Introduction to Microcontroller

Objectives

- Number Systems
- Computer Architecture
- CPU, Microprocessor and Microcontroller
- AVR Microcontrollers
- ATmega328PB Microcontroller Registers

Number Systems (1)

- Binary number Systems
  - 0 and 1
  - Used inside computer system
  - Suffix: 'B' or 'b'
- Decimal number Systems
  - 0 through 9
- Hexadecimal number Systems
  - 0 through 9, and 'A' through 'F'
  - Suffix: 'H' or 'h'

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Number Systems (2)

- Convert a decimal to binary number

  Example: Convert 25_{10} to binary number

	Quotient	Remainder	
25/2 =	12	1	LSB
12/2 =	6	0	
6/2 =	3	0	
3/2 =	1	1	
1/2 =	0	1	MSB

$25_{10} = 11001b$

Number Systems (3)

- Convert a binary to decimal number

  Example: Convert 11001b to decimal number

Weight	2^4=16	2^3=8	2^2=4	2^1=2	2^0=1	
Digit	1	1	0	0	1	
Sum	16 +	8 +	0 +	0 +	1	= 25

$11001b = 25_{10}$

Number Systems (4)

- Convert a binary to hexadecimal number

  Example: Convert 100111110101b to hexadecimal number

  1. Make 4-bit groups from the decimal point

     1001 1111 0101

  2. Convert each 4-bit binary number group to hexadecimal

     1001 1111 0101

      9 F 5

     100111110101b = 9F5h

Number Systems (5)

- Binary addition and subtraction

A + B	Carry	Sum
0 + 0	0	0
0 + 1	0	1
1 + 0	0	1
1 + 1	1	0

A - B	Borrow	Difference
0 - 0	0	0
0 - 1	1	1
1 - 0	0	1
1 - 1	0	1

Number Systems (6)

- 1's complement

  Convert 1 to 0, and 0 to 1

  EX: 1's complement of 1001b = 0110b

```
  1   0   0   1
  ↓   ↓   ↓   ↓
  0   1   1   0
```

- 2's complement

  Add 1 to 1's complement

  EX: 2's complement of 1001b = 0111b

```
  0   1   1   0
+                 1
-----------------------
  0   1   1   1
```

Number Systems (7)

- BCD (Binary-coded Decimal)
  - A digital encoding method for decimal numbers
  - Each digit is represented by its own binary sequence
  - A numeral is usually represented by four bits
  - Represents the decimal range 0 through 9

Decimal	0	1	2	3	4	5	6	7	8	9
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Number Systems (8)

- Unpacked BCD

  EX: 91

Decimal	9	1
Binary	0000 1001	0000 0001

2 bytes

- Packed BCD

  EX: 91

Decimal	9 1
Binary	1001 0001

1 byte

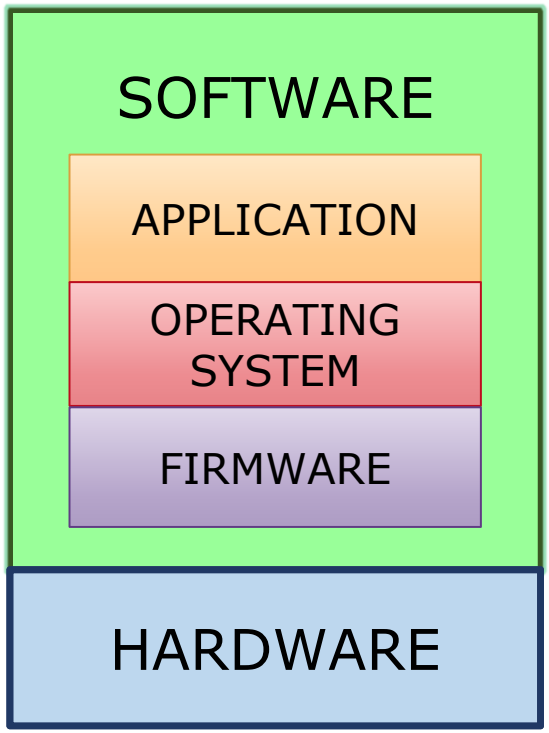
Number Systems (9)

