



GEBZE  
TEKNİK ÜNİVERSİTESİ

CSE 101  
Slide Set 1

Doç. Dr. Mehmet Göktürk  
Department of Computer Engineering

[www.gtu.edu.tr](http://www.gtu.edu.tr)

1

---

---

---

---

---

---



Welcome

This is an introduction course

- Principles of Computing
- Assembly Programming
- C Programming
- Some hardware interaction (Arduino kit)
- Introduction to Computer Engineering Career
- Invited Speakers
- Project and Homeworks Midterm + Final
- Videos will be watched and included in the exams
- Microsoft Teams resources

[www.gtu.edu.tr](http://www.gtu.edu.tr)

2

---

---

---

---

---

---



HYBRID COURSE STRUCTURE

Classes will be in class and online simultaneously

You do not have to come but will be welcomed if come

Exams are in class

Homeworks are important

Term Project

30%MT 40%FIN %20HW 10%PROJ

[www.gtu.edu.tr](http://www.gtu.edu.tr)

3

---

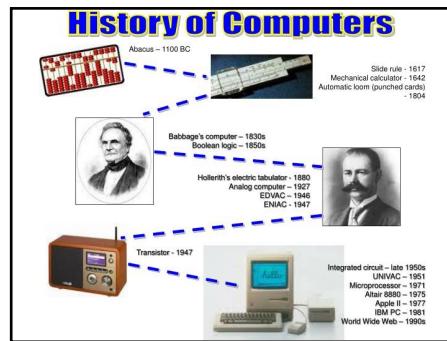
---

---

---

---

---



4

---

---

---

---

---

---

---

---



5

---

---

---

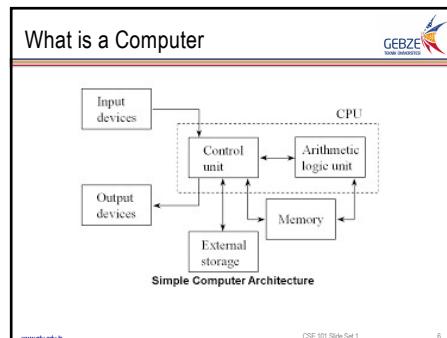
---

---

---

---

---



6

---

---

---

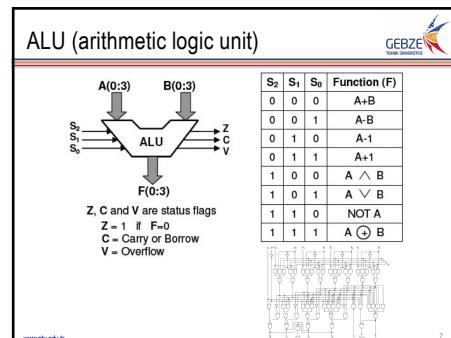
---

---

---

---

---



7

---

---

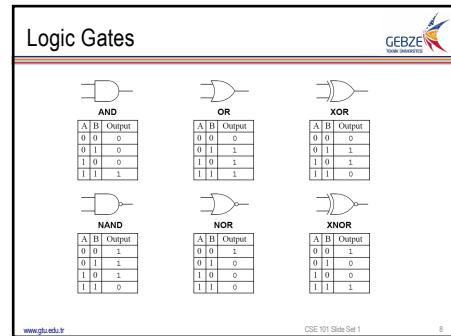
---

---

---

---

---



8

---

---

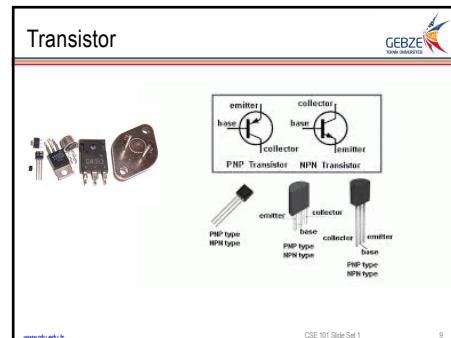
---

---

---

---

---



9

---

---

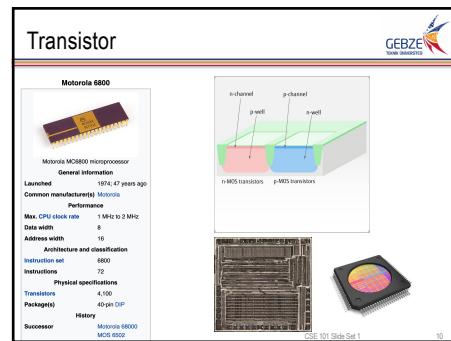
---

---

---

---

---



10

---



---



---



---



---



---



---



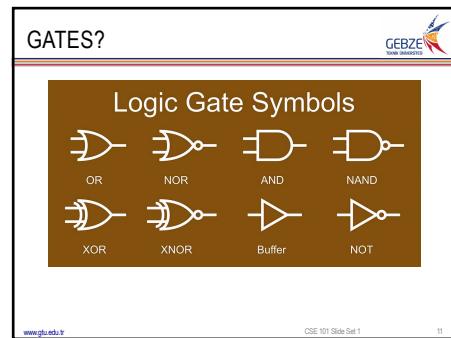
---



---



---



11

---



---



---



---



---



---



---



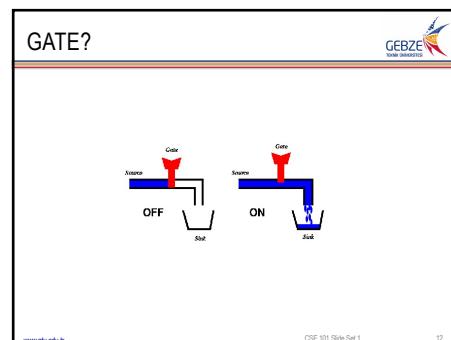
---



---



---



12

---



---



---



---



---



---



---



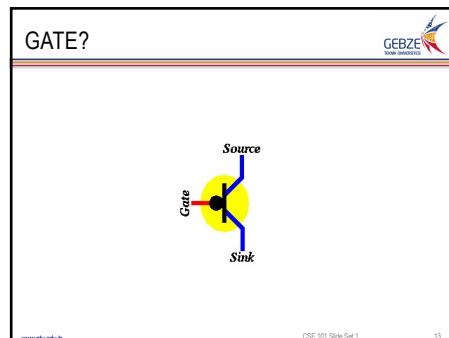
---



---



---



13

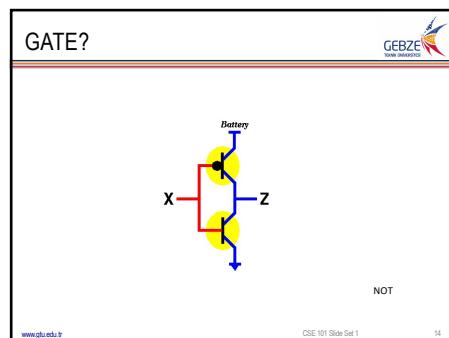
---

---

---

---

---



14

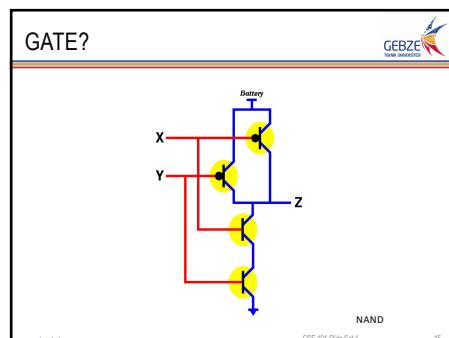
---

---

---

---

---



15

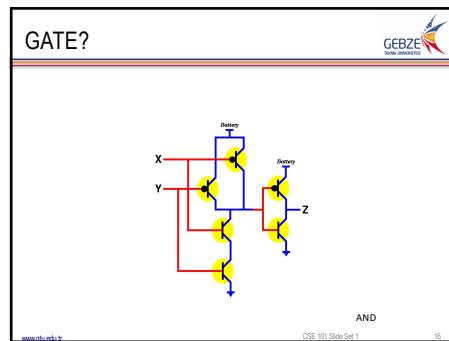
---

---

---

---

---



16

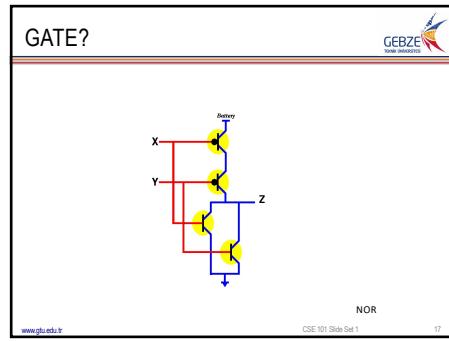
---

---

---

---

---



17

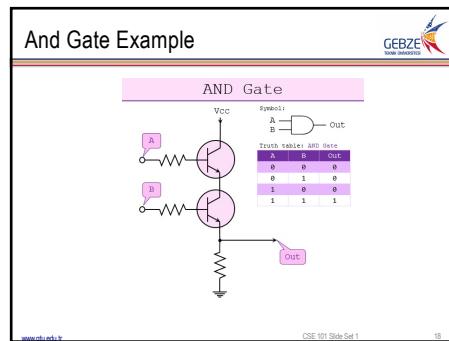
---

---

---

---

---



18

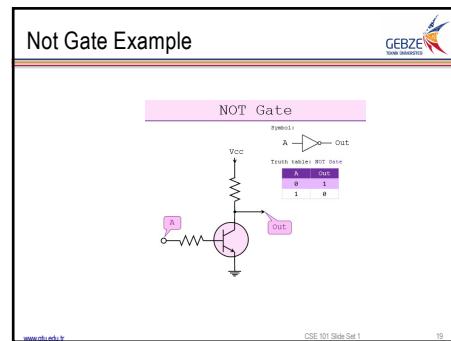
---

---

---

---

---



19

---

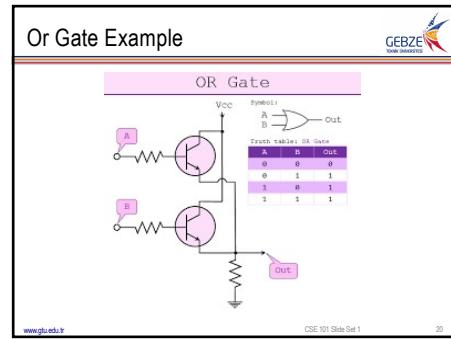
---

---

---

---

---



20

---

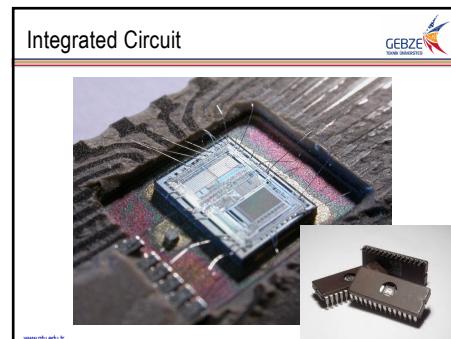
---

---

---

---

---



21

---

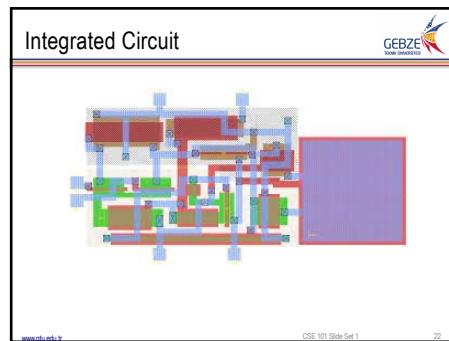
---

---

---

---

---



22

---



---



---



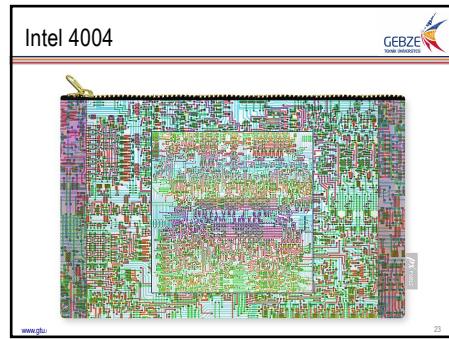
---



---



---



23

---



---



---



---



---



---

1970-76						
Processor	MOS transistor count	Date of introduction	Designer	MOS process (nm)	Area (mm²)	
SP944 (20-bit, 6-chip, 20 chips total)	74,442 (1,390 each, 20 chips total)	1970/1974	Gennett Associates	?	?	
Intel 4004 (4-bit, 16-pin)	2,300	1971	Intel	10,000 nm	12 mm²	
MX 1156 (7-bit, 24-pin)	3,079 <sup>TM</sup>	1971	Texas Instruments	?	90 nm²	
Intel 8008 (8-bit, 18-pin)	3,500	1972	Intel	10,000 nm	14 nm²	
EC 1204A (4-bit, 42-pin)	2,500 <sup>TM</sup>	1973	NEC	7,500 nm <sup>TM</sup>	59 nm²	
Intel 1100 (8-bit, 12-18-pin)	11,000 <sup>TM</sup>	1973	Toshiba	6,000 nm	59 nm²	
Intel 4040 (8-bit, 40-pin)	3,000	1974	Intel	10,000 nm	12 nm²	
Motorola 6800 (8-bit, 40-pin)	4,100	1974	Motorola	6,000 nm	16 nm²	
Intel 8008 (8-bit, 40-pin)	6,000	1974	Intel	6,000 nm	20 nm²	
MS 1000 (4-bit, 28-pin)	8,000	1974 <sup>TM</sup>	Texas Instruments	6,000 nm	11 nm²	
LSI Technology 1100 (8-bit, 40-pin)	4,000 <sup>TM</sup>	1975	LSI Technology	6,000 nm	21 nm²	
Intel 4004 (12-bit, 40-pin, alias of PDP-4)	4,000	1975	Intel	?	?	
DP 100 (8-bit, 24-pin, 40-pin)	5,000	1975	PIKA	?	?	
ICA 1600 (8-bit, 40-pin)	5,000	1976	PIKA	6,000 nm	27 nm²	
Zilog Z80 (8-bit, 40 ALU, 40-pin)	8,500 <sup>TM</sup>	1976	Zilog	4,000 nm	18 nm²	

24

---



---



---



---



---



