

6 ŞUBAT 2023 – 04:17 | 13:24



Lisans: ODTÜ Elektrik-Elektronik Müh (2005-2011)

Master: TOBB ETÜ Bilgisayar Müh (2011-2013)

Doktora: Ankara Üni Elektrik-Elektronik Müh (2018-∞)

2011-2021 TOBB ETÜ Bilgisayar Mühendisliği Asistan

2012-2022 Roketsan

2022-... Yongatek Microelectronics (Hacettepe Teknokent)

TOBB ETÜ – BİLGİSAYAR MİMARİSİ dersi

MİMCE’de 3 dönem VHDL ile FPGA PROGRAMLAMA kursu

Roketsan’da kurum içi VHDL ile FPGA PROGRAMLAMA kursu

TOBB ETÜ – MANTIKSAL DEVRE TASARIMI dersi

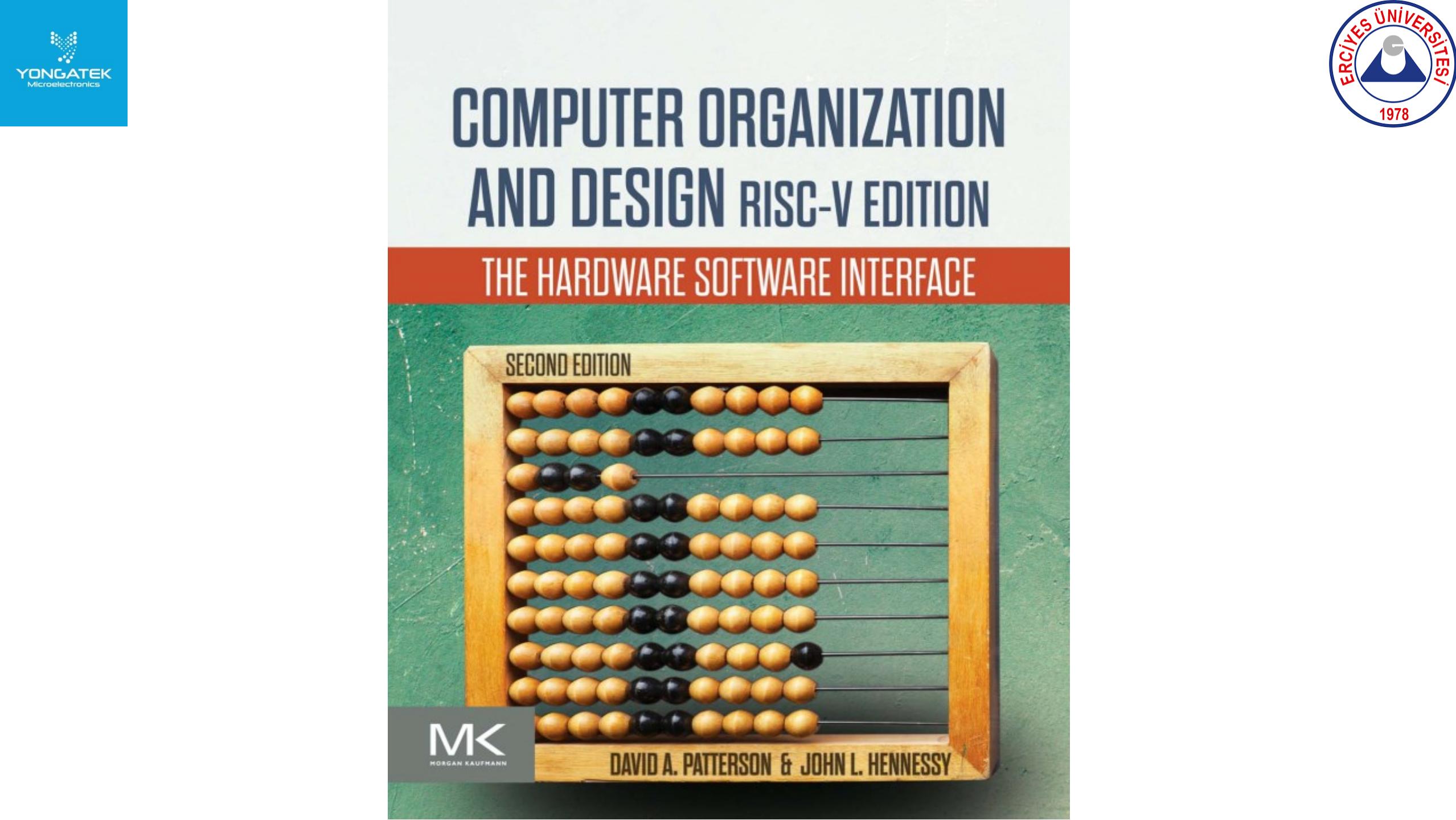
YouTube’da “VHDL ile FPGA PROGRAMLAMA DERSLERİ” serisi

YouTube’da “Mantıksal Devre Tasarımı - Logic Design” serisi

YouTube’da “ZYNQ SoC HW/SW TASARIMI” serisi

YouTube’da “ÇİP TASARIMI” serisi





Digital Design and Computer Architecture

RISC-V Edition



Sarah L Harris
David Money Harris

COMPUTER PRINCIPLES AND DESIGN IN VERILOG HDL

Yamin Li



TSINGHUA
UNIVERSITY PRESS



WILEY

Bilgisayar Mimarisi Dersinin Çerçeveesi

Prof. Dr. Oğuz Ergin



Bilgisayar Mimarisi

Oğuz Ergin

48 videos 142,304 views Last updated on Oct 18,...

≡+ ↗ ⋮
▶ Play all
🔀 Shuffle

Bilgisayar mimarisi bir bilgisayar donanımının tasarımı için bilinmesi gereken konuları kapsar. Bu ders dizisinde bilgisayar mimarisi kavramlarını, başarının nasıl ölçüldüğünü, buyruk kümesi mimarisini, işlemci tasarımını, boru hattı yöntemini, bellek, önbellek, sanal bellek konularını öğrenebileceğiniz gibi güncel mimarilerin iç yapısı ve çözmeye çalıştığı

- Bilgisayar Mimarisi Dersinin Çerçeveesi**

Prof. Dr. Oğuz Ergin



4:25

Bilgisayar Mimarisi Dersinin Çerçeveesi

Oğuz Ergin • 37K views • 2 years ago
- Bilgisayar Mimarısına Giriş**

Prof. Dr. Oğuz Ergin



58:12

Bilgisayar Mimarısına Giriş

Oğuz Ergin • 18K views • 2 years ago
- Bilgisayar Başarıımı**

Prof. Dr. Oğuz Ergin



1:05:33

Bilgisayar Başarıımı

Oğuz Ergin • 10K views • 2 years ago
- Apple M1 Yoncasının İçinde Ne Var?**

Prof. Dr. Oğuz Ergin

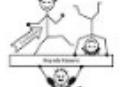


30:24

What's inside Apple's M1 Chip?

Oğuz Ergin • 11K views • 2 years ago
- Buyruk Kümesi - 1**

Prof. Dr. Oğuz Ergin



1:15:43

Buyruk Kümesi - 1

Oğuz Ergin • 11K views • 2 years ago
- Buyruk Kümesi - 2**

Prof. Dr. Oğuz Ergin



1:00:20

Buyruk Kümesi - 2

Oğuz Ergin • 7K views • 2 years ago

Livestream - Digital Design and Computer Architecture - ETH...

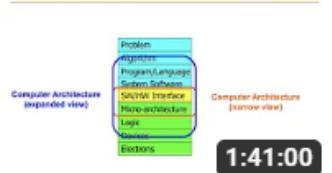
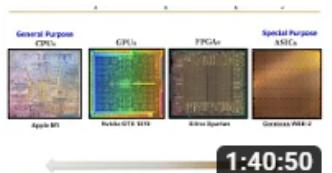
Onur Mutlu Lectures
36 videos 38,334 views Last updated on Oct 8, 2022

⋮

▶ Play all ⚡ Shuffle

Onur Mutlu's livestream lecture videos from the freshman-level Digital Design and Computer Architecture course taught at ETH Zürich in Spring 2022.

Course website including all slides and assignments:
<http://www.cseweb.ee.ethz.ch/~onur/>

- 1  1:41:00
Digital Design and Computer Architecture - Lecture 1: Introduction and Basics (2022)
 Onur Mutlu Lectures • 15K views • Streamed 1 year ago
- 2  1:45:45
Digital Design and Comp. Arch. - Lecture 2: Tradeoffs, Metrics, Mysteries in Comp Arch (Spring 2022)
 Onur Mutlu Lectures • 7.4K views • Streamed 1 year ago
- 3  1:36:55
Digital Design & Computer Arch. - Lecture 3: Mysteries in Comp Arch., FPGAs, Labs (Spring 2022)
 Onur Mutlu Lectures • 5.2K views • Streamed 1 year ago
- 4  1:40:50
Digital Design & Computer Architecture - Lecture 4: Combinational Logic I (Spring 2022)
 Onur Mutlu Lectures • 4.6K views • Streamed 1 year ago
- 5  1:46:15
Digital Design & Computer Architecture - Lecture 5: Combinational Logic II (Spring 2022)
 Onur Mutlu Lectures • 4.3K views • Streamed 11 months ago
- 6  1:45:51
Digital Design & Computer Architecture - Lecture 6: Sequential Logic Design (Spring 2022)
 Onur Mutlu Lectures • 3.9K views • Streamed 11 months ago

DERSLER VE PUANLAMALAR

Puanlama:

30 proje + 25 Vize + 45 Final

Bonus:

10 Ödevler

Proje:

4'erli gruplar oluşturulacak. İleride açıklayacağım.