- ASCII

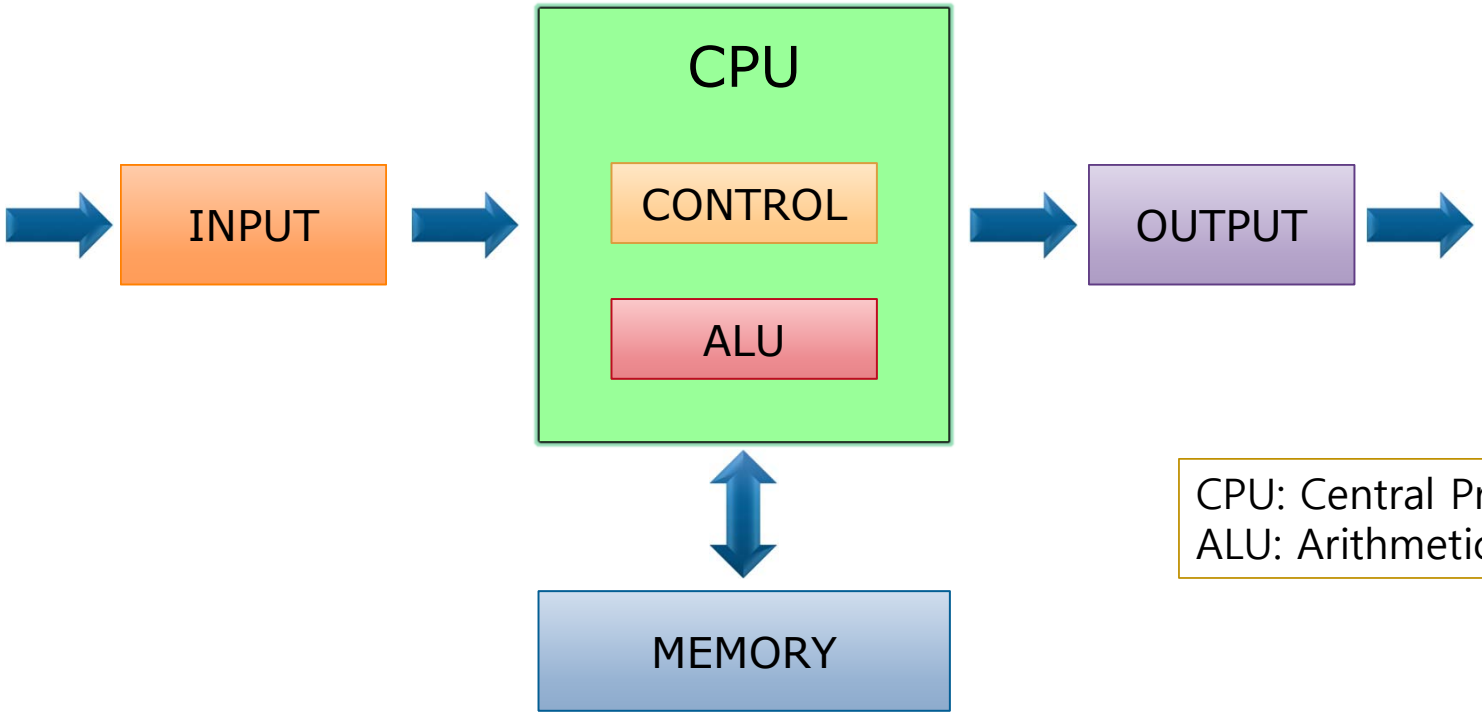
  American **S**tandard **C**ode for **I**nformation **I**nterchange

b_7 →					0	0	0	0	1	1	1	1
b_6 →					0	0	1	1	0	0	1	1
b_5 →					0	1	0	1	0	1	0	1
Bits	b_4 ↓	b_3 ↓	b_2 ↓	b_1 ↓	Column → / Row ↓	0	1	2	3	4	5	6	7
	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
	0	0	1	0	2	STX	DC2	"	2	B	R	b	r
	0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
	0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
	0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
	0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
	1	0	0	0	8	BS	CAN	(	8	H	X	h	x
	1	0	0	1	9	HT	EM	)	9	I	Y	i	y
	1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
	1	0	1	1	11	VT	ESC	+	;	K	[	k	{
	1	1	0	0	12	FF	FC	,	<	L	\	l	|
	1	1	0	1	13	CR	GS	-	=	M	]	m	}
	1	1	1	0	14	SO	RS	.	>	N	^	n	~
	1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Computer Organization



