Kodu & Assembly Dili

Makine dili, mikroişlemcinin tasarımından doğan komut seti (instruction set) komutlarıyla oluşturulan programlardır. Makine dili komutları doğrudan ikilik taban (binary) ya da onaltılık taban (hexadecimal) şeklinde gösterilebilen sayıt formatındadır. Bu nedenle makine dilinde bir programın kodlanması oldukça zor ve karmaşıktır.

Sayısal formdaki her bir makine kodunun, insanlar için daha anlamlı olan ve komutun icrasını hatırlatan kısaltma şeklindeki kelimelere dönüştürülmesiyle Assembly Dili oluşturulmuştur.

Makine ve Assembly dilleri, doğrudan mikroişlemciye özgü makine dilleri olup, ek bir derleyiciye ihtiyaç duymazlar. Derleme işleminde sadece kelime olan komutlar ikilik sayılara dönüştürülür. Bu diller, mikroişlemciye sıkı sıkıya bağlıdır.

Programlama Dilleri

Belirli bir Derleyici ya da Yorumlayıcıya bağlı olarak, kendine özgü yazım kuralları çerçevesinde makine dilinden soyutlanmış ve konuşma diline yakın formatta komutlar yazılabilmesini sağlayan kurallar bütünüdür.

Bir dilin derleyicileri, birçok mikroişlemci türüne göre makine dili üretebilecek şekilde geliştirilmiş olabilir. Kodun hangi mikroişlemcide çalışması isteniyorsa, kod o mikroişlemci için derlenmelidir.

Böylece, aynı dilin farklı mikroişlemciler için yazılmış derleyicileri ile, kaynak kodda herhangi değişiklik yapmadan tekrar derleme yoluyla farklı platformlar için çalıştırılabilir kodlar elde etmek mümkün olur.

Programlama Dilli Yaklaşımları

Programlama Dillerinde Dört yaklaşım vardır;

- Prosedürel Yönelimli (Fortran, Pascal, ..)
- Nesne Yönelimli (C++, C#, Java, ..)
- Mantık Yönelimli (Prolog, ..)
- Görev Yönelimli (Verilog, SQL, Mathematica, ..)

Programlama Dilleri, **Veri yapıları** ve **Algoritmalardan** oluşur.

İşletim Sistemleri (Operating Systems - OS)

Donanım, uygulama yazılımları ve kullanıcı(lar) arasında standart bir platform sağlayan, donanımın işlevselliği ve sistem kaynaklarının yönetiminden sorumlu yazılımdır.

Bazı gömülü sistemlerde işletim sistemi bulunmaz ve kullanıcı doğrudan mikrodenetleyiciyi kodlamak zorundadır. Ancak Bilgisayar gibi genel amaçlı ve büyük sistemlerde işletim sistemi bulunur ve uygulama geliştirme, işletim sistemlerinin sağladığı platformlara göre gerçekleştirilir.

Window, Linux, MacOS, Novell, Unix, Android, iOS, SymbianOS, Minix, RTOS birer işletim sistemleridir ve birçok farklı dağıtım ve sürümleri mevcut olabilir.

İşletim Sistemlerinin Görevleri

POST süreci başarıyla tamamlandıktan sonra BIOS, Disk, CD-ROM, Flash bellek gibi depolama aygıtlarında yükleyici arar (Boot Loader; XP:NTLDR, WIN7:BOOTMGR, Linux:LILO vs.) ve bulduğunda görevi yükleyiciye devreder. Yükleyici ise işletim sistemini başlatır.

İşletim Sisteminin Görevleri

- **CPU Yönetimi**
- **Görev Zaman Paylaşımı**
- **Giriş-Çıkış İşlemleri**
- **Ana Bellek Yönetimi**
- **Dosya Sistemi (Disk) Yönetimi**
- **Kullanıcı Yönetimi**
- **Güvenlik ve Koruma**