Dersi Alanlar:

YZ210 – BZ403

Ders Saati:

Salı 08:30 – 10:30

DERSLERİ KAYDEDECEĞİM

YOUTUBE KANALIMDA PAYLAŞACAĞIM (Mehmet Burak Aykenar)

burak.aykenar@yongatek.com

Linkedin:

Mehmet Burak AYKENAR

- 1. Hafta:** Sayı sistemleri, onluk/ikilik taban sayı gösterimleri, mantıksal kapılar, computer system overview, başarım (performance)
- 2. Hafta:** Birleşik (Combinational), sıralı (sequential) mantıksal devre ve sonlu durum makinası (Finite state machine) tasarımı
- 3. Hafta:** Verilog donanım tanımlama dili ve örnek uygulamalar
- 4. Hafta:** Aritmetik devre tasarımları: Toplama, çıkarma, çarpma, bölme
- 5. Hafta:** Fixed ve Floating-Point sayı gösterimleri
- 6. Hafta:** RISC-V buyruk kümesi mimarisi (ISA) ve buyrukların tanıtımı
- 7. Hafta:** RISC-V buyruk kümesi mimarisi (ISA) ve buyrukların tanıtımı
- 8. Hafta:** Tek-çevrim işlemci tasarımı (single-cycle CPU)
- 9. Hafta:** Çok-çevrim işlemci tasarımı (multi-cycle CPU)
- 10. Hafta:** Boruhatlı işlemci tasarımı (pipelined CPU)
- 11. Hafta:** Bellek sistemi ve hiyerarşisi
- 12. Hafta:** İleri mimari konuları: Branch prediction, superscalar cpu, out-of-order execution, multi-core systems
- 13. Hafta:** Gömülü sistemler, mikrodenetleyiciler, SoCs

PIAZZA – Ders Notu, Soru-Cevap vs.

PIAZZA

BZ 403 ▾ Q & A Resources Statistics Manage Class

New Post Search or add a post...

LIVE Q&A Drafts hw1 hw2 hw3 hw4 hw5 hw6 hw7 hw8 hw9 hw10 project exam logistics other

Unread Updated Unresolved Following i ▾

Show Actions

PINNED

Private Search for Teammates! 3/20/23

TODAY

Private Introduce Piazza to your stu... 11:28 AM

Private Get familiar with Piazza 11:28 AM

Private Tips & Tricks for a successf... 11:28 AM

Welcome to Piazza! 11:28 AM

Piazza is a Q&A platform designed to get you great answers from classmates and instructors fast. We've put together thi

We're asking you to make a financial contribution if Piazza has delivered value to you this term [\(Why am I seeing this?\)](#)

You are seeing this prompt because your school has not purchased a Piazza enterprise license (administrators, click here to obtain a license). Contributions from users like yourself allow us to continue offering an unpaid version of Piazza. If Piazza has delivered value to you, take a minute to make a financial contribution. This prompt goes away for the remainder of the term after you've made a contribution. If you are one of our contributors, we thank you.

\$1 \$2 \$3

Payment Terms: By clicking to contribute, you authorize Piazza to charge your payment method and agree that your use of and access to the Piazza Service will be governed by these Payment Terms, in addition to Piazza's Terms of Service. In the event of a conflict between these Payment Terms and the Terms of Service, these Payment Terms shall govern.

Class at a Glance Updated 3 minutes ago. Reload Go to Live Q&A

no unread posts

no unanswered questions

no unanswered followups

license status contribution-supported
5 total posts
5 total contributions
0 instructors' responses
0 students' responses
1 sec avg. response time

Student Enrollment 0 enrolled out of 50 (estimated) Edit

https://piazza.com/demo_login?nid=lfgeki31mrb4z4&auth=63235cd

Gmail

Postalarda arama yapın

Oluştur

Ayarlar

Genel Etiketler Gelen Kutusu **Hesaplar ve İçe Aktarma İşlemi** Filtreler ve Engellenen Adresler Yönlendirme ve POP/I

Çevrimdışı Temalar

Hesap ayarlarını değiştir:

- Şifreyi değiştir
- Şifre kurtarma seçeneklerini değiştir
- Diger Google Hesabı ayarları

Gmail ürününi iş amaçlı kullanıyo musunuz?

İşletmeler Google Workspace ile, adiniz@example.com e-posta hesabından, daha fazla yararlanabilir. [Daha fazla bilgi edinin](#)

Postaları ve kişileri içe aktar:

Yahoo!, Hotmail, AOL veya diğer e-posta ya da POP3 hesaplarından alın.

[Postaları ve kişileri içe aktar](#)

Postaları şu adresden gönder:

Burak a <aykenar.burak@gmail.com>

(Diğer e-posta adreslerinizden göndermek için Gmail ürünü kullanın)

[Bir e-posta adresi daha ekle](#)

Diger hesaplardaki postaları kontrol et:

[\(POP3\)](mailto:maykenar@etu.edu.tr)

Son kontrol: 46 dakika önce. [Geçmiş görsütle](#) [Şimdi postaları kontrol et](#)

[Daha fazla bilgi edinin](#)

[Posta hesabı ekle](#)

Gelen Kutusu

Yıldızlı

Ertelenenler

Önemli

Gönderilmiş Postalar

Taslaklar 1

Spam

Kategoriler

Diğer

Etiketler +

Asistanlık

Deleted Items

GEM5



ÖN SÖZ – İLK UYARILAR

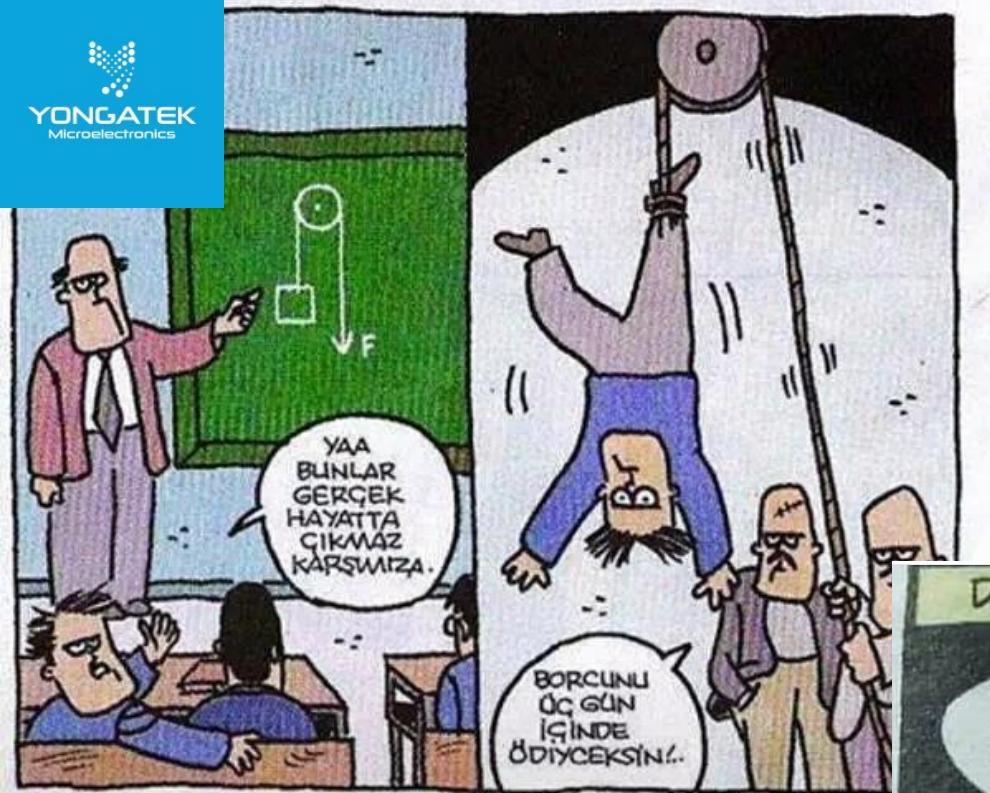
Ders çok kolay !!!

Gerçekten !!!

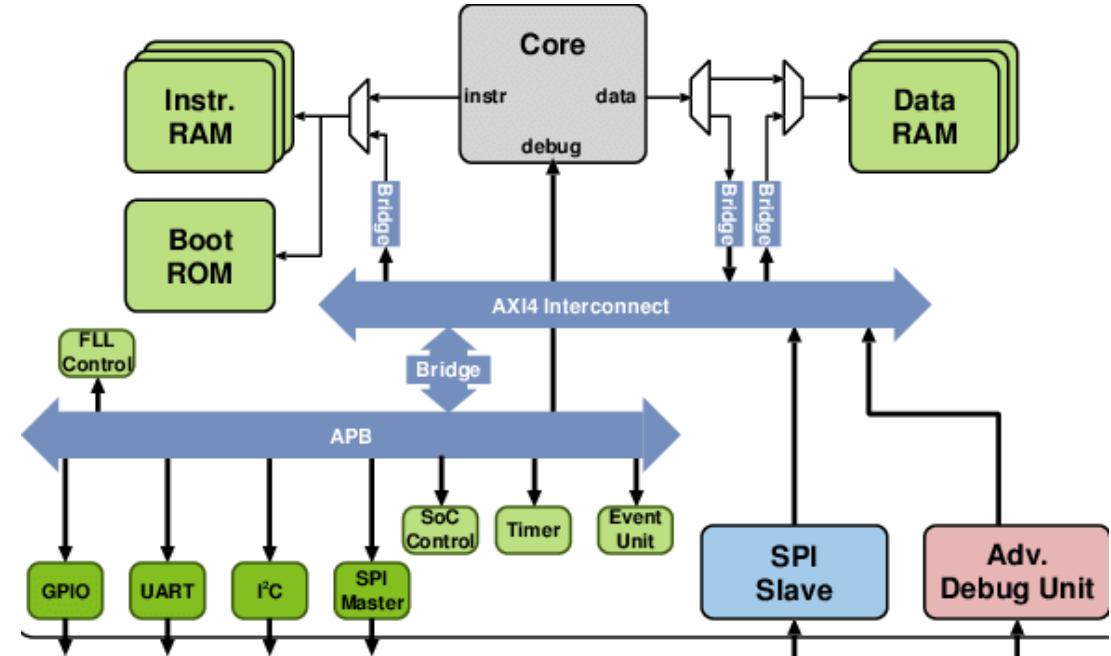


Bu dersi öğrenince ne olacak?