Computer Hardware



CPU: Central Processing Unit
ALU: Arithmetic Logic Unit

Computer Software (1)

- Program
  - A set of instructions that the computer hardware can execute
- Machine Instruction / Machine Language
  - All programs are stored in the computer's memory in the form of binary number (machine instruction)
  - It is difficult and not productive

- Ex: Machine language program
  - 1001 0100 0001 0011b

    stands for "increment the contents of R1 register by 1."
  - 0000 1100 0010 0001b

    stands for "add the contents of R1 register to the contents of R2 register."

Computer Software (2)

- Assembly Language
  - Invented to simplify the programming
  - Consists of assembly instructions
  - Mnemonic representation of a machine instruction

- Ex: Assembly language program
  - INC R1

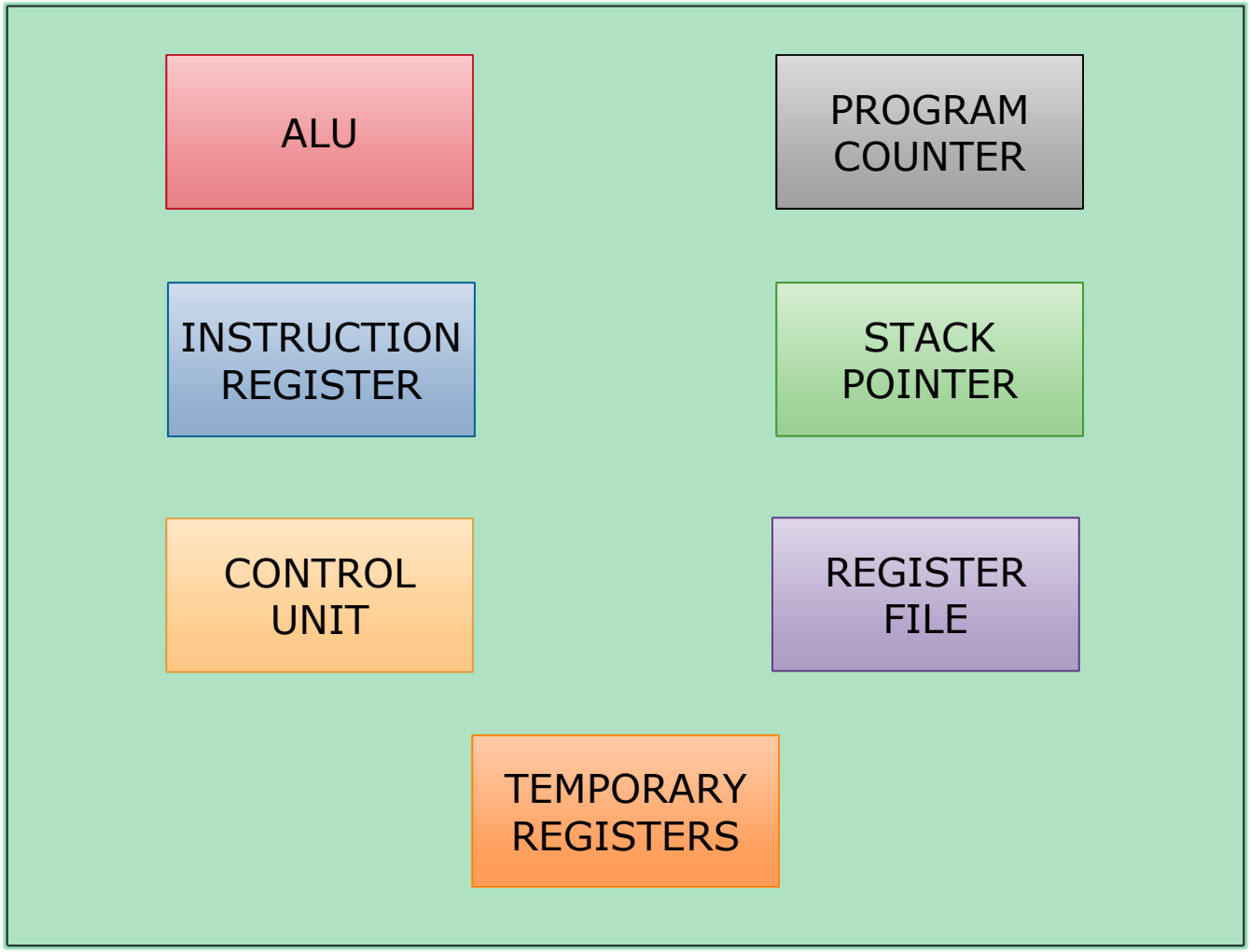
    stands for “increment the contents of R1 register by 1.”
  - ADD R2, R1