---

1976-86						
						
Intel 8008 (8-bit, 40 pin)	6,000	1976	Intel	3,000 nm	20 mm <sup>2</sup>	
Intel 8008 (8-bit)	8,000	1976	Texas Instruments	7 nm	?	
Intel 8080 (8-bit)	7,000	1977	Bell Labs	5,000 nm	?	
Intel 8080 (8-bit with some 16-bit features, 40 pin)	9,000	1978	Motorola	5,000 nm	21 mm <sup>2</sup>	
Intel 8080 (8-bit, 40 pin)	20,000	1978	Intel	3,000 nm	30 mm <sup>2</sup>	
Intel 8080 (8-bit)	17,000 <sup>(1)</sup>	1979	Zilog	7 nm	?	
Intel 8080 (8-bit, 8081 data bus)	29,000	1979	Intel	3,000 nm	20 mm <sup>2</sup>	
Intel 8080 (8-bit, 8082 MM, 32 bit registers, 16-bit ALU)	68,000 <sup>(2)</sup>	1979	Motorola	3,000 nm	44 mm <sup>2</sup>	
Intel 8081 (8-bit, 40 pin)	55,000	1980	Intel	7 nm	?	
DEC 4010	11,500 <sup>(3)</sup>	1981	DEC	5,000 nm	6 mm <sup>2</sup>	
DEC 32400	40,000	1981	IBM	2,000 nm	?	
Intel 8085 (16-bit, 68 pin)	55,000	1981	Intel	2,000 nm	60 mm <sup>2</sup>	
Intel 8085 (16-bit, 95 pin)	134,000	1982	Intel	1,500 nm	40 mm <sup>2</sup>	
DEC 4010 (8-bit, 95 pin)	22,000 <sup>(4)</sup>	1983	DEC	3,000 nm <sup>(5)</sup>	9 mm <sup>2</sup>	
DEC V30	63,000	1984	NEC	7 nm	?	
Intel 8086 (16-bit, 80 pin, 114 pins select)	190,000 <sup>(6)</sup>	1984	Motorola	2,000 nm	60 mm <sup>2</sup>	
Intel 8086 (16-bit, 128 pin, no cache)	23,000	1985	Intel	2,000 nm	50 mm <sup>2</sup>	
IBM 3270 (no cache)	33,000 <sup>(7)</sup>	1985	Acorn	2,000 nm	40 mm <sup>2</sup>	
Atmel 16C416 (16-bit)	16,000 <sup>(8)</sup>	1986	Harris Corporation	3,000 nm <sup>(9)</sup>	?	
SPARC IC M86000 (32-bit, no cache)	110,000 <sup>(10)</sup>	1986	Fujitsu	1,200 nm	?	
DEC V30 (32-bit, no cache)	375,000	1986	NEC	1,500 nm	?	
IBM 3270 (32-bit, 84 pin, no cache)	27,000 <sup>(11)</sup>	1986	Acorn	2,000 nm	36.25 mm <sup>2</sup>	
Motorola 68000 (32-bit, small cache)	91,000	1986	Zilog	7 nm	?	

25

1987-94						
						
DEC V30P (32-bit, no cache)	380,000	1987	NEC	1,200 nm	?	
Hitachi GM32000 <sup>(12)</sup>	750,000	1987	Hitachi	1,000 nm	?	
Motorola 68000 (32-bit, very small cache)	273,000	1987	Motorola	800 nm	100 mm <sup>2</sup>	
Futuretek 32400 (no memory cache)	153,000 <sup>(13)</sup>	1987	Intel	1,000 nm	100 mm <sup>2</sup>	
DEC V30L (no memory cache)	160,000 <sup>(14)</sup>	1988	DEC	1,000 nm	61 mm <sup>2</sup>	
Intel 80486 (32-bit, 384 KB of cache)	250,000 <sup>(15)</sup>	1988	Intel	1,200 nm <sup>(16)</sup>	?	
IBM 80386 (32-bit, no cache, no MMU)	600,000 <sup>(17)</sup>	1989	Intel	800 nm	143 mm <sup>2</sup>	
Hitachi BH-1 (32-bit, no cache)	1,000,000 <sup>(18)</sup>	1989	Intel	7 nm	?	
Intel 80486DX (32-bit, 4 KB of cache)	1,180,235	1989	Intel	1,000 nm	125 mm <sup>2</sup>	
IBM 80386 (32-bit, 4 KB of cache)	31,000	1989	Acorn	1,000 nm	87 mm <sup>2</sup>	
Atmel 6500 (32-bit, 8 KB of cache)	1,200,000	1990	Motorola	450 nm	102 mm <sup>2</sup>	
4000 (64-bit, 16 KB of cache)	1,350,000	1991	MPFS	1,000 nm	213 mm <sup>2</sup>	
IBM 3280 (32-bit, no cache for the 80 variant)	35,000	1991	ARM	800 nm	?	
Hitachi BH-1 (32-bit, no cache)	650,000 <sup>(19)</sup>	1992 <sup>(20)</sup>	Hitachi	800 nm	10 mm <sup>2</sup>	
Intel 80486DX (32-bit, 1 MB of cache)	900,000 <sup>(21)</sup>	1992	Intel	7 nm	?	
DEC Alpha 21064 (32-bit, 200 pin, 16 KB of cache)	1,800,000	1992	DEC	750 nm	203.52 mm <sup>2</sup>	
Hitachi H480 (32-bit, 8 KB of cache)	2,800,000 <sup>(22)</sup>	1993	Hitachi	500 nm	267 mm <sup>2</sup>	
Intel 80486DX (32-bit, 16 KB of cache)	3,100,000	1993	Intel	800 nm	294 mm <sup>2</sup>	
IRMX 32 (32-bit, 16 KB of cache)	578,877 <sup>(23)</sup>	1994	ARM	700 nm	68.51 mm <sup>2</sup>	
Alpha21164 <sup>(24)</sup> (40 pin, includes video)	7,000 <sup>(25)</sup>	1994	Office Enterprises	1200 nm	?	
Motorola 68040 (32-bit, 16 KB of cache)	2,000,000	1994	Motorola	600 nm	218 mm <sup>2</sup>	
PowerPC 601 (32-bit, 32 KB of cache)	2,800,000 <sup>(26)</sup>	1994	Apple/Motorola	600 nm	121 mm <sup>2</sup>	