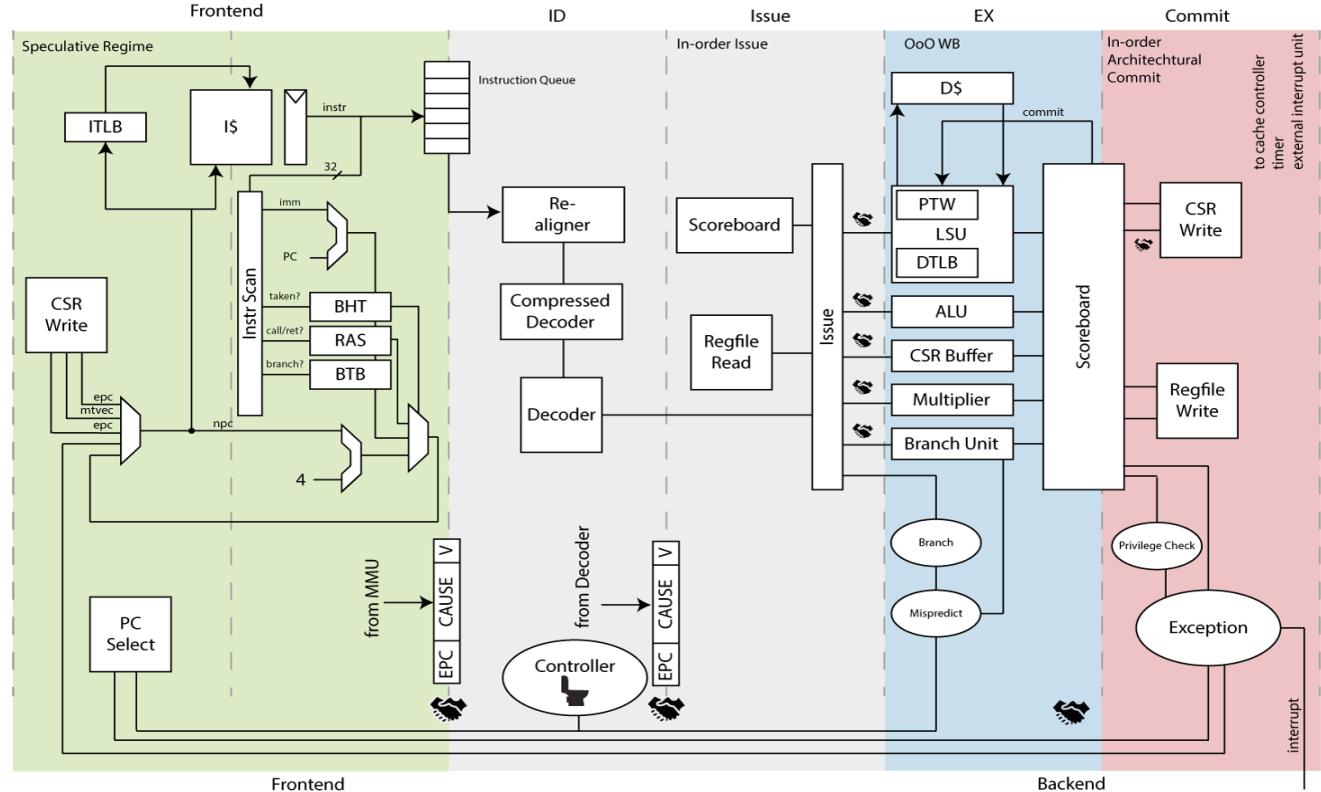
Gerçek hayatta!? ne işimize yarayacak?