    stands for “add the contents of R1 register to the contents of R2 register.”

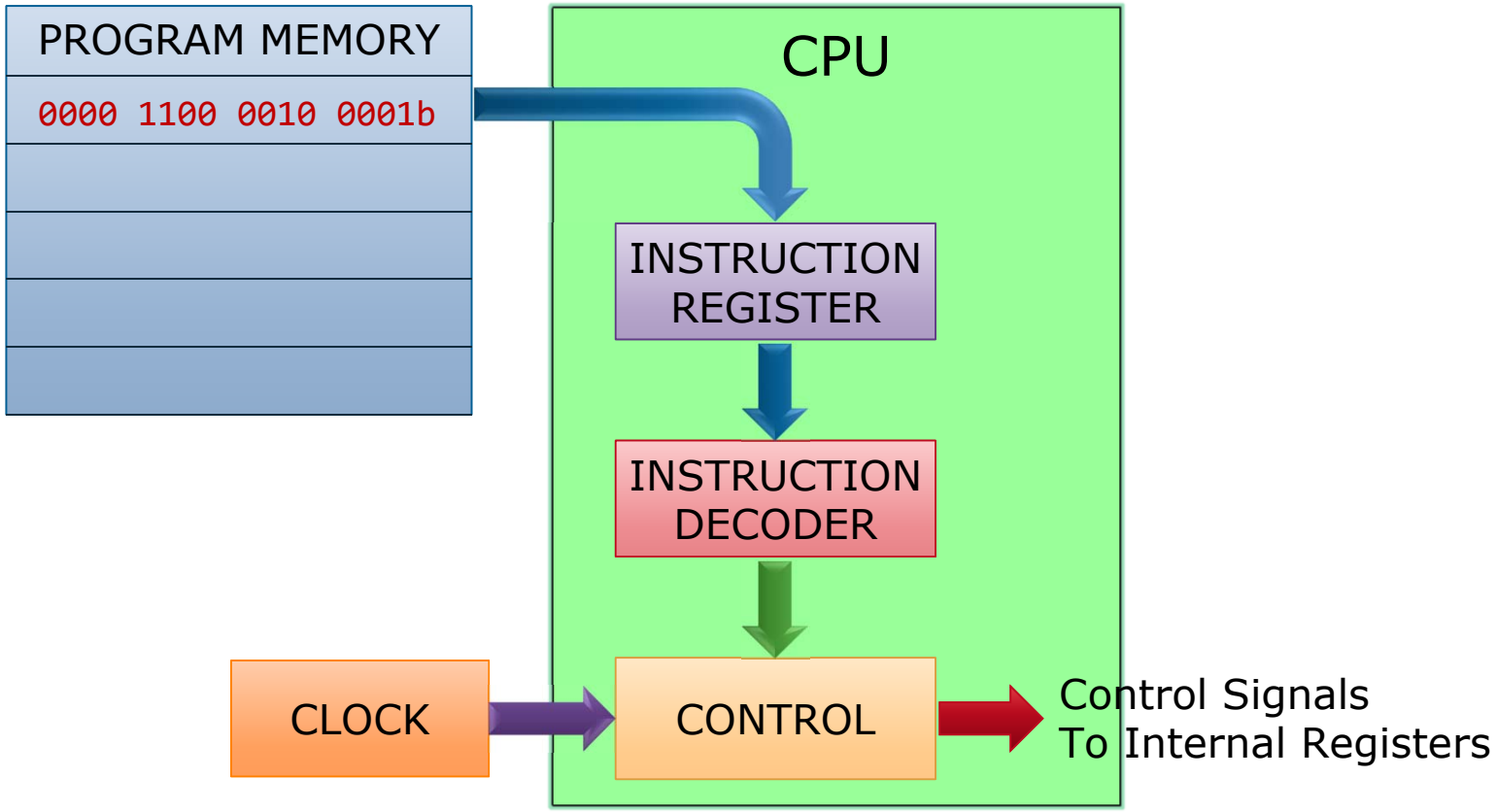
Central Processing Unit (CPU) (1)