26

1995-2002						
						
4-110 (24-bit, 8 KB of cache)	2,500,000 <sup>(27)</sup>	1995	Atmel/DEC/Alpha	350 nm	90 mm <sup>2</sup>	
Intel 486 Pro (32-bit, 16 KB of cache), L2 cache on-package, but on separate board	5,500,000 <sup>(28)</sup>	1995	Intel	500 nm	327 mm <sup>2</sup>	
MD 20 (32-bit, cache)	4,300,000	1995	AMD	600 nm	251 mm <sup>2</sup>	
Hitachi SH-1 (32-bit, cache)	10,000,000 <sup>(29)</sup>	1997	Hitachi	200 nm <sup>(30)</sup>	42 mm <sup>2</sup>	
Intel 486DX4 (32-bit, 16 KB of cache, 144-bit SIMD, cache)	7,500,000	1997	Intel	500 nm	165 mm <sup>2</sup>	
MD 40 (32-bit, 16 KB of cache)	8,800,000	1997	AMD	350 nm	165 mm <sup>2</sup>	
21 (21-bit, includes a video)	15,000	1997 <sup>(31)</sup>	Orbitz Enterprises	?	?	
VR5 (8-bit, 4 KB, w/rom BIOS)	140,000 (48,000 est. memory) <sup>(32)</sup>	1997	Nordic VLSI/Alpha	?	?	
Intel 9 (Deschutes (32-bit, large cache))	7,500,000	1998	Intel	250 nm	113 mm <sup>2</sup>	
IRM ITAM (32-bit, no cache)	111,000 <sup>(33)</sup>	1998	Acorn	300 nm	4.8 mm <sup>2</sup>	
Intel 9 (Deschutes (32-bit, 16 KB of cache, SIMD, cache))	9,000,000	1998	Intel	250 nm	128 mm <sup>2</sup>	
MediaDirect (32-bit, 16 KB of cache)	13,500,000 <sup>(34)</sup>	1998	MediaDirect	250 nm	128 mm <sup>2</sup>	
Intel 9 (Media Direct (32-bit, cache))	37,400,000	1999	Intel	180 nm	180 mm <sup>2</sup>	
MD 40-II (32-bit, cache)	21,300,000	1999	AMD	250 nm	118 mm <sup>2</sup>	
MD 40-II (32-bit, cache)	22,000,000	1999	AMD	250 nm	184 mm <sup>2</sup>	
Becko (32-bit, large cache)	21,000,000 <sup>(35)</sup>	2000	IBM/Nintendo	180 nm	43 mm <sup>2</sup>	
Intel 9 (Coppermine (50-bit, large cache))	21,000,000	2000	Intel	180 nm	80 mm <sup>2</sup>	
Intel 9 (Coppermine (50-bit, large cache))	21,000,000	2000	Intel	180 nm	217 mm <sup>2</sup>	
PAW8240 (8-bit, large cache)	191,000,000 <sup>(36)</sup>	2001	Fujitsu	1,300 nm	128 mm <sup>2</sup>	
Intel 9 (Tulsa (32-bit, large cache))	45,000,000	2001	Intel	130 nm	81 mm <sup>2</sup>	
Intel 9 (Northwood (32-bit, large cache))	55,000,000	2002	Intel	130 nm	145 mm <sup>2</sup>	

27

2002-2007					
<b>GEBZE</b> Intel Inside®					
pentium 3 Mahoney (64-bit, large cache)	230,000,000	2002	Intel	180 nm	421 mm²
ICC Alpha 2120 (64-bit, 345-pn, SIMD, very large cache)	130,000,000 <sup>[7]</sup>	2002	DEC	180 nm	397 mm²
barton (32-bit, large cache)	54,300,000	2003	AMD	130 nm	101 mm²
MD 17 (64-bit, large cache)	105,900,000	2003	AMD	130 nm	165 mm²
duronto 2 Medon (64-bit)	410,000,000	2003	Intel	130 nm	374 mm²
duronto 4 Medon (64-bit, large cache)	112,000,000	2004	Intel	90 nm	112 mm²
SPARCIV 10 (64-bit, SIMD, cache)	400,000,000 <sup>[7]</sup>	2004	Fujitsu	90 nm	264 mm²
duronto 2 Risc-64 (64-bit, cache)	580,000,000	2004	Intel	130 nm	430 mm²
duronto 4 Precision-2M (32-bit, large cache)	169,000,000	2005	Intel	90 nm	143 mm²
duronto 5 Grinnell (32-bit, large cache)	238,000,000	2005	Intel	90 nm	206 mm²
jenos (64-bit, 128-bit SIMD, large cache)	165,000,000	2005	IBM	90 nm	7 mm²
all 128-bit cache	250,000,000 <sup>[7]</sup>	2005	Sony/IBM/Toshiba	90 nm	221 mm²
duronto 6 Durango (64-bit, large cache)	184,000,000	2005	Intel	90 nm	80 mm²
duronto 7 Preller (32-bit, large cache)	380,000,000	2006	Intel	65 nm	165 mm²
core 2 Duo Corvo (Dual-core 64-bit, large cache)	291,000,000	2006	Intel	65 nm	143 mm²
Multi-core Duran (2.64-bit, SIMD, large cache)	1,700,000,000 <sup>[7]</sup>	2006	Intel	90 nm	586 mm²
MD-K10 quad-core 2M (32-bit, large cache)	480,000,000 <sup>[7]</sup>	2007	AMD	65 nm	283 mm²
UMC Corner-Ap (32-bit, optional SIMD, cache)	26,000,000 <sup>[7]</sup>	2007	ARM	45 nm	31 mm²
core 2 Duo Wolfe (32-bit, SIMD, cache)	41,000,000	2007	Intel	45 nm	107 mm²
Core 2 Duo (32-bit, large cache)	780,000,000	2007	IBM	45 nm	341 mm²
core 2 Duo (32-bit, large cache)	180,000,000	2007	Intel	65 nm	111 mm²
Hyper	250,000,000 <sup>[7]</sup>	2007	Matsushita	45 nm	7 mm²
SPARCIV 10 (64-bit SIMD, large caches)	540,000,000	2007 <sup>[7]</sup>	Fujitsu	90 nm	421 mm²
duronto 7 Cache	1,000,000,000 <sup>[7]</sup>	2007	Intel	45 nm	400 mm²

28

2008-2012					
<b>GEBZE</b> Intel Inside®					
core 2 Duo Wolfe-SM (Dual-core 64-bit, SIMD, large cache)	290,000,000	2008	Intel	45 nm	83 mm²
core 1 (quad-core 64-bit, SIMD, large cache)	731,000,000	2008	Intel	45 nm	263 mm²
MD-K10 quad-core 900 (96-bit, SIMD, large cache)	788,000,000 <sup>[7]</sup>	2008	AMD	45 nm	258 mm²
core 2 Duo (32-bit, cache)	170,000,000	2008	Intel	45 nm	24 mm²
MD-K10 (32-bit, SIMD, large cache)	200,000,000	2008	AMD	45 nm	143 mm²
MD-K10 (32-bit, SIMD, large cache)	300,000,000	2008 <sup>[7]</sup>	Fujitsu	45 nm	140 mm²
Be-core Xeon T400 (32-bit, SIMD, large cache)	1,900,000,000	2008	Intel	45 nm	503 mm²
Be-core Opteron 3400 (64-bit, SIMD, large cache)	604,000,000	2009	AMD	45 nm	346 mm²
SPARCIV 10 (64-bit SIMD, large cache)	780,000,000 <sup>[7]</sup>	2009	Fujitsu	45 nm	513 mm²
Phenom T3 11-core 64-bit, SIMD, large cache)	1,000,000,000 <sup>[7]</sup>	2010	Sun/Oracle	40 nm	377 mm²
Be-cores 7 (Gulliver)	1,700,000,000	2010	Intel	32 nm	340 mm²
Phenom II X4 945 (32-bit, SIMD, large cache)	1,200,000,000	2010	IBM	32 nm	247 mm²
Dual-core i1100 <sup>[7]</sup> (64-bit, very large cache)	1,400,000,000	2010	IBM	45 nm	512 mm²
Dual-core i1100 <sup>[7]</sup> (64-bit, SIMD, large cache)	2,000,000,000 <sup>[7]</sup>	2010	Intel	65 nm	699 mm²
Ge-n Nehalem-EX (8-core 64-bit, SIMD, large cache)	2,000,000,000 <sup>[7]</sup>	2010	Intel	45 nm	684 mm²
SPARCIV 10 (64-bit SIMD, large cache)	1,870,000,000 <sup>[7]</sup>	2011	Fujitsu	40 nm	484 mm²
Dual-core + GPU Core 7 (32-bit, SIMD, large cache)	1,160,000,000	2011	Intel	32 nm	216 mm²
Dual-core + GPU Core 7 Cache	2,270,000,000 <sup>[7]</sup>	2011	Intel	32 nm	434 mm²
Be-Westmere-EX (10-core 64-bit, SIMD, large cache)	2,600,000,000	2011	Intel	32 nm	512 mm²
iron-Medfield (32-bit)	430,000,000 <sup>[7]</sup>	2012	Intel	32 nm	64 mm²
Phenom X4 945 (32-bit, SIMD, cache)	1,260,000,000 <sup>[7]</sup>	2012	Fujitsu	28 nm	600 mm²
MD-K10 quad-core 64-bit (32-bit, SIMD, cache)	1,800,000,000 <sup>[7]</sup>	2012	AMD	32 nm	311 mm²
Dual-core + GPU AMD Trinity (64-bit, SIMD, cache)	1,300,000,000	2012	AMD	32 nm	246 mm²
Dual-core + GPU Core 7 (32-bit, SIMD, cache)	1,400,000,000	2012	Intel	32 nm	160 mm²