CPU – SoC Mimarileri

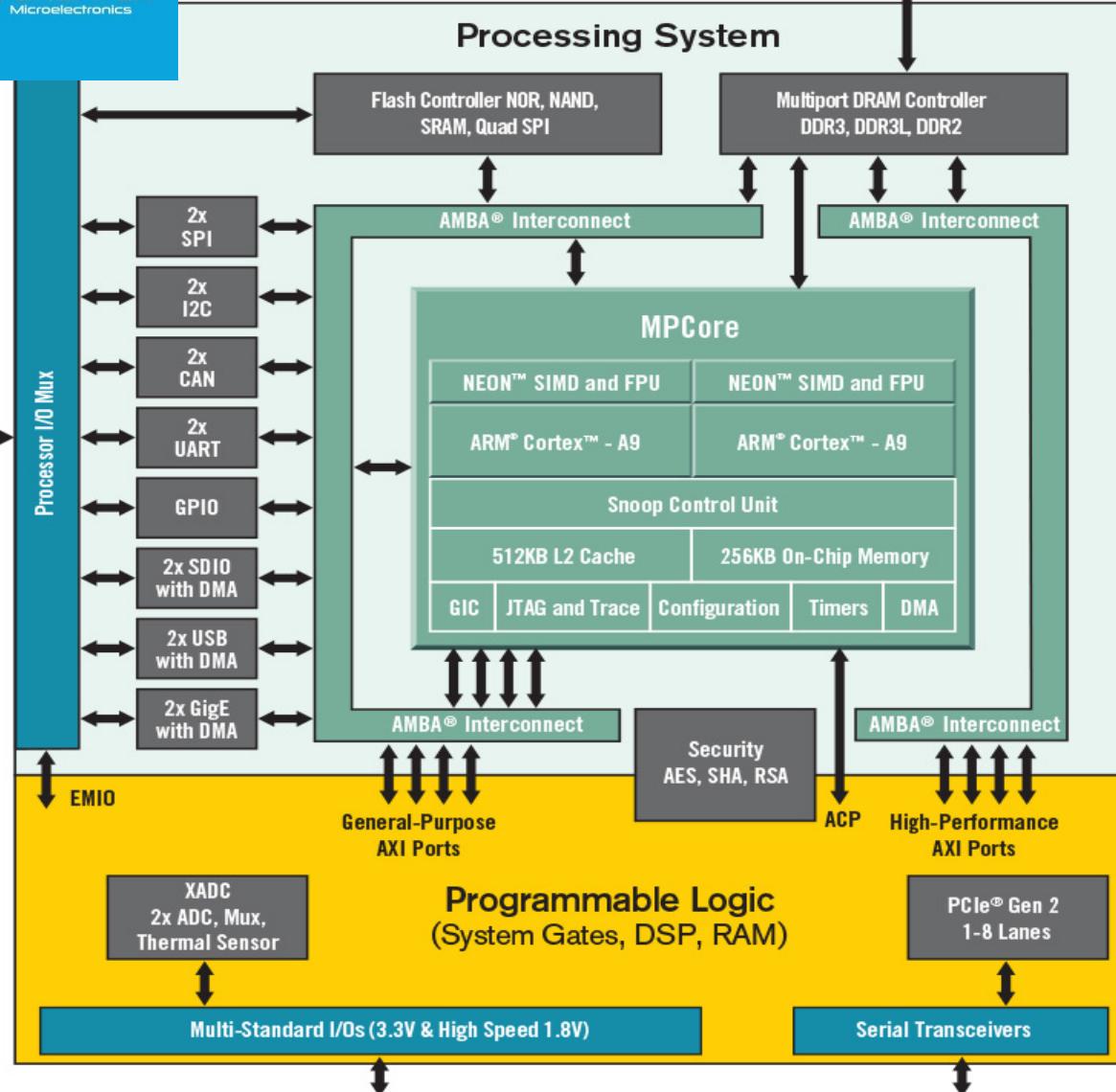


Pulpino SoC: ETH Pulp Team

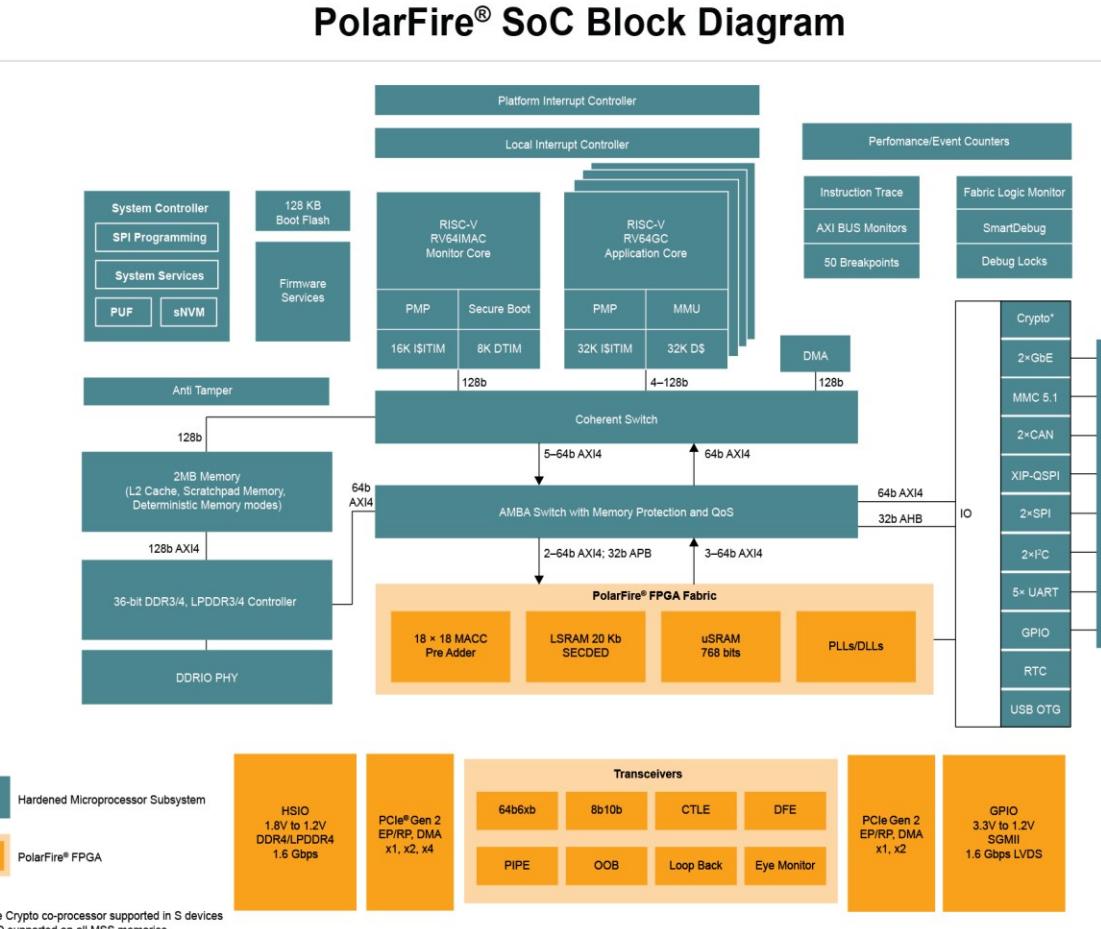


CVA6 RISC-V CPU: OPENHWGROUP

CPU – SoC Mimarileri

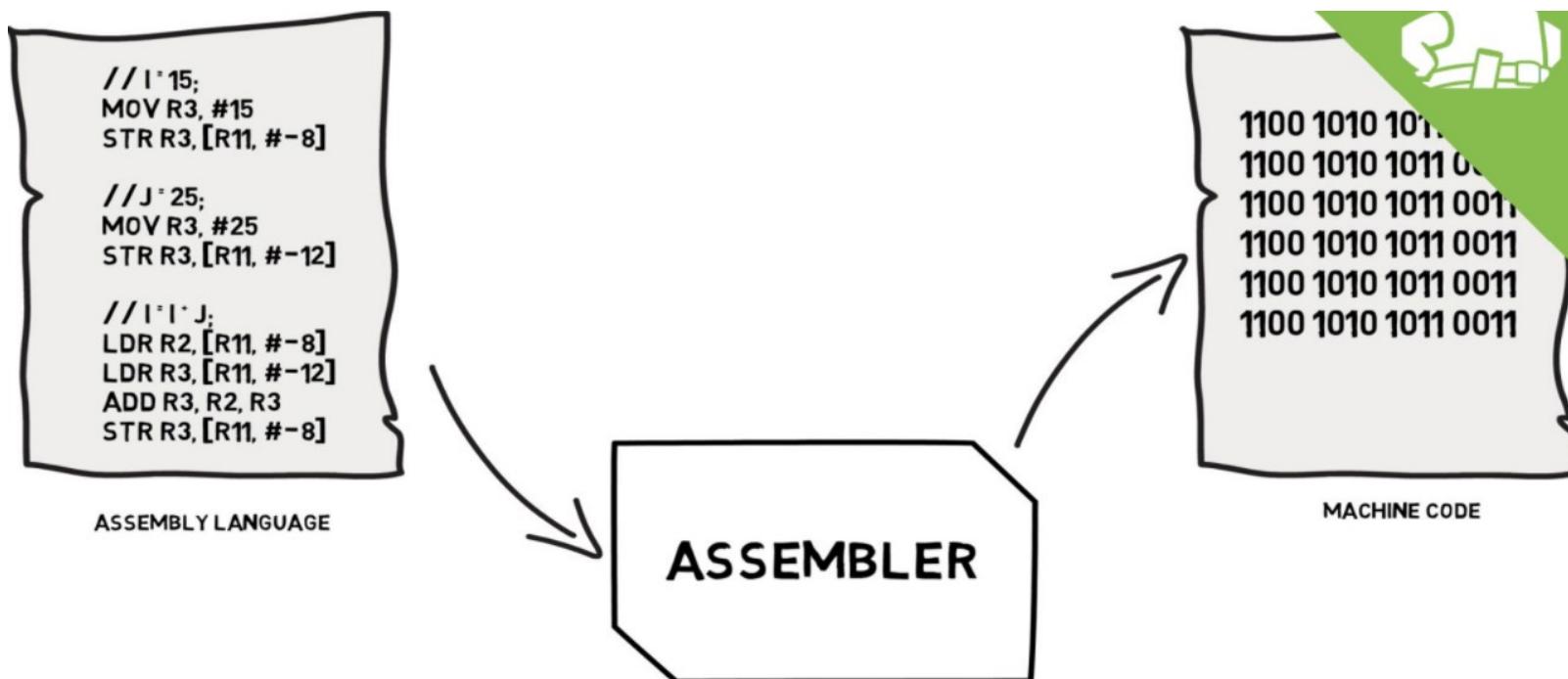


XILINX ZYNQ SoC



MICROCHIP POLARFIRE SoC

HOW COMPUTERS WORK ?



High-level
language
program
(in C)

```
swap(size_t v[], size_t k)
{