- Arithmetic Logic Unit (ALU)
  - Execute numerical and logical operation
- Register file
  - Storage location inside the CPU
  - Used to hold data / memory address
  - Access to data in register is much faster than memory
  - Number of registers varies depending on CPU
- Control unit
  - Hardware instruction logic
  - Decodes and monitors the execution of instructions
  - System clock synchronizes the activities of CPU

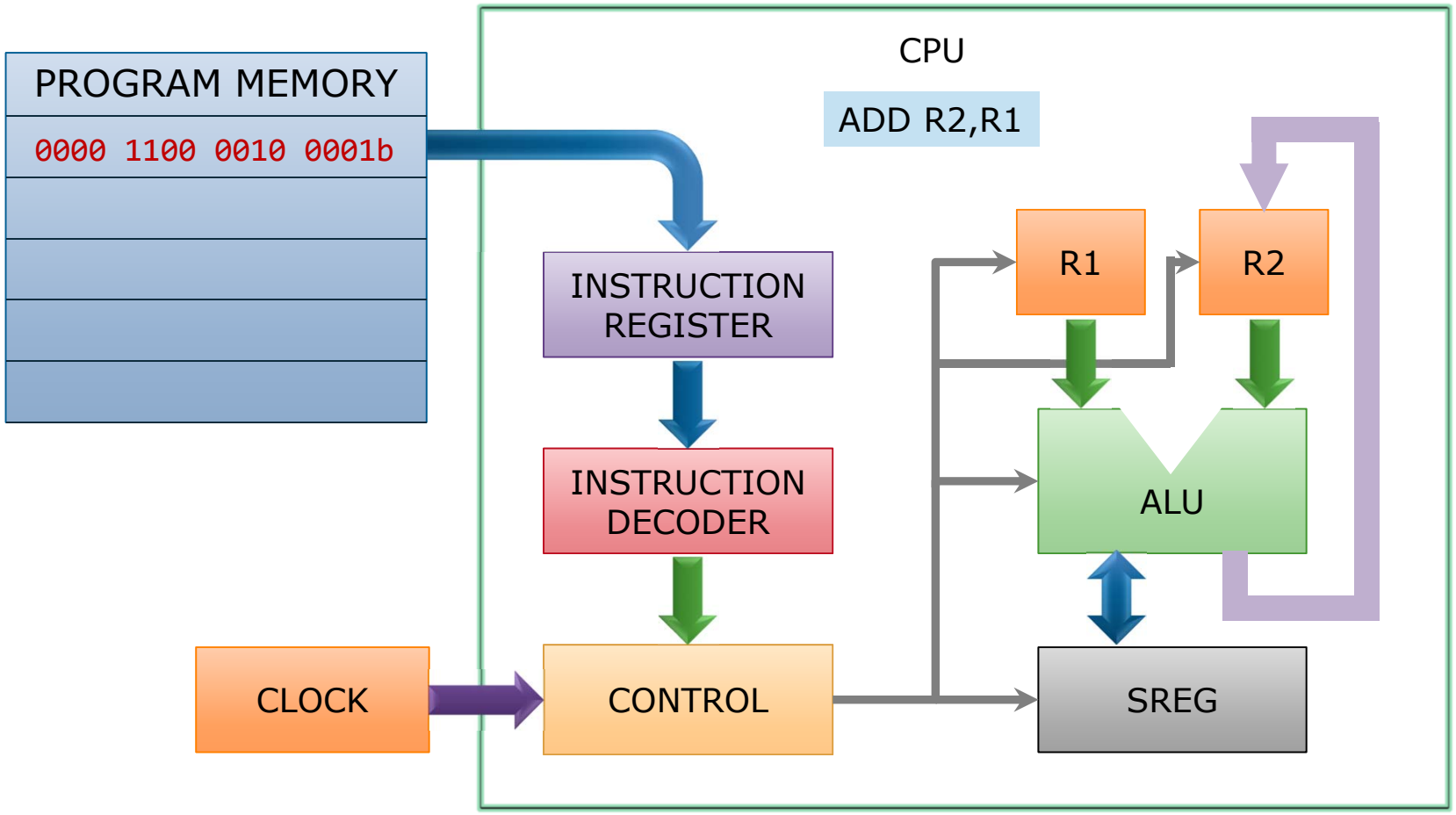
Central Processing Unit (CPU) (2)