29

2012-2015					
<b>GEBZE</b> Intel Inside®					
POWER7 V1 cores 94-bit (SIMD, 60 MB L3 cache)	2,100,000,000	2012	IBM	32 nm	607 mm²
Be-cores JEC12 (32-bit, SIMD, large cache)	2,790,000,000	2012	IBM	32 nm	597 mm²
saturn (Poulson) (8-core 64-bit, SIMD, cache)	3,100,000,000	2012	Intel	32 nm	644 mm²
Be-cores 7 (32-bit, SIMD, cache)	5,000,000,000 <sup>[7]</sup>	2012	Intel	22 nm	720 mm²
Be-cores 7 (64-bit, SIMD, cache)	2,000,000,000 <sup>[7]</sup>	2012	Apple	22 nm	100 mm²
Be-cores 7 (64-bit, 32-ARMv7, 64-bit SIMD, cache)	2,000,000,000 <sup>[7]</sup>	2013	Intel	22 nm	256 mm²
Be-cores C17 (Be-cores 64-bit 32-ARMv7, 64-bit SIMD, cache)	1,980,000,000	2013	Intel	22 nm	256 mm²
POWER8 V1 cores 94-bit (SIMD, cache)	4,200,000,000	2013	IBM	22 nm	650 mm²
Be-cores One Xeon 50 (64-bit, SIMD, cache)	5,000,000,000	2013	Microchip/IBM	28 nm	363 mm²
Dual-core + GPU Core 7 (32-bit, SIMD, cache)	1,400,000,000 <sup>[7]</sup>	2014	Intel	22 nm	177 mm²
Dual-core + GPU Core 7 Cache (32-bit, SIMD, cache)	4,000,000,000 <sup>[7]</sup>	2014	Intel	22 nm	387 mm²
Core i7 Haswell-E (8-core 64-bit, SIMD, cache)	2,000,000,000 <sup>[7]</sup>	2014	Intel	22 nm	566 mm²
spira ARK (8-core 64-bit 2 ARMv8 "mobile SoC", SIMD, cache)	3,000,000,000 <sup>[7]</sup>	2014	Apple	20 nm	128 mm²
Ge-n Braswell-EP (15-core 64-bit, SIMD, cache)	4,200,000,000 <sup>[7]</sup>	2014	Intel	22 nm	541 mm²
Core i7 Haswell-E (10-core 64-bit, SIMD, cache)	5,500,000,000 <sup>[7]</sup>	2014	Intel	22 nm	661 mm²
Be-cores C17 (10-core 64-bit, SIMD, cache)	1,790,000,000	2015	Intel	14 nm	122 mm²
Dual-core + GPU Core 7 (Be-cores 64-bit, SIMD, cache)	1,900,000,000 <sup>[7]</sup>	2015	Intel	14 nm	100 mm²
Marconi 49 (Marconi 64GB 32-ARMv8 "mobile SoC", SIMD, cache)	2,000,000,000+	2015	Apple	10 nm	96 mm² (Samsung) (Samsung)
spira ARK (dual core 64GB 32-ARMv8 "mobile SoC", SIMD, cache)	3,000,000,000+	2015	Apple	10 nm	104.5 mm² (TSMC) (TSMC)

30

2015-2017					
					
IBM z10 (64-bit, server)	5,980,000,000 <sup>(1)</sup>	2015	IBM	22 nm	678 mm <sup>2</sup>
IBM z10 Storage Controller	7,100,000,000	2015	IBM	22 nm	678 mm <sup>2</sup>
Qualcomm MT 220 (64-bit, SIMD, cache)	10,000,000,000 <sup>(2)</sup>	2015	Oracle	20 nm	?
Qualcomm Snapdragon 830 (octa-core 64/2-bit ARMv8 "mobile SoC", SIMD, cache)	3,000,000,000 <sup>(3)(4)</sup>	2016	Qualcomm	10 nm	72.3 mm <sup>2</sup>
Intel Broadwell (10-core 64-bit SIMD, cache)	5,200,000,000 <sup>(5)</sup>	2016	Intel	14 nm	246 mm <sup>2</sup> <sup>(6)</sup>
Apple A9 (quad-core 64-bit SIMD, cache)	3,300,000,000	2016	Apple	18 nm	125 mm <sup>2</sup>
Silicon Labs K90 (octa-core 64/2-bit ARMv8 "mobile SoC", SIMD, cache)	4,000,000,000 <sup>(7)</sup>	2016	Huawei	16 nm	110.0 mm <sup>2</sup>
Qualcomm SDM 625 (22-core 64/8-bit SIMD, cache)	7,200,000,000 <sup>(8)</sup>	2016	Intel	14 nm	458 mm <sup>2</sup>
Qualcomm SDM 630 (64-bit SIMD, cache)	8,000,000,000	2016	Intel	14 nm	663 mm <sup>2</sup>
Ty CPU 130 (64-bit SIMD, cache)	1,266 k-LUT <sup>(9)</sup>	2016	GigaPixel Technology	?	?
Qualcomm Snapdragon 845 (octa-core 64/2-bit ARMv8 "mobile SoC", SIMD, cache)	5,300,000,000 <sup>(10)</sup>	2017	Qualcomm	10 nm	94 mm <sup>2</sup>
Qualcomm Snapdragon 850 (octa-core 64/2-bit ARMv8 "mobile SoC", SIMD, cache)	5,300,000,000 <sup>(11)</sup>	2017	Qualcomm	10 nm	94 mm <sup>2</sup>
Apple A11 Fusion (hexa-core 64/2-bit ARMv8 "mobile SoC", SIMD, cache)	4,300,000,000	2017	Apple	10 nm	88.22 mm <sup>2</sup>
Qualcomm SDM 730 (hexa-core 64/2-bit ARMv8 "mobile SoC", SIMD, cache)	4,800,000,000 <sup>(12)</sup>	2017	AMD	14 nm	182 mm <sup>2</sup>
Qualcomm SDM 732 (hexa-core 64/2-bit SIMD, cache)	4,800,000,000 <sup>(13)</sup>	2017	AMD	14 nm	213 mm <sup>2</sup>
Qualcomm SDM 735 (hexa-core 64/2-bit SIMD, cache)	4,800,000,000 <sup>(14)</sup>	2017	AMD	14 nm	213 mm <sup>2</sup>
IBM z11 (64-bit SIMD, cache)	8,100,000,000	2017	IBM	14 nm	666 mm <sup>2</sup>
IBM z13 (64-bit SIMD, cache)	8,100,000,000	2017	IBM	14 nm	850 mm <sup>2</sup>

31

2017-2018					
					
Silicon Labs K97 (octa-core 64/2-bit ARMv8 "mobile SoC", SIMD, cache)	5,000,000,000 <sup>(15)</sup>	2017	Huawei	10 nm	96.72 mm <sup>2</sup>
Dove One X (Project Dove man SoC (64-bit SIMD, cache))	7,000,000,000 <sup>(16)</sup>	2017	Microsoft/AMD	10 nm	360 mm <sup>2</sup> <sup>(17)</sup>
Qualcomm SDM 730 (hexa-core 64/2-bit SIMD, cache)	8,000,000,000 <sup>(18)(19)</sup>	2017	Intel	14 nm	?
Qualcomm SDM 732 (hexa-core 64/2-bit SIMD, cache)	8,000,000,000	2017	IBM	14 nm	695 mm <sup>2</sup>
Freescale U200 Base Platform Chip (ZEN, 4x16/4) RISC-V (64-bit, cache)	250,000,000 <sup>(20)</sup>	2017	SiFive	28 nm	>80 mm <sup>2</sup>
PNASiK X1 (12-core 64-bit SIMD, cache)	5,400,000,000 <sup>(21)</sup>	2017	Fujitsu	28 nm	756 mm <sup>2</sup>
Qualcomm SDM 730 (hexa-core 64/2-bit ARMv8 "mobile SoC", SIMD, cache)	4,200,000,000 <sup>(22)</sup>	2017	Apple	10 nm	98.40 mm <sup>2</sup>
Sentry 2400 (64/2-bit SIMD, cache)	18,000,000,000 <sup>(23)</sup>	2017	Qualcomm	10 nm	398 mm <sup>2</sup>
AMD Ryzen 3 3200G (64-bit SIMD, cache)	18,000,000,000	2017	AMD	14 nm	765 mm <sup>2</sup>
Silicon Labs K90 (octa-core 64-bit ARMv8 "mobile SoC", SIMD, cache)	8,000,000,000 <sup>(24)</sup>	2018	Huawei	10 nm	?
Qualcomm SDM 660 (64-bit SIMD, cache)	8,000,000,000 <sup>(25)</sup>	2018	Apple	7 nm	83.27 mm <sup>2</sup>
Silicon Labs K90 (octa-core ARMv8 "mobile SoC", SIMD, cache)	8,000,000,000 <sup>(26)</sup>	2018	Huawei	7 nm	74.15 mm <sup>2</sup>
Qualcomm Snapdragon 8cx / (SCX9180 octa-core ARMv8 "mobile SoC", SIMD, cache)	8,000,000,000 <sup>(27)</sup>	2018	Qualcomm	7 nm	112 mm <sup>2</sup>
Qualcomm SDM 660 (64-bit SIMD, cache)	6,700,000,000 <sup>(28)</sup>	2019	Qualcomm	7 nm	73 mm <sup>2</sup>
Qualcomm Snapdragon 865 (octa-core 64/2-bit ARMv8 "mobile SoC", SIMD, cache)	10,300,000,000 <sup>(29)</sup>	2020	Qualcomm	7 nm	83.54 mm <sup>2</sup> <sup>(30)</sup>
Qualcomm SDM 660 (64-bit SIMD, cache)	10,000,000,000 <sup>(31)</sup>	2018	Apple	7 nm	122 mm <sup>2</sup>