    size_t temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

Compiler

Assembly
language
program
(for RISC-V)

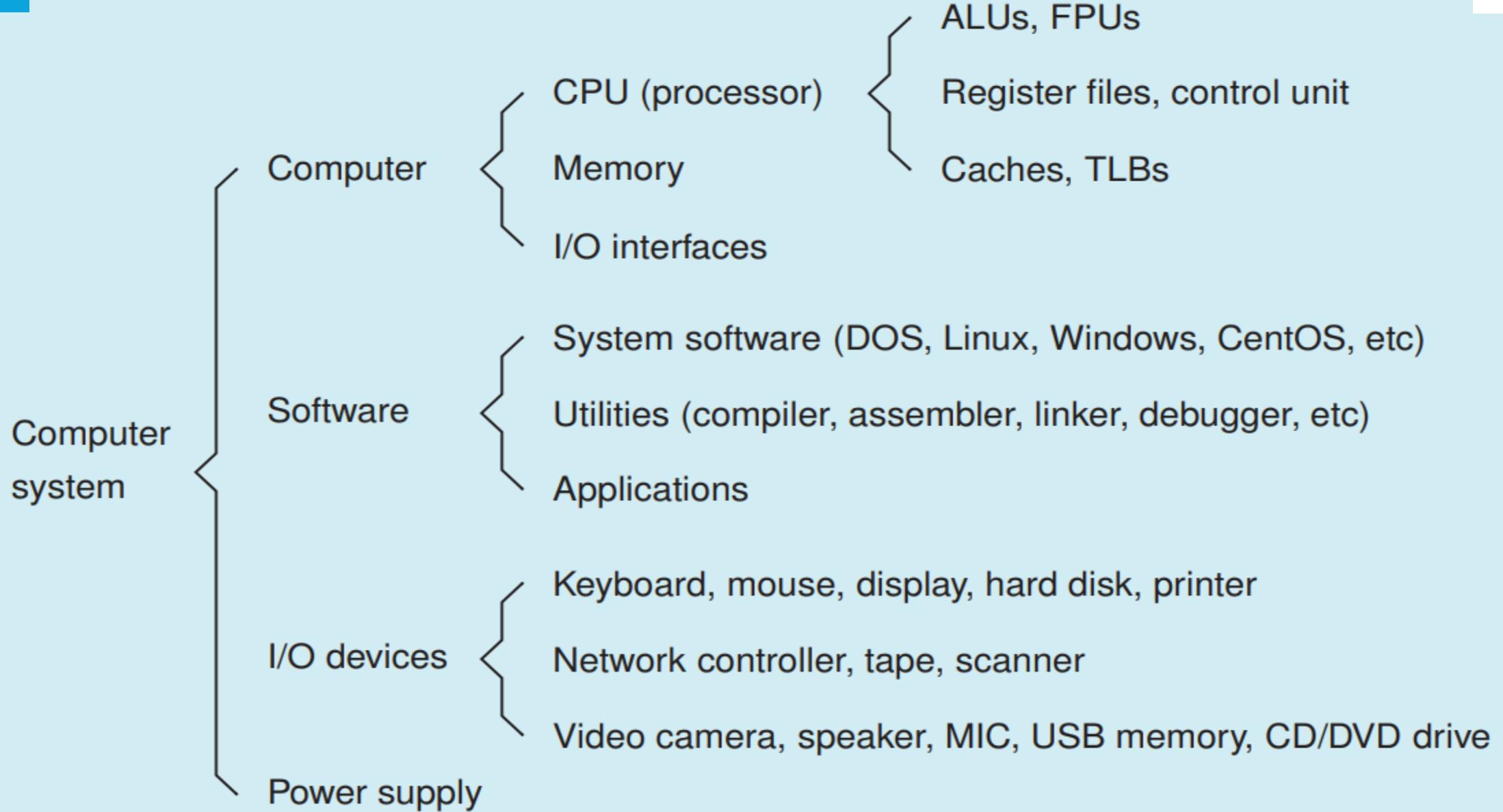
```
swap:
    slli x6, x11, 3
    add x6, x10, x6
    lw x5, 0(x6)
    lw x7, 4(x6)
    sw x7, 0(x6)
    sw x5, 4(x6)
    jalr x0, 0(x1)
```

Assembler

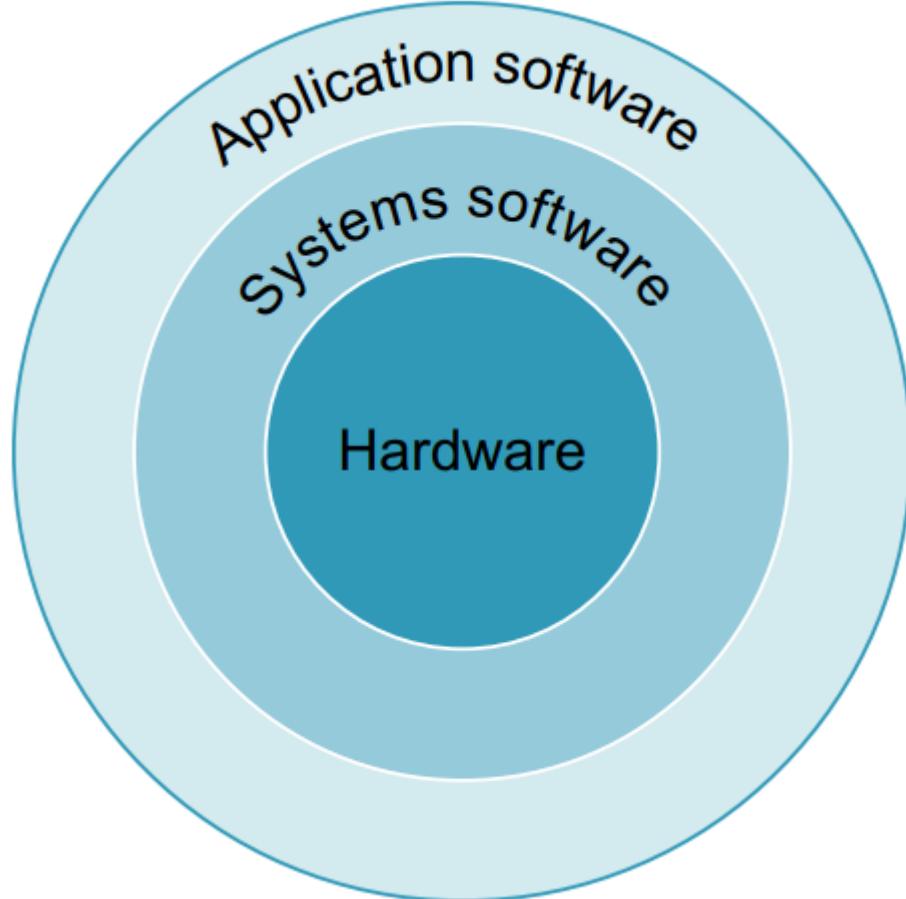
Binary machine
language
program
(for RISC-V)