Control Unit



ALU, Register File, and Control Unit



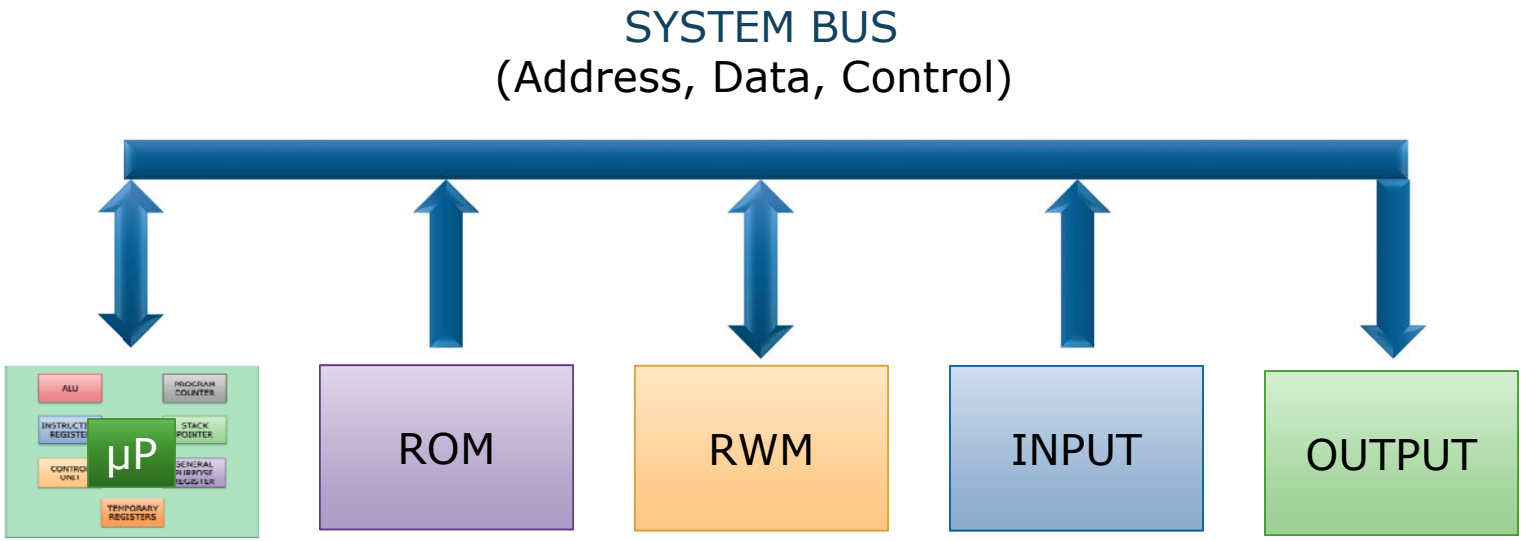
Microprocessor (μP)

- A processor fabricated in a single IC
- The number of bits of μP refers to

  the number of bits that μP can manipulate in one operation
- Limitation of μP
  - Requires external memory to execute programs.
  - Cannot directly interface with I/O devices.