32

2018-2021					
					
NVIDIA AAFX (64/2-bit SIMD, cache)	8,790,000,000 <sup>(32)</sup>	2018 <sup>(33)</sup>	Fujitsu	7 nm	?
NVIDIA Xavier SoC (8-bit)	9,000,000,000 <sup>(34)</sup>	2018	Nvidia	7 nm	350 mm <sup>2</sup>
AMD Ryzen 3 3200G (8-bit SIMD, cache, IO die)	5,800,000,000 <sup>(35)</sup>	2019	AMD	7 nm	12 nm (TSMC) 199 (7+125) mm <sup>2</sup>
Silicon Labs K90 4G	8,000,000,000 <sup>(36)</sup>	2019	Huawei	7 nm	90.0 mm <sup>2</sup>
Apple A13 (hexa-core 64-bit ARMv8 "mobile SoC", SIMD, cache)	8,500,000,000 <sup>(37)(38)</sup>	2019	Apple	7 nm	58.40 mm <sup>2</sup>
AMD Ryzen 3 3200G (8-bit SIMD, cache, IO die)	9,800,000,000 <sup>(39)</sup>	2019	AMD	7 nm	273 mm <sup>2</sup> (TSMC)
Silicon Labs K90 900 5G	10,300,000,000 <sup>(40)</sup>	2019	Huawei	7 nm	113.31 mm <sup>2</sup>
Wolfs Gravure 19-bit, 64-core ARM-based, SIMD, cache	30,000,000,000	2019	Amazon	7 nm	?
AMD Epyc Rome (64-bit SIMD, cache)	39,540,000,000 <sup>(41)</sup>	2019	AMD	7 nm	12 nm (TSMC) 1008 mm <sup>2</sup>
Jasmin TOHAKA (ARM A72, DSP, SRAM)	3,200,000,000 <sup>(42)</sup>	2020	Texas Instruments	16 nm	?
Apple A14 Bionic (hexa-core 64-bit ARMv8 "mobile SoC", SIMD, cache)	11,600,000,000 <sup>(43)</sup>	2020	Apple	5 nm	88 mm <sup>2</sup>
Apple M1 (octa-core 64-bit ARMv8 "mobile SoC", SIMD, cache)	16,000,000,000 <sup>(44)</sup>	2020	Apple	5 nm	119 mm <sup>2</sup>
Silicon Labs K9000	15,300,000,000 <sup>(45)(46)</sup>	2020	Huawei	5 nm	114 mm <sup>2</sup>
Apple A15	15,000,000,000 <sup>(47)</sup>	2021	Apple	5 nm	?
AMD Ryzen 7 5800H (8-bit SIMD, cache, IO and GPU)	10,700,000,000 <sup>(48)</sup>	2021	AMD	5 nm	180 mm <sup>2</sup>

33



WHERE TO GO?

www.gsu.edu.tr

The AND operation			
AND $\begin{array}{ccccc} 0 & & 0 & & 0 \\ 0 & \text{AND} & 1 & = & 0 \\ \hline 0 & & 0 & & 0 \end{array}$	AND $\begin{array}{ccccc} 0 & & 1 & & 1 \\ 1 & \text{AND} & 0 & = & 0 \\ \hline 1 & & 0 & & 0 \end{array}$	AND $\begin{array}{ccccc} 1 & & 0 & & 1 \\ 0 & \text{AND} & 1 & = & 0 \\ \hline 0 & & 1 & & 1 \end{array}$	AND $\begin{array}{ccccc} 1 & & 1 & & 1 \\ 1 & \text{AND} & 1 & = & 1 \\ \hline 1 & & 1 & & 1 \end{array}$
The OR operation			
OR $\begin{array}{ccccc} 0 & & 0 & & 0 \\ 0 & \text{OR} & 1 & = & 1 \\ \hline 0 & & 1 & & 1 \end{array}$	OR $\begin{array}{ccccc} 0 & & 1 & & 1 \\ 1 & \text{OR} & 0 & = & 1 \\ \hline 1 & & 0 & & 1 \end{array}$	OR $\begin{array}{ccccc} 1 & & 0 & & 1 \\ 0 & \text{OR} & 1 & = & 1 \\ \hline 1 & & 1 & & 1 \end{array}$	OR $\begin{array}{ccccc} 1 & & 1 & & 1 \\ 1 & \text{OR} & 1 & = & 1 \\ \hline 1 & & 1 & & 1 \end{array}$
The XOR operation			
XOR $\begin{array}{ccccc} 0 & & 0 & & 0 \\ 0 & \text{XOR} & 1 & = & 1 \\ \hline 0 & & 1 & & 1 \end{array}$	XOR $\begin{array}{ccccc} 0 & & 1 & & 1 \\ 1 & \text{XOR} & 0 & = & 1 \\ \hline 1 & & 0 & & 1 \end{array}$	XOR $\begin{array}{ccccc} 1 & & 0 & & 1 \\ 0 & \text{XOR} & 1 & = & 1 \\ \hline 1 & & 1 & & 0 \end{array}$	XOR $\begin{array}{ccccc} 1 & & 1 & & 0 \\ 1 & \text{XOR} & 1 & = & 0 \\ \hline 0 & & 0 & & 0 \end{array}$

34

---



---



---



---



---



---



---



---

AND				OR			
Inputs				Inputs			
Inputs	Output	Inputs	Output	Inputs	Output	Inputs	Output
0 0	0	0 0	0	0 0	0	0 1	1
0 1	0	0 1	1	1 0	1	1 0	1
1 0	0	1 0	1	1 1	1	1 1	1
1 1	1	1 1	1				

35

---



---



---



---



---



---



---



---

XOR				NOT			
Inputs				Inputs			
Inputs	Output	Inputs	Output	Inputs	Output	Inputs	Output
0 0	0	0 1	1	0	1	1 0	0
0 1	1	1 0	1	1	0	1 1	0
1 0	1	1 1	0				
1 1	0						