```
000000000001101011001001100010011
0000000000011001010000001100110011
0000000000000000110011001010000011
0000000001000001100110011100000011
000000000011100110011000000100011
000000000010100110011010000100011
000000000000000000001000000001100111
```

COMPUTER SYSTEM



COMPUTER ABSTRACTION LAYERS

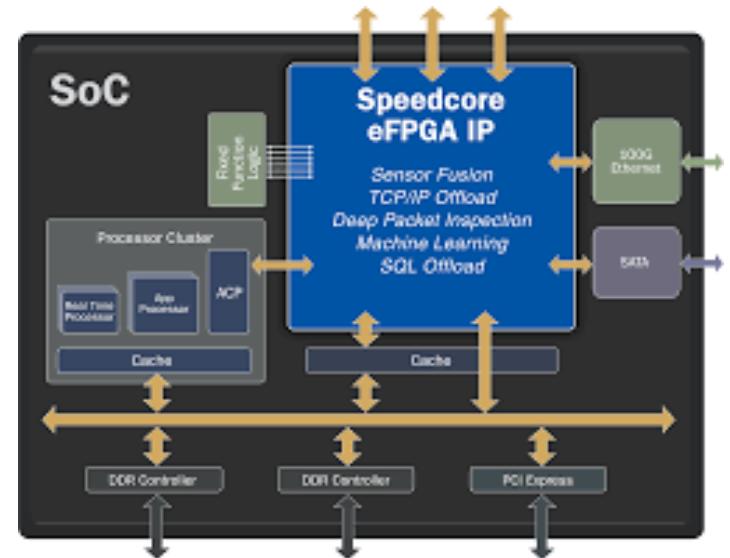


APPLICATION SW: Microsoft Office, Chrome, Adobe PDF Reader, MATLAB, Half-Life, Dota, Calculator, Ziraat Mobil, A101 Plus, etc

SYSTEM SW: Microsoft Windows, Linux, Android, MacOS, GNU Compiler Collection(GCC), Clang, Microsoft Visual Compiler, Linkers, Loaders, Device Drivers, etc.

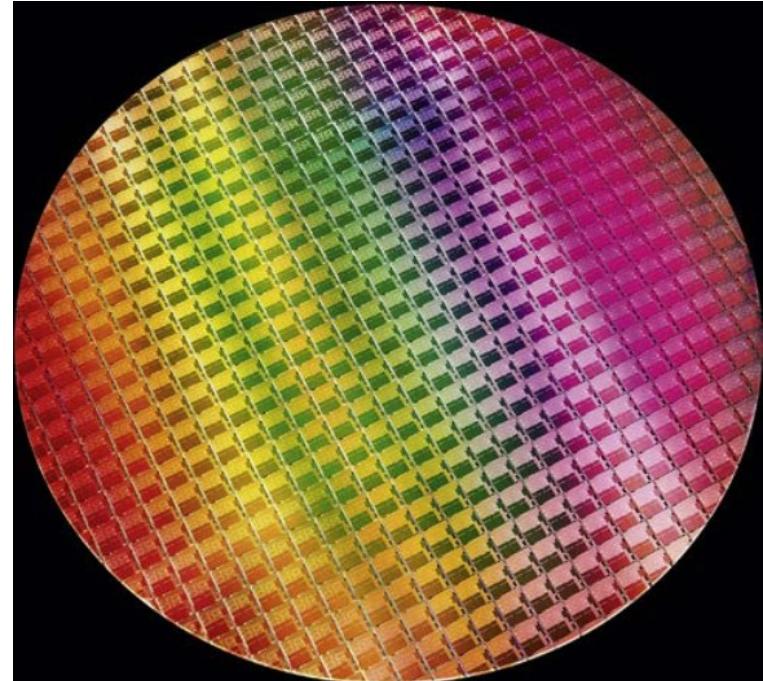
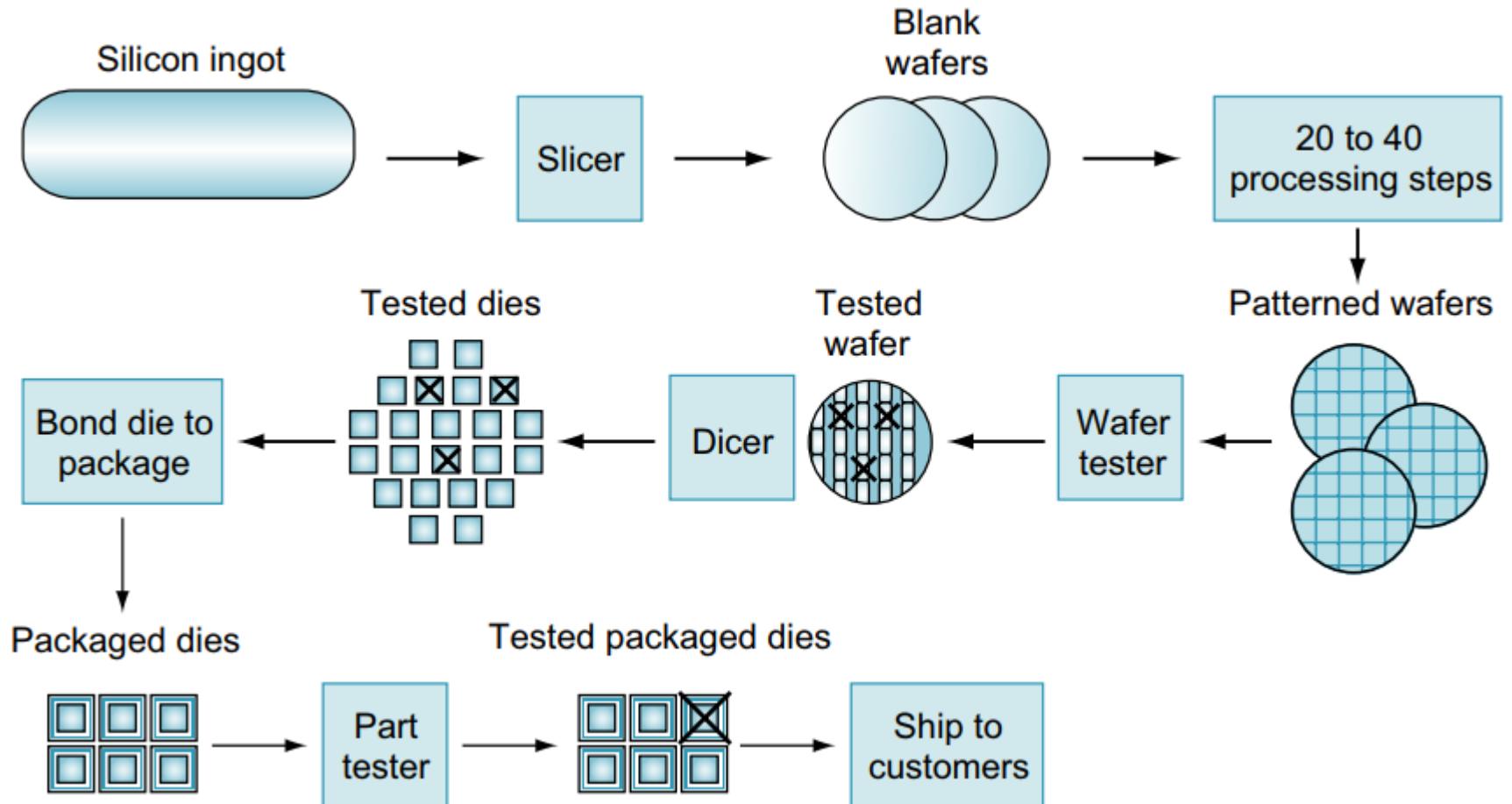
HW: CPU, Memory, I/O, Hard Disk, etc.

FPGA Sayısal Tasarım



RTL Tasarım: HDLs → Verilog, VHDL, SystemVerilog

ASIC Tasarım



Embedded Systems - IoT

- NXP: ARM and PowerPC based MCUs
- Microchip: PIC and AVR based MCUs (also ARM & MIPS based in 32-bit MCUs)
- Renesas: Renesas, ARM and RISC-V based MCUs
- ST: ARM based MCUs (also ST based in 8-bit MCUs)
- Infineon: ARM and Tricore based MCUs

Leading MCU suppliers (USD million)							
2021 Rank	Company	Headquarters	2020	2021	21/20 change (%)	2021 marketshare	
1	NXP	Europe	2,980	3,795	27%	18.8%	
2	Microchip	USA	2,872	3,584	25%	17.8%	
3	Renesas	Japan	2,748	3,420	24%	17.0%	
4	ST	Europe	2,506	3,374	35%	16.7%	
5	Infineon	Europe	1,953	2,378	22%	11.8%	



MICROCHIP



life.augmented



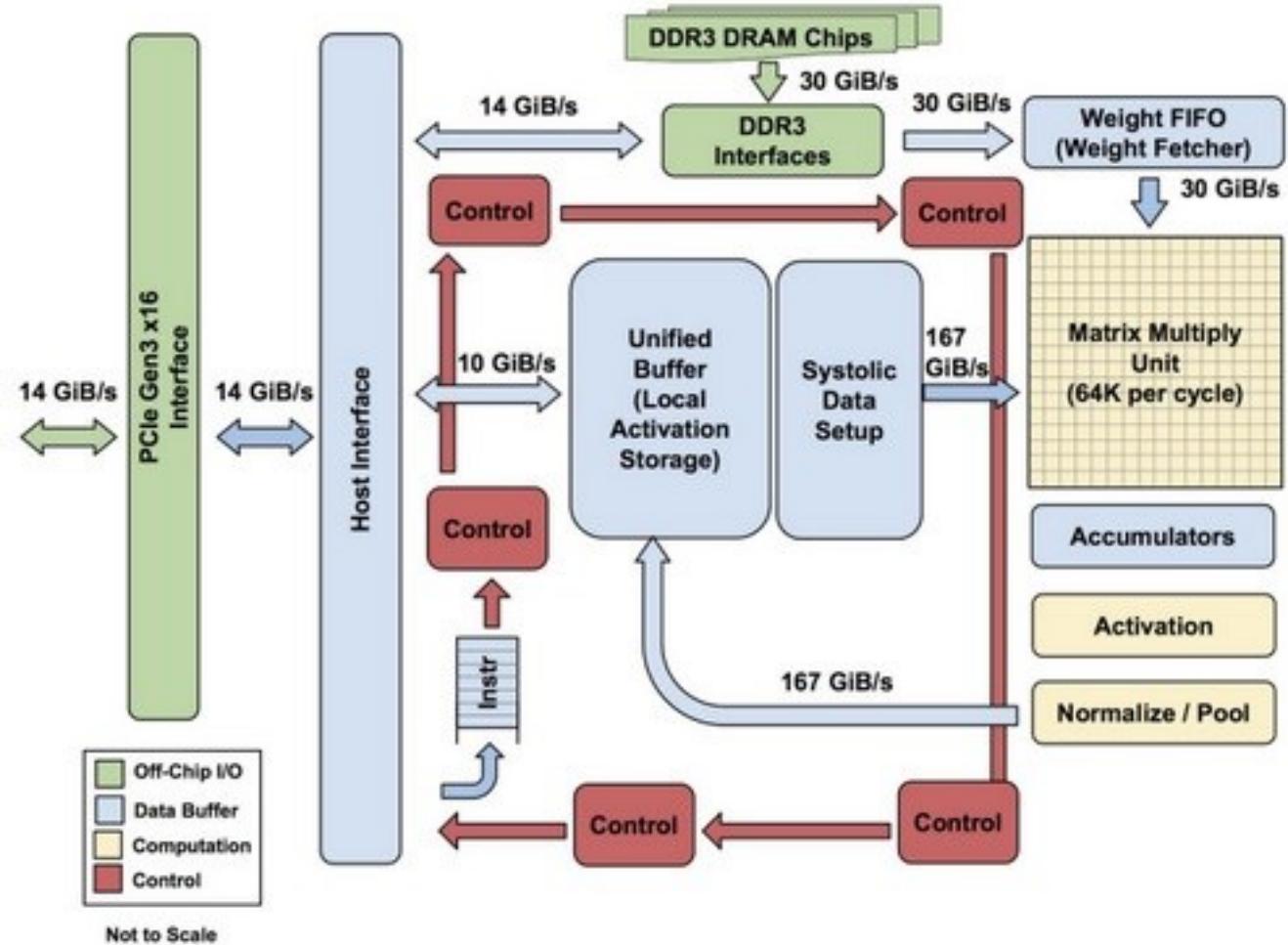
Embedded Systems - IoT

ESP32-C3

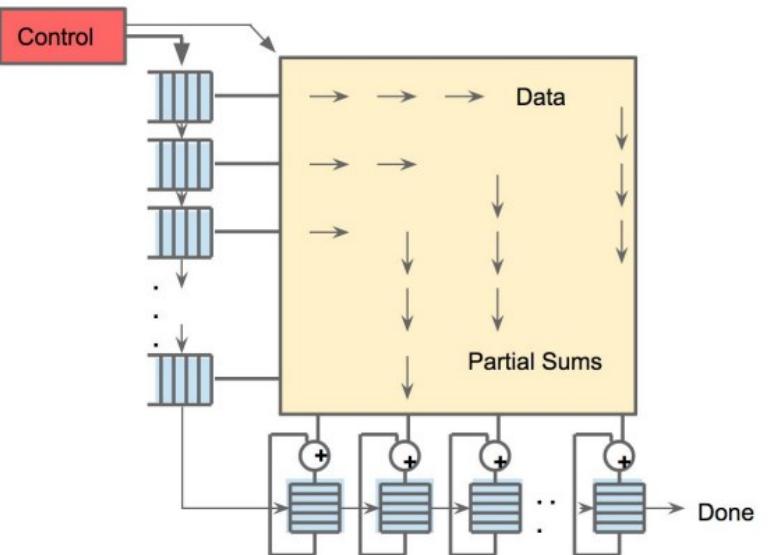
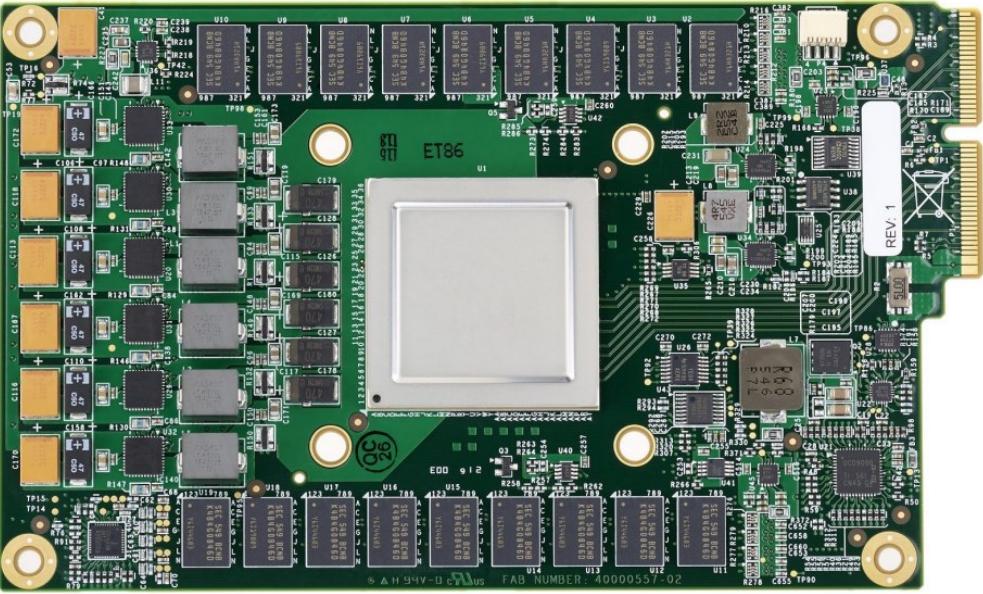
A cost-effective RISC-V MCU with Wi-Fi and Bluetooth 5 (LE) connectivity for secure IoT applications



AI REVOLUTION



Google TPU



AI REVOLUTION



- ▶ Six GPCs
 - Each GPC has:
 - Seven TPCs (each including two SMs)
 - 14 SMs
- ▶ 84 Volta SMs
 - Each SM has:
 - 64 FP32 cores
 - 64 INT32 cores
 - 32 FP64 cores
 - 8 Tensor Cores
 - Four texture units
- ▶ Eight 512-bit memory controllers (4096 bits total)

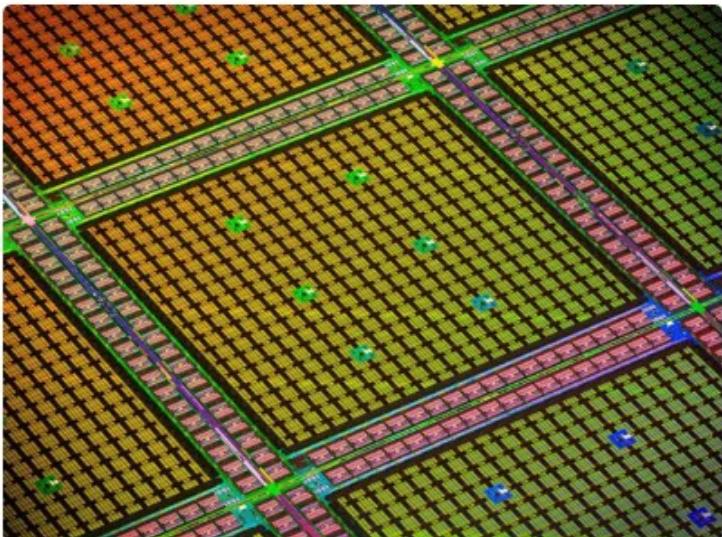
NVIDIA Volta GV100

TESLA Makes It's own Chips



FSD Chip

Build AI inference chips to run our Full Self-Driving software, considering every small architectural and micro-architectural improvement while squeezing maximum silicon performance-per-watt. Perform floor-planning, timing and power analyses on the design. Write robust tests and scoreboards to verify functionality and performance. Implement drivers to program and communicate with the chip, focusing on performance optimization and redundancy. Finally, validate the silicon chip and bring it to mass production in our vehicles.



Dojo Chip

Build AI training chips to power our Dojo system. Implement bleeding-edge technology from the smallest training nodes to the multi-die training tiles. Design and architect for maximum performance, throughput and bandwidth at every granularity. Dictate physical methodology, floor-planning and other physical aspects of the chip. Develop pre-silicon verification and post-silicon validation methods to ensure functional correctness. Write compilers and drivers to optimize power and performance for our neural networks throughout the entire Dojo system. For more information about Dojo's arithmetic formats and methods, [download our latest whitepaper](#).

APPLE Makes It's own Chips

5 nanometer process



Thunderbolt / USB 4 controller

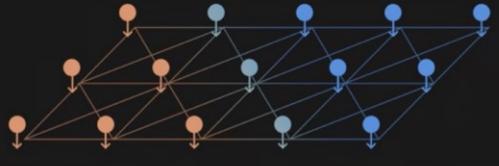


Media encode and decode engines

16 billion
transistors



Advanced image signal processor



Machine learning accelerators

Apple M1

16-core

Neural Engine

11 trillion operations per second

Up to

8-core GPU

8-core CPU



Secure Enclave



Unified memory architecture

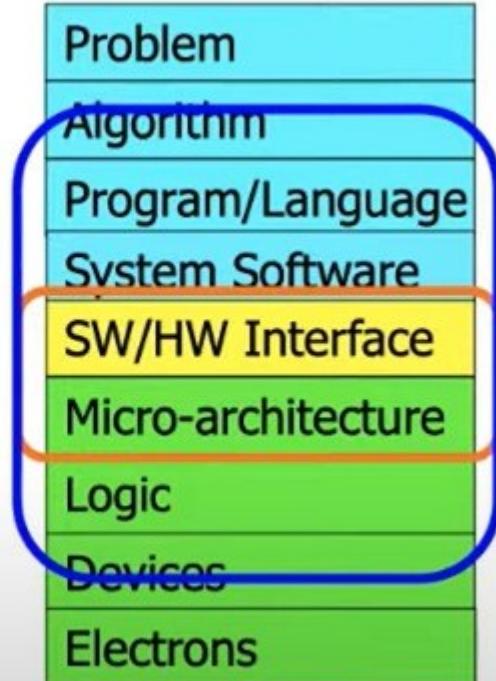
Industry-leading performance per watt



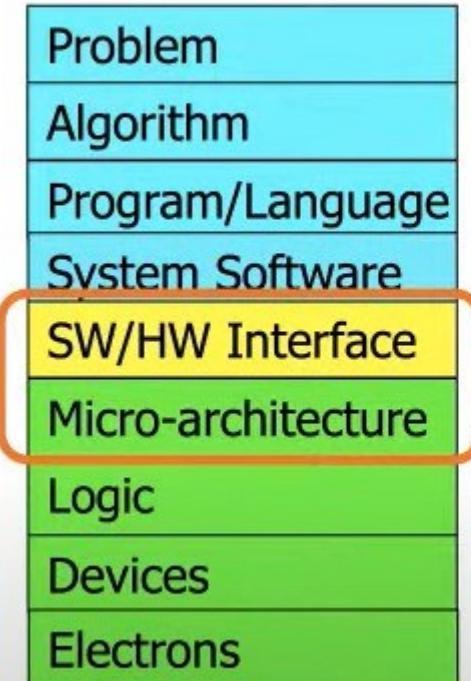
KASIRGA'YA TEBRİKLER !!!

DERSİN GENEL ÇERÇEVESİ

**Computer Architecture
(expanded view)**



**Computer Architecture
(narrow view)**



Taban (Base) Nedir?

Base : Temel, taban (ing. foundation)