    Peripheral chips are needed.
  - Address decoders and buffers are needed.
  - Bigger system size

Microprocessor-Based System



Memory Devices (1)

- Used for information storage
- Semiconductor memory devices

ROM	Mask ROM		Non-volatile	Read only
	PROM			
	EPROM	UV EPROM		Read/Write
		EEPROM		
		Flash Memory		
RAM (RWM)	SRAM		Volatile	
	DRAM			

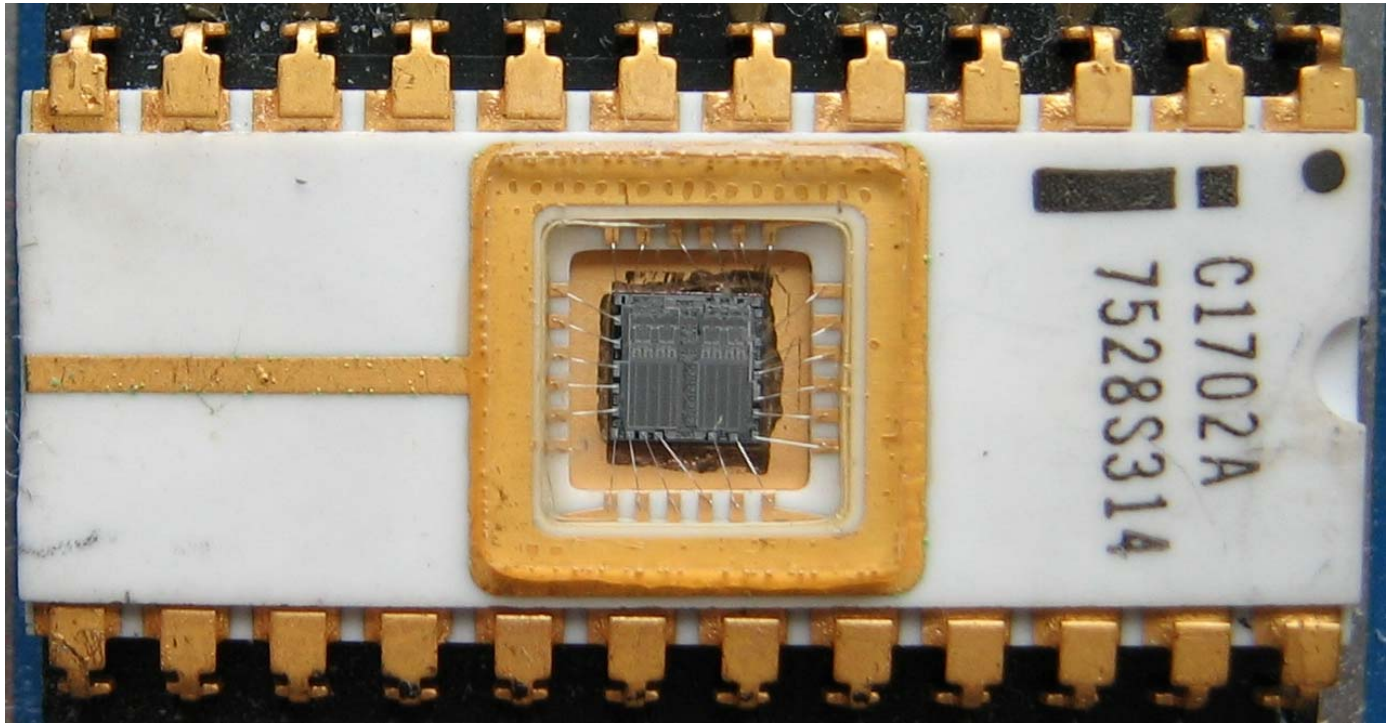
ROM: Read Only Memory
PROM: Programmable Read Only Memory
EPROM: Erasable Programmable Read Only Memory
EEPROM: Electrically Erasable Programmable Read Only Memory
UV EPROM: Ultra-Violet Erasable Programmable Read Only Memory
SRAM: Static Random Access Memory
DRAM: Dynamic Random Access Memory