36

---



---



---



---



---



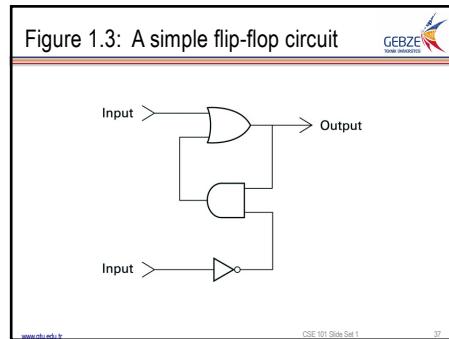
---



---



---



37

---

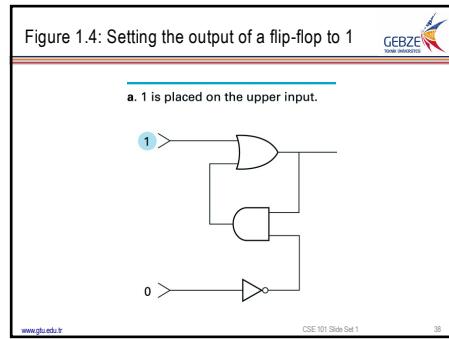
---

---

---

---

---



38

---

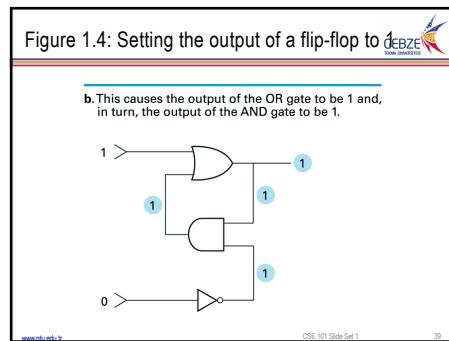
---

---

---

---

---



39

---

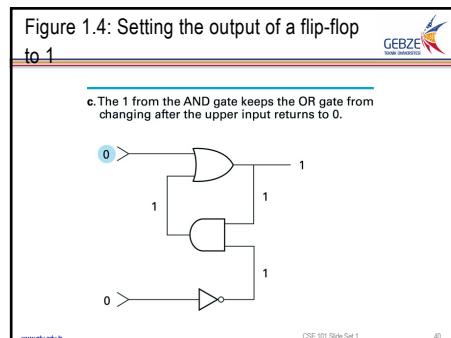
---

---

---

---

---



40

---



---



---



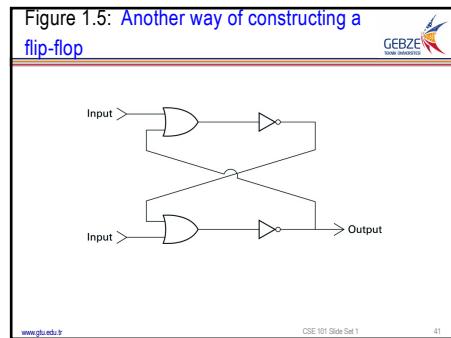
---



---



---



41

---



---



---



---



---



---

Figure 1.6: The hexadecimal coding system

Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

www.gsu.edu.tr CSE 101 Slide Set 1 42

42

---



---



---



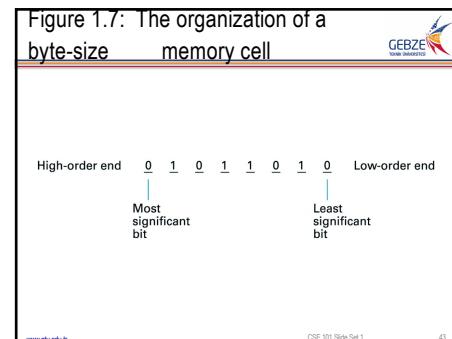
---



---



---



43

---

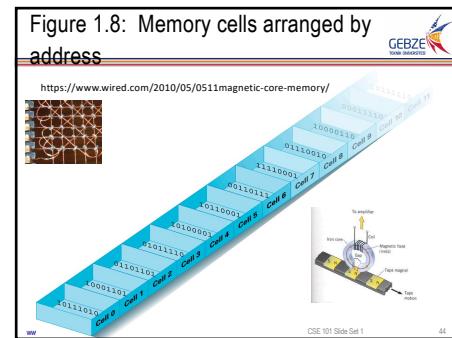
---

---

---

---

---



44

---

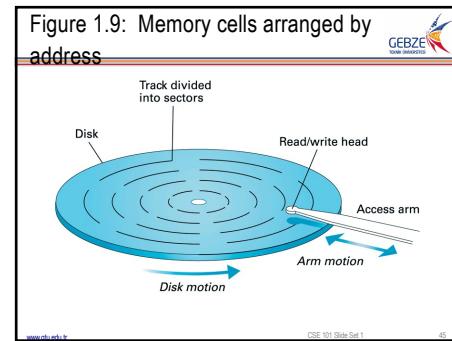
---

---

---

---

---



45

---

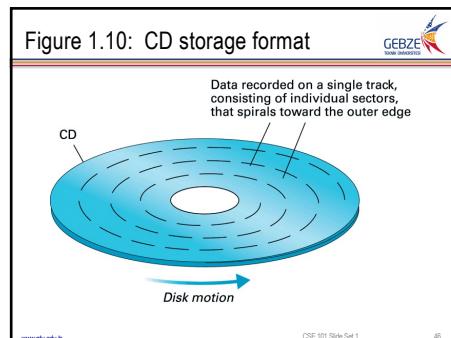
---

---

---

---

---



46

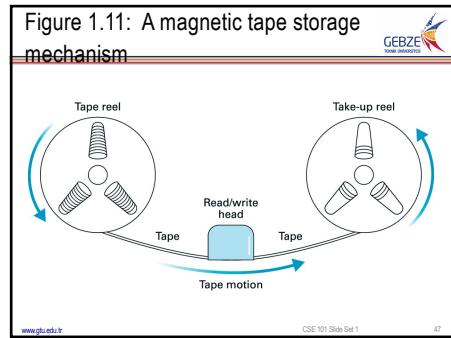
---

---

---

---

---



47

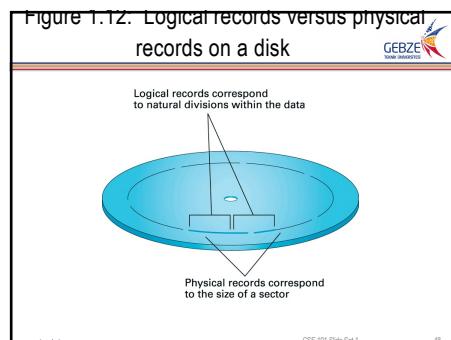
---

---

---

---

---



48

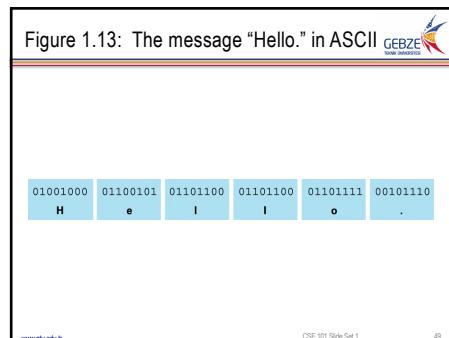
---

---

---

---

---



49

---



---



---



---



---



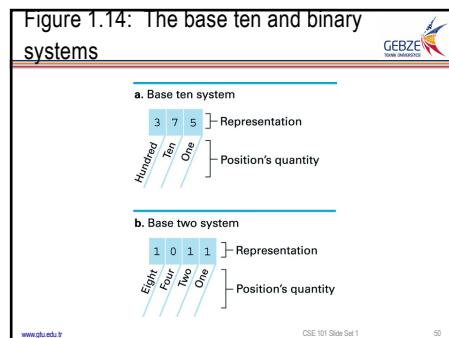
---



---



---



50

---



---



---



---



---



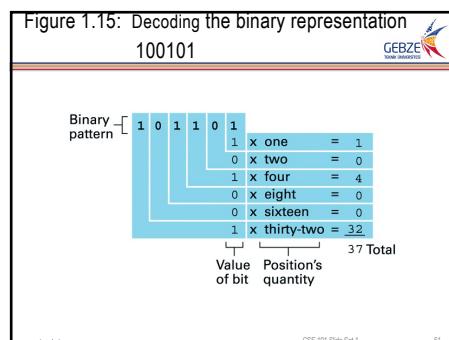
---



---



---



51

---



---



---



---



---



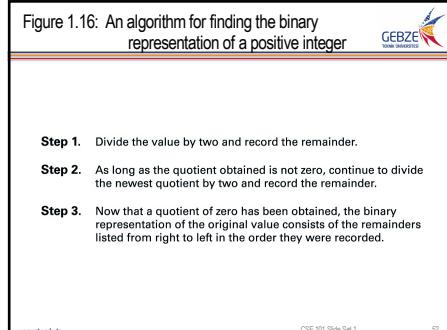
---



---



---



52

---



---



---



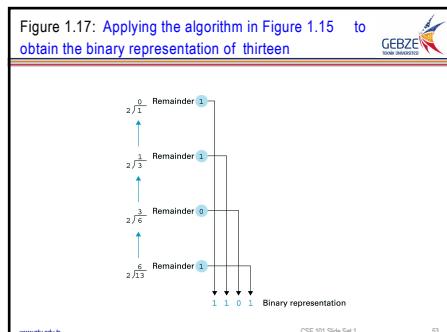
---



---



---



53

---



---



---



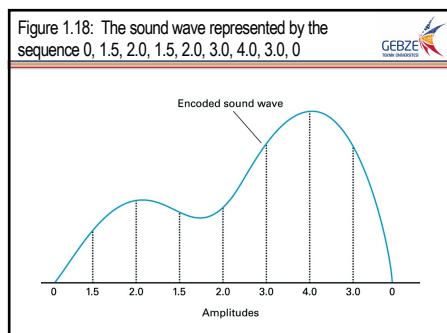
---



---



---



54

---



---



---



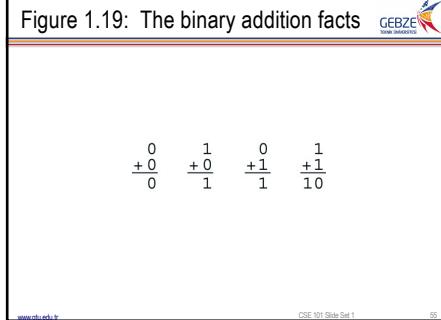
---



---



---



55

---

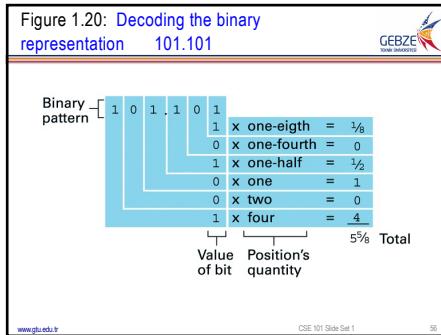
---

---

---

---

---



56

---

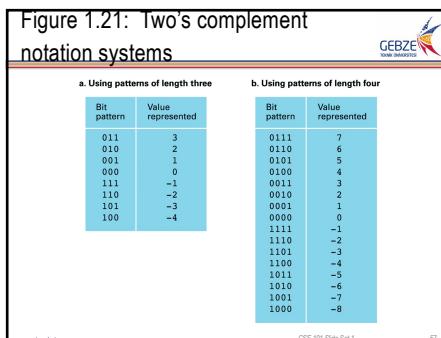
---

---

---

---

---



57

---

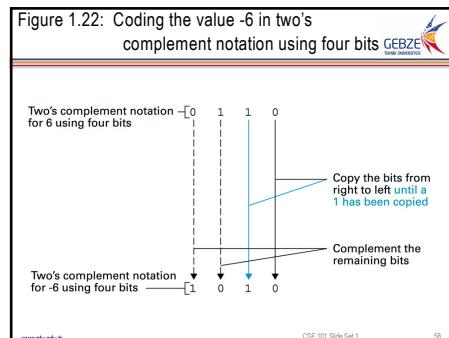
---

---

---

---

---



58

---



---



---



---



---



---

Figure 1.23: Addition problems converted to two's complement notation

Problem in base ten	Problem in two's complement	Answer in base ten
$3 + 2$	$\begin{array}{r} 0011 \\ + 0010 \\ \hline 0101 \end{array}$	5
$-3 + -2$	$\begin{array}{r} 1101 \\ + 1110 \\ \hline 1011 \end{array}$	-5
$7 + -5$	$\begin{array}{r} 0111 \\ + 1011 \\ \hline 0010 \end{array}$	2

59

---



---



---



---



---



---

Figure 1.24: An excess eight conversion table

Bit pattern	Value represented
1111	7
1110	6
1101	5
1100	4
1011	3
1010	2
1001	1
1000	0
0111	-1
0110	-2
0101	-3
0100	-4
0011	-5
0010	-6
0001	-7
0000	-8

60

---



---



---



---



---



---

Figure 1.25: An excess notation system using bit patterns of length three



Bit pattern	Value represented
111	3
110	2
101	1
100	0
011	-1
010	-2
001	-3
000	-4

www.ogu.edu.tr CSE 101 Slide Set 1 61

61

---



---



---



---



---



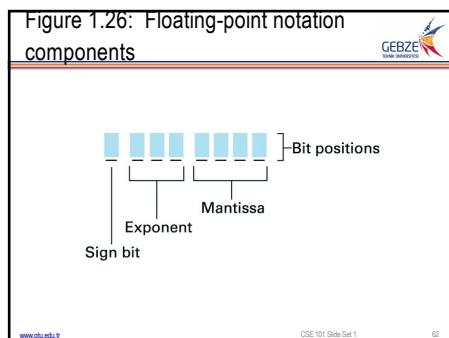
---



---



---



62

---



---



---



---



---



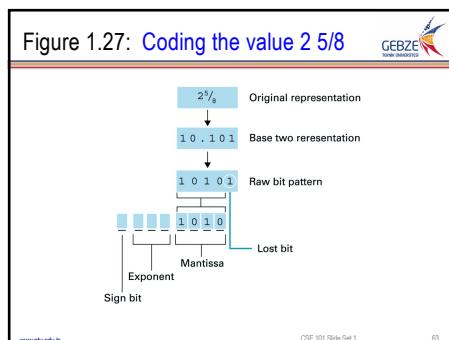
---



---



---



63

---



---



---



---



---



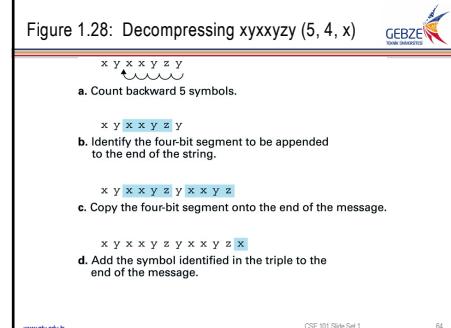
---



---



---



64

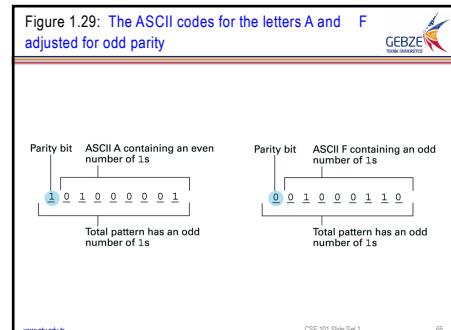
---

---

---

---

---



65

---

---

---

---

---

**Figure 1.30: An error-correcting code**

Symbol	Code
A	0000000
B	0011110
C	0100011
D	0111000
E	1000110
F	1010011
G	1101011
H	1110101

www.cse.bilkent.edu.tr CSE 101 Slide Set 1 66

66

---

---

---

---

---

Figure 1.31: Decoding the pattern 010100 using the code in Figure 1.30

Character	Distance between the received pattern and the character being considered
A	2
B	4
C	3
D	① Smallest distance
E	3
F	5
G	2
H	4