Radix : Kök, temel (ing. root) – Radical: Kökten, temelden – Örn. Radikal Değişim

İnsanlık tarihi: 10'luk taban (decimal) – 10 parmak

Decimal: Onluk (Lat. Decimus)

Latince: 1-10

Unus(1), Duo(2), Tres(3), Quattuor(4), Quinque(5), Sex(6), Septem(7), Octo(8),
Novem(9), Decem(10)

10'luk Taban

$$10864 \mid (10864)_{10}$$

$$1*10^4 + 0*10^3 + 8*10^2 + 6*10^1 + 4*10^0 =$$

$$10000 + 0 + 800 + 60 + 4$$

$$10864$$

10'luk taban kümesi: {0,1,2,3,4,5,6,7,8,9}

N taban kümesi: {0,1,...,N-1}

Ondalıklı sayılar ve genel kuram:

$$a_5a_4a_3a_2a_1a_0 \cdot a_{-1}a_{-2}a_{-3}$$

$$10^5a_5 + 10^4a_4 + 10^3a_3 + 10^2a_2 + 10^1a_1 + 10^0a_0 + 10^{-1}a_{-1} + 10^{-2}a_{-2} + 10^{-3}a_{-3}$$



There are only 10 types
of people in the world:
Those who understand binary
and those who don't.

2'lük Taban

$(11001)_2$

$$1*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 =$$

$$16 + 8 + 0 + 0 + 1$$

$(25)_{10}$

Ondalıklı sayılar ve genel kuram:

$$(11010.11)_2 = (?)_{10}$$

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 26.75$$

Genel Taban Kuralları ve Örnekler

$$\begin{aligned} & a_n \cdot r^n + a_{n-1} \cdot r^{n-1} + \dots + a_2 \cdot r^2 + a_1 \cdot r + a_0 + a_{-1} \cdot r^{-1} \\ & + a_{-2} \cdot r^{-2} + \dots + a_{-m} \cdot r^{-m} \end{aligned}$$

$$(4021.2)_5 = 4 \times 5^3 + 0 \times 5^2 + 2 \times 5^1 + 1 \times 5^0 + 2 \times 5^{-1} = (511.4)_{10}$$

$$(127.4)_8 = 1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} = (87.5)_{10}$$

$$(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15 \times 16^0 = (46,687)_{10}$$

Hex karakterler : 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Örnek Gösterim

: 0x35 → 0011 0101 (8-bit = 1 byte)

: 0x13AF → 0001 0011 1010 1111 (16-bit / 1 word)

: 0xA7D394FB → 1010 0111 1101 0011 1001 0100 1111 1011 (32-bit / 1 word)

Taban Dönüşümü (Base Conversion)

$$(11001)_2 = (?)_{10}$$

$$1*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 =$$

$$16 + 8 + 0 + 0 + 1$$

$$(25)_{10}$$

$$(25)_{10} = (?)_2$$

Sayı	Bölüm	Kalan	Bit Değer	Bit Sıra
25	12	1	1	0
12	6	0	0	1
6	3	0	0	2
3	1	1	1	3
1	0	1	1	4

25'e en yakın 2'nin üssel değeri: $2^4 = 16$, $25-16=9$

9'a en yakın 2'nin üssel değeri: $2^3 = 8$, $9-8=1$

1'e en yakın 2'nin üssel değeri: $2^0 = 1$

$$1*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 = 11001$$

Taban Dönüşümü (Base Conversion)

$$(25)_{10} = (?)_2$$

Sayı	Bölüm	Kalan	Bit Değer	Bit Sıra
25	12	1	1	0
12	6	0	0	1
6	3	0	0	2
3	1	1	1	3
1	0	1	1	4

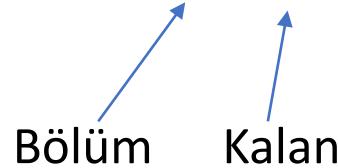
25'e en yakın 2'nin üssel değeri: $2^4 = 16$, $25-16=9$

9'a en yakın 2'nin üssel değeri: $2^3 = 8$, $9-8=1$

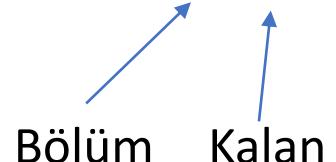
1'e en yakın 2'nin üssel değeri: $2^0 = 1$

$$1*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 = 11001$$

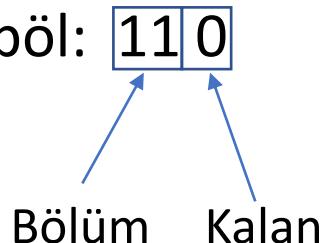
$$11001 \rightarrow 2'ye \text{ böl}: \boxed{1100} \boxed{1}$$



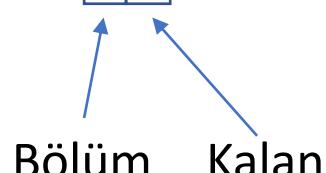
$$1100 \rightarrow 2'ye \text{ böl}: \boxed{110} \boxed{0}$$



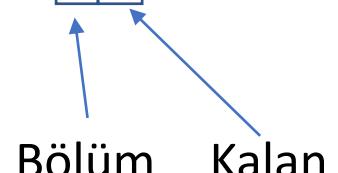
$$110 \rightarrow 2'ye \text{ böl}: \boxed{11} \boxed{0}$$



$$11 \rightarrow 2'ye \text{ böl}: \boxed{1} \boxed{1}$$



$$1 \rightarrow 2'ye \text{ böl}: \boxed{0} \boxed{1}$$



Taban Dönüşümü (Ondalıklı Sayılar)

$$(11010.11)_2 = (?)_{10}$$

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 26.75$$

$$(0.6875)_{10} = (?)_2$$

Sayı	Çarpım	Elde	Bit Değer	Bit Sıra
0.6875	0.375	1	1	-1
0.375	0.75	0	0	-2
0.75	0.5	1	1	-3
0.5	0	1	1	-4

Ondalıklı kısmı için sürekli 2 ile çarp, elde değeri bit değerini ifade eder. Bit sırası -1,-2,... şeklindedir. Çarpım '0' olana kadar devam et.

$$(0.6875)_{10} = (0.1011)_2$$

Taban Dönüşümü (Base Conversion)

$$(0.6875)_{10} = (?)_2$$

Sayı	Çarpım	Elde	Bit Değer	Bit Sıra
0.6875	0.375	1	1	-1
0.375	0.75	0	0	-2
0.75	0.5	1	1	-3
0.5	0	0	1	-4

Ondalıklı kısmı için sürekli 2 ile çarp, elde değeri bit değerini ifade eder. Bit sırası -1,-2,... şeklindedir. Çarpım '0' olana kadar devam et.

$$(0.6875)_{10} = (0.1011)_2$$

$$0.1011 \rightarrow 2 \text{ ile çarp: } 1.011$$