67

---



---



---



---



---



---



---



---



68

---



---



---



---



---



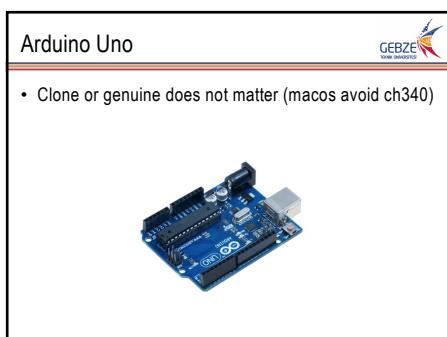
---



---



---



69

---



---



---



---



---



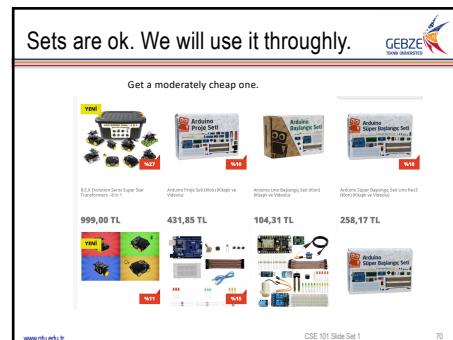
---



---



---



70

---

---

---

---

---

---



71

---

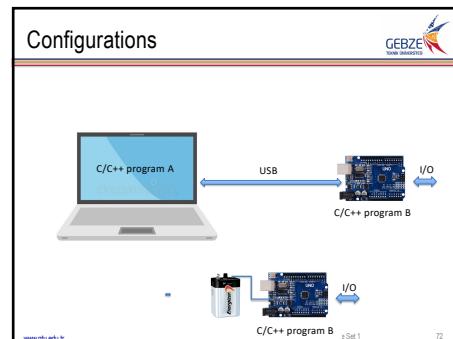
---

---

---

---

---



72

---

---

---

---

---

---

## Chapter 2: Data Manipulation

- 2.1 Computer Architecture
- 2.2 Machine Language
- 2.3 Program Execution
- 2.4 Arithmetic/Logic Instructions
- 2.5 Communicating with Other Devices
- 2.6 Program Data Manipulation
- 2.7 Other Architectures

73

---



---



---



---



---



---



---

## Computer Architecture

- Central Processing Unit (CPU) or processor
  - Arithmetic/Logic unit versus Control unit
  - Registers
    - General purpose
    - Special purpose
- Bus
- Motherboard

74

---



---



---



---



---

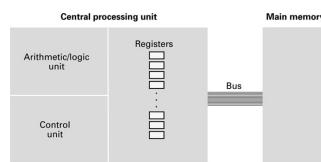


---



---

Figure 2.1 CPU and main memory connected via  
a bus



75

---



---



---



---



---



---



---

## Stored Program Concept



A program can be encoded as bit patterns and stored in main memory. From there, the CPU can then extract the instructions and execute them. In turn, the program to be executed can be altered easily.

76

---



---



---



---



---



---

## Terminology



- **Machine instruction:** An instruction (or command) encoded as a bit pattern recognizable by the CPU
- **Machine language:** The set of all instructions recognized by a machine

77

---



---



---



---



---



---

## Machine Language Philosophies



- Reduced Instruction Set Computing (RISC)
  - Few, simple, efficient, and fast instructions
  - Examples: PowerPC from Apple/IBM/Motorola and ARM
- Complex Instruction Set Computing (CISC)
  - Many, convenient, and powerful instructions
  - Example: Intel

78

---



---



---



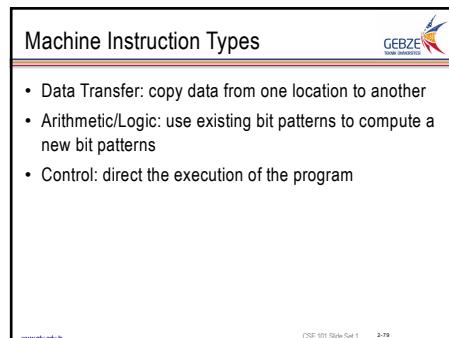
---



---



---



79

---



---



---



---



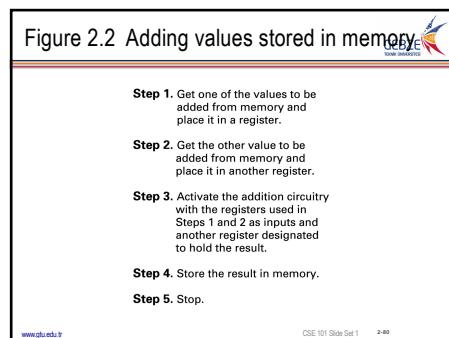
---



---



---



80

---



---



---



---



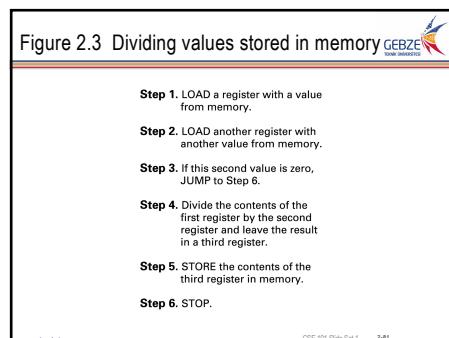
---



---



---



81

---



---



---



---



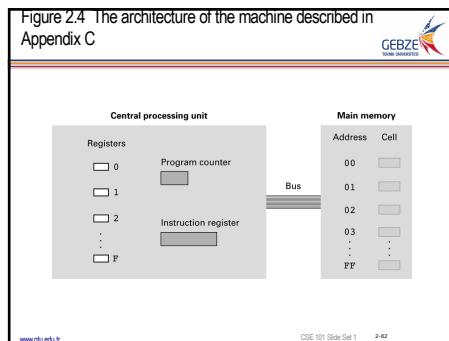
---



---



---



82

---

---

---

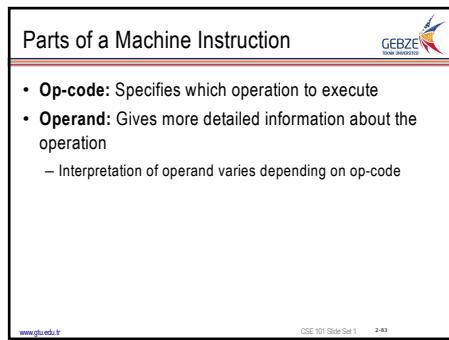
---

---

---

---

---



83

---

---

---

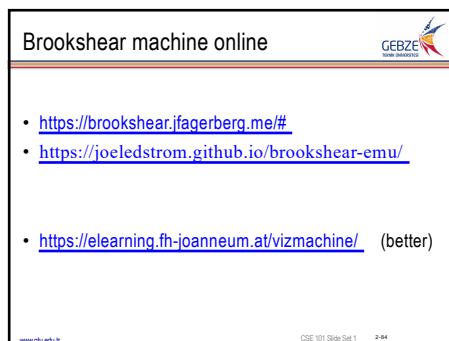
---

---

---

---

---



84

---

---

---

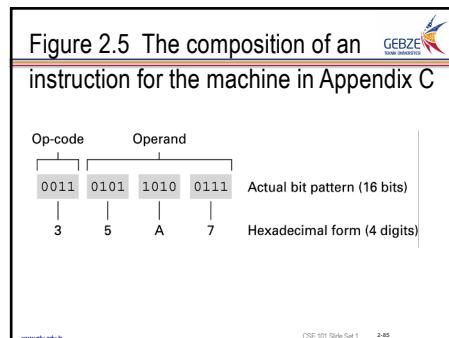
---

---

---

---

---



85

---

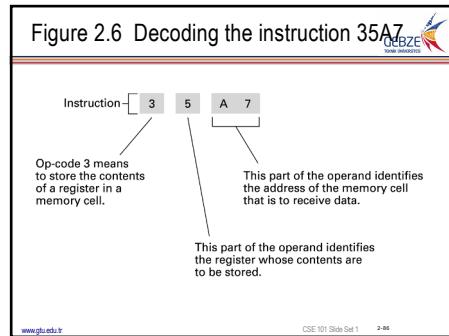
---

---

---

---

---



86

---

---

---

---

---

---

**Figure 2.7 An encoded version of the instructions in Figure 2.2**

Encoded instructions	Translation
156C	Load register 5 with the bit pattern found in the memory cell at address 6C.
166D	Load register 6 with the bit pattern found in the memory cell at address 6D.
5056	Add the contents of register 5 and 6 as though they were two's complement representations and leave the result in register 0.
306E	Store the contents of register 0 in the memory cell at address 6E.
C000	Halt.

www.cse.bilkent.edu.tr      GEBZE teknolojisi

CSE 101 Slide Set 1      2-47

87

---

---

---

---

---

---

**Program Execution**

- Controlled by two special-purpose registers
  - Program counter: address of next instruction
  - Instruction register: current instruction
- Machine Cycle
  - Fetch
  - Decode
  - Execute

www.gsu.edu.tr GEBZE  
Yıldız Teknik Üniver.

CSE 101 Slide Set 1 2-48

88

---



---



---



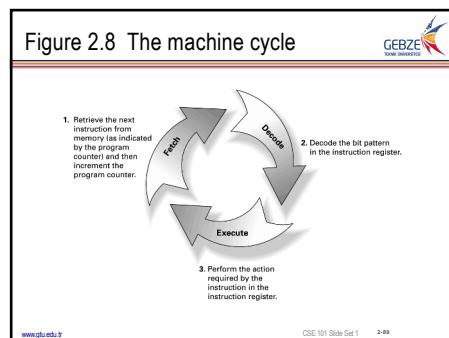
---



---



---



89

---



---



---



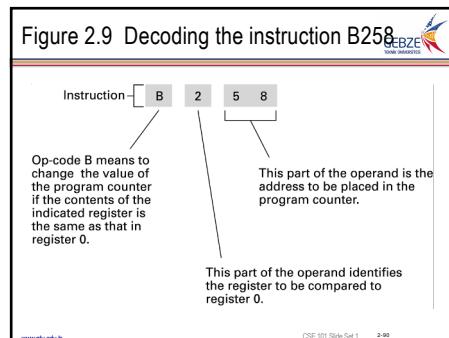
---



---



---



90

---



---



---



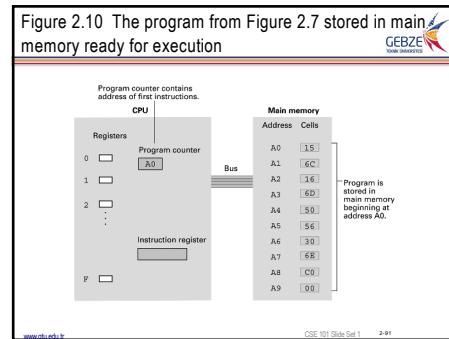
---



---



---



91

---



---



---



---



---



---



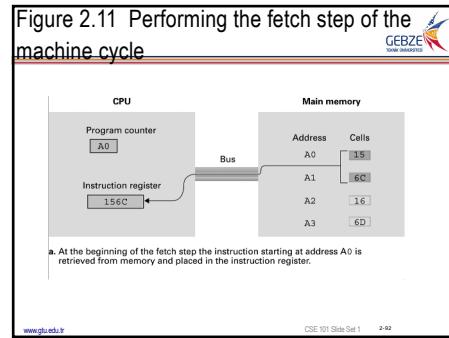
---



---



---



92

---



---



---



---



---



---



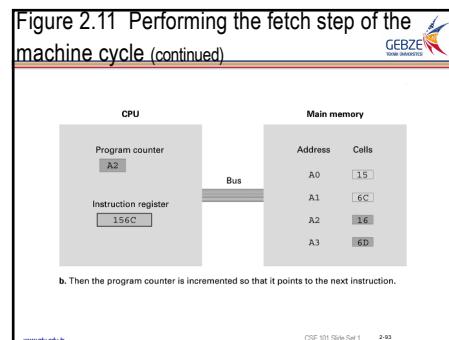
---



---



---



93

---



---



---



---



---



---



---



---



---

**Arithmetic/Logic Operations**

- Logic: AND, OR, XOR
  - Masking
- Rotate and Shift: circular shift, logical shift, arithmetic shift
- Arithmetic: add, subtract, multiply, divide
  - Precise action depends on how the values are encoded (two's complement versus floating-point).

www.cse.bilkent.edu.tr GEBZE  
Yıldız İkinci Yılı

CSE 101 Slide Set 1 2-94

94

---



---



---



---



---



---



---



---

**Figure 2.12 Rotating the bit pattern 65 (hexadecimal) one bit to the right**

The original bit pattern  
The bits move one position to the right. The rightmost bit shifts out the end and is placed in the hole at the other end.  
The final bit pattern

www.cse.bilkent.edu.tr GEBZE  
Yıldız İkinci Yılı

CSE 101 Slide Set 1 2-95

95

---



---



---



---



---



---



---



---

**Communicating with Other Devices**

- **Controller:** An intermediary apparatus that handles communication between the computer and a device
  - Specialized controllers for each type of device
  - General purpose controllers (USB and FireWire)
- **Port:** The point at which a device connects to a computer
- **Memory-mapped I/O:** CPU communicates with peripheral devices as though they were memory cells

www.cse.bilkent.edu.tr GEBZE  
Yıldız İkinci Yılı

CSE 101 Slide Set 1 2-96

96

---



---



---



---



---



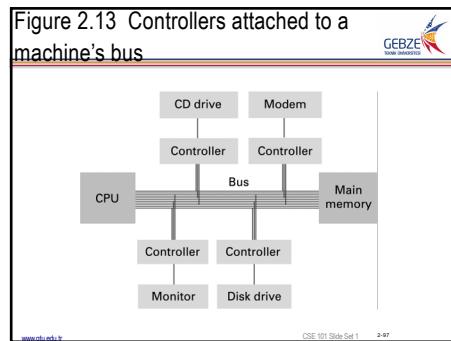
---



---



---



97

---

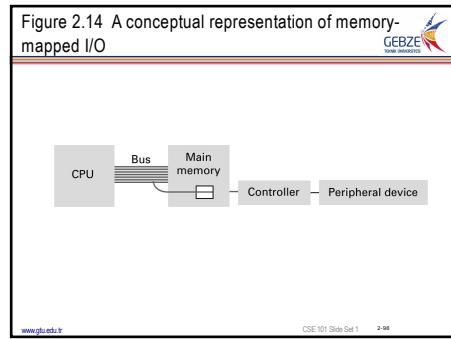
---

---

---

---

---



98

---

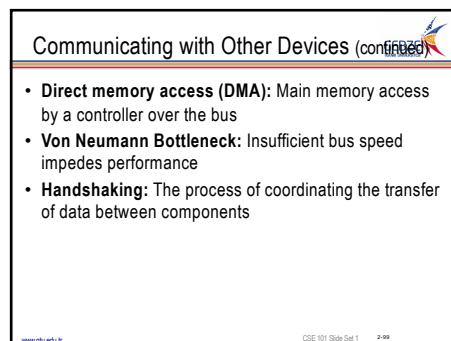
---

---

---

---

---



99

---

---

---

---

---

---

### Communicating with Other Devices (continued)

- **Parallel Communication:** Several communication paths transfer bits simultaneously.
- **Serial Communication:** Bits are transferred one after the other over a single communication path.

100

---



---



---



---



---



---

### Data Communication Rates

- Measurement units
  - Bps: Bits per second
  - Kbps: Kilo-bps (1,000 bps)
  - Mbps: Mega-bps (1,000,000 bps)
  - Gbps: Giga-bps (1,000,000,000 bps)
- Bandwidth: Maximum available rate

101

---



---



---



---



---



---

### Programming Data Manipulation

- Programming languages shields users from details of the machine:
  - A single Python statement might map to one, tens, or hundreds of machine instructions
  - Programmer does not need to know if the processor is RISC or CISC
  - Assigning variables surely involves LOAD, STORE, and MOVE op-codes

102

---



---



---



---



---



---

### Bitwise Problems as Python Code

```
print(bin(0b10011010 & 0b11001001)
# Prints '0b10001000'

print(bin(0b10011010 | 0b11001001)
# Prints '0b11011011'

print(bin(0b10011010 ^ 0b11001001)
# Prints '0b10100111'
```

We will not be using PYTHON in  
this course...

103

---



---



---



---



---



---

### Control Structures

- If statement:

```
if (water_temp > 140):
    print('Bath water too hot!')
```

- While statement:

```
while (n < 10):
    print(n)
    n = n + 1
```

104

---



---



---



---



---



---

### Functions

- Function:** A name for a series of operations that should be performed on the given parameter or parameters
- Function call:** Appearance of a function in an expression or statement

```
x = 1034
y = 1056
z = 2078
biggest = max(x, y, z)
print(biggest) # Prints '2078'
```

105

---



---



---



---



---



---

### Functions (continued)



- **Argument Value:** A value plugged into a parameter
  - **Fruitful functions return a value**
  - **void functions, or procedures,** do not return a value
- ```
sideA = 3.0
sideB = 4.0
# Calculate third side via Pythagorean Theorem
hypotenuse = math.sqrt(sideA**2 + sideB**2)
print(hypotenuse)
```

106

---



---



---



---



---



---



---

### Input / Output



```
# Calculates the hypotenuse of a right triangle
import math

# Inputting the side lengths, first try
sideA = int(input('Length of side A? '))
sideB = int(input('Length of side B? '))

# Calculate third side via Pythagorean Theorem
hypotenuse = math.sqrt(sideA**2 + sideB**2)
print(hypotenuse)
```

107

---



---



---



---



---



---



---



---

### Marathon Training Assistant



```
# Marathon training assistant.

import math

# This function converts a number of minutes and
# seconds into just seconds.
def total_seconds(min, sec):
    return min * 60 + sec

# This function calculates a speed in miles per hour given
# a time (in seconds) to run a single mile.
def speed(time):
    return 3600 / time
```

108

---



---



---



---



---



---



---



---

Marathon Training Assistant (continued) GEBZE TEKNOLOJİLERİ

```

# Prompt user for pace and mileage.
pace_minutes = int(input('Minutes per mile? '))
pace_seconds = int(input('Seconds per mile? '))
miles = int(input('Total miles? '))

# Calculate and print speed.
mph = speed(total_seconds(pace_minutes, pace_seconds))
print('Your speed is ' + str(mph) + ' mph')

# Calculate elapsed time for planned workout.
total = miles * total_seconds(pace_minutes, pace_seconds)
elapsed_minutes = total // 60
elapsed_seconds = total % 60
print('Your elapsed time is ' + str(elapsed_minutes) +
      ' mins ' + str(elapsed_seconds) + ' secs')

```

www.gtu.edu.tr CSE 101 Slide Set 1 2-109

109

---



---



---



---



---



---



---



---

Figure 2.15 Example Marathon Training GEBZE TEKNOLOJİLERİ

Data

| Time Per Mile |         |       | Total Elapsed Time |         |         |
|---------------|---------|-------|--------------------|---------|---------|
| Minutes       | Seconds | Miles | Speed (mph)        | Minutes | Seconds |
| 9             | 14      | 5     | 6.49819494584      | 46      | 10      |
| 8             | 0       | 3     | 7.5                | 24      | 0       |
| 7             | 45      | 6     | 7.74193548387      | 46      | 30      |
| 7             | 25      | 1     | 8.08988764044      | 7       | 25      |

www.gtu.edu.tr CSE 101 Slide Set 1 2-110

110

---



---



---



---



---



---



---



---

- Other Architectures GEBZE TEKNOLOJİLERİ
- Technologies to increase throughput:
    - Pipelining: Overlap steps of the machine cycle
    - Parallel Processing: Use multiple processors simultaneously
      - SISD: No parallel processing
      - MIMD: Different programs, different data
      - SIMD: Same program, different data
- www.gtu.edu.tr CSE 101 Slide Set 1 2-111

111

---



---



---



---



---



---



---



---