$$1.011 \rightarrow 2 \text{ ile çarp: } 10.11$$

$$10.11 \rightarrow 2 \text{ ile çarp: } 101.1$$

$$101.1 \rightarrow 2 \text{ ile çarp: } 1011$$

İkilik tabanda

Sola kaydırma \rightarrow 2 ile çarpma
Sağa kaydırma \rightarrow 2'ye bölme

Onluk tabanda

Sola kaydırma \rightarrow 10 ile çarpma
Sağa kaydırma \rightarrow 10'a bölme

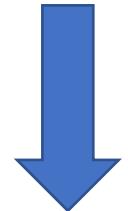
Taban Dönüşümü (Ondalıklı Sayılar)

$$(0.513)_{10} = (?)_2$$

Sayı	Çarpım	Elde	Bit Değer	Bit Sıra
0.513	0.026	1	1	-1
0.026	0.052	0	0	-2
0.052	0.104	0	0	-3
0.104	0.208	0	0	-4
0.208	0.416	0	0	-5
0.416	0.832	0	0	-6
0.832	0.664	1	1	-7
0.664	0.328	1	1	-8

Eğer ondalıklı kısmı için 8-bit ayrıldıysa 8 kere iterasyon devam ettirilir.

$$(0.513)_{10} = (0.10000011)_2$$



$$(0.10000011)_2 = (0.51171875)_{10}$$

Bilgi kaybı yaşandı!

Eğer ondalıklı kısmı için sadece 4-bit ayrılmış olsaydı 0.513 sayısı ancak 0.5 olarak ifade edilebilecekti!

Sabit Noktalı x Kayan Noktalı (Fixed-Point vs Floating Point)

$$(326.513)_{10} = (?)_2$$

Sabit-Noktalı Gösterim: Tamsayı ve ondalıklı sayılar için sabit uzunlukta yer (bit) ayrılmıştır. Sayının tamsayı ya da ondalıklı sayı kısmının büyüklüğü, bu sabit uzunluğu değiştirmez. Örnek olarak tamsayı 10, ondalıklı 8 bit olursa:

$$(326.513)_{10} = (0101000110.10000011)_2$$

Kayan-Noktalı Gösterim: Tamsayı ve ondalıklı sayılar için sabit uzunlukta yer (bit) ayrılmamıştır. Sayının tamsayı ve ondalıklı kısımlarına göre noktanın yeri değişir, kayar. Bilgisayarlarda ondalıklı sayıları ifade edebilmek için IEEE-754 Single Precision ve Double Precision standartları oluşturulmuştur.

Single Precision : 32-bit

Double Precision : 32-bit

IEEE-754 standard ve fixed-point hakkında detaylı bilgi ileriki haftalarda anlatılacak

Taban Tümleme (Radix Complement)

Çıkarma işlemini basitleştirme amacıyla kullanılmaktadır

Çıkarma işlemi, aslında toplama işleminde 2. toplanan sayının işaretinin değiştirilip toplanması ile meydana gelir

Tümleme işlemi 10'luk tabanda sayılar için 9 (10-1) ve 10'a tümleyen şeklinde tanımlanmıştır

Örn: 3: 9'a tümleyen → 6, 10'a tümleyen → 7

Örn: 9: 9'a tümleyen → 0, 10'a tümleyen → 1

Örn: 7: 9'a tümleyen → 2, 10'a tümleyen → 3

$$3+9 = 12$$

$$3-9 = -6$$

$$3-9 = 3+1-10 = -6$$

$$9+7 = 16$$

$$9-7 = 2$$

$$9-7 = 9+3-10 = 2$$

$$\begin{array}{r} 3 & 3 & 4 \\ - 9 & + 1 & - 10 \\ \hline -6 & 4 & -6 \end{array} \qquad \begin{array}{r} 9 & 9 & 12 \\ - 7 & + 3 & - 10 \\ \hline 2 & 12 & 2 \end{array}$$

Kural: Eğer 10'a tümleyenle topladığında elde yoksa sonuç kalan sayının 10'a tümleyeninin eksi işaretlidir

Kural: Eğer 10'a tümleyenle topladığında elde varsa sonuç elde hariç kalan sayıdır

Taban Tümleme (Radix Complement)

Tümleme işlemi 2'lik tabanda sayılar için 1 (2-1) ve 2'ye tümleyen şeklinde tanımlanmıştır

Örn: 0: 1'e tümleyen → 1, 2'ye tümleyen → 0

Örn: 1: 1'e tümleyen → 0, 2'ye tümleyen → 1

Örn: 1101: 1'e tümleyen → 0010, 2'ye tümleyen → 0011

$$\begin{array}{r} 0011 \\ + 1001 \\ \hline 1100 \end{array}$$

$$\begin{array}{r} 1001 \\ + 0111 \\ \hline 10000 \end{array}$$

$$\begin{array}{r} 0011 \\ + 0111 \\ \hline 1010 \end{array}$$

$$\begin{array}{r} 1001 \\ + 1001 \\ \hline 10010 \end{array}$$

Kuralı hatırla: 2. toplananın tümleyenini alıp 1. ile topladıktan sonra elde oluşmadıysa, sonuç kalan sayının tümleyeninin eksi işaretli değeridir → Kalan sayı $1010 = 10$, tümleyeni 6, sonuç -6

\rightarrow 1'e tümleyeni 0101 \rightarrow 2'ye tümleyeni 0110 \rightarrow Eksi işaretlisi -6

Kuralı hatırla: 2. toplananın tümleyenini alıp 1. ile topladıktan sonra eğer elde varsa sonuç elde hariç kalan sayıdır

\rightarrow Elde var \rightarrow Sonuç elde hariç kalan kısım $\rightarrow 0010 = 2$

Likewise, the two's complement of zero is zero: inverting gives all ones, and adding one changes the ones back to zeros (since the overflow is ignored).

 **Himanshu Bhardwaj** · Follow
engineering@microsoft · 7y

Assuming you are representing 0 as one byte .i.e $0 = (00000000)_2$

one's complement = $(11111111)_2$
 2's complement = 1's complement + 1 = $(11111111)_2 + 1 = (00000000)_2$