Memory Devices (2)

PROM (Programmable Read Only Memory)

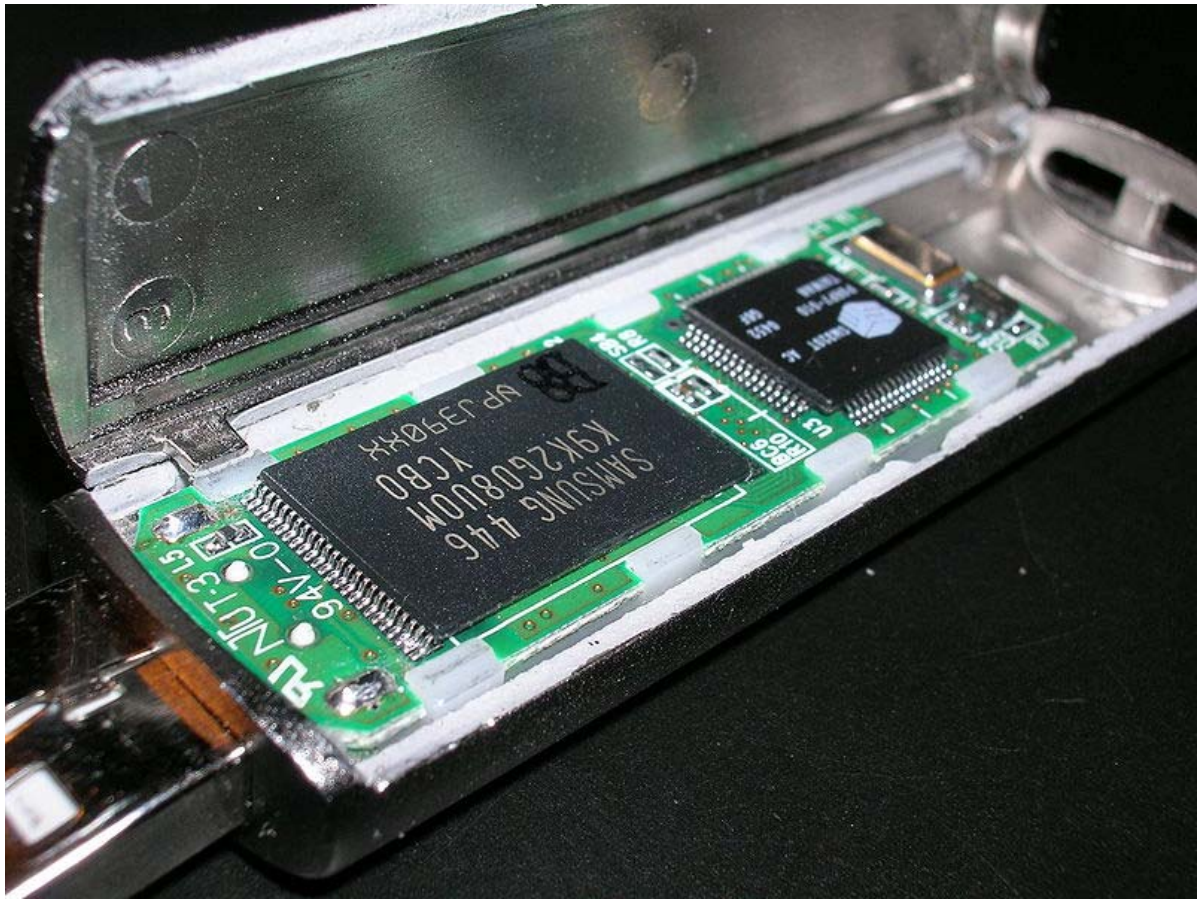
Memory Devices (3)

UV EPROM (Ultra-Violet Erasable Programmable Read Only Memory)



Memory Devices (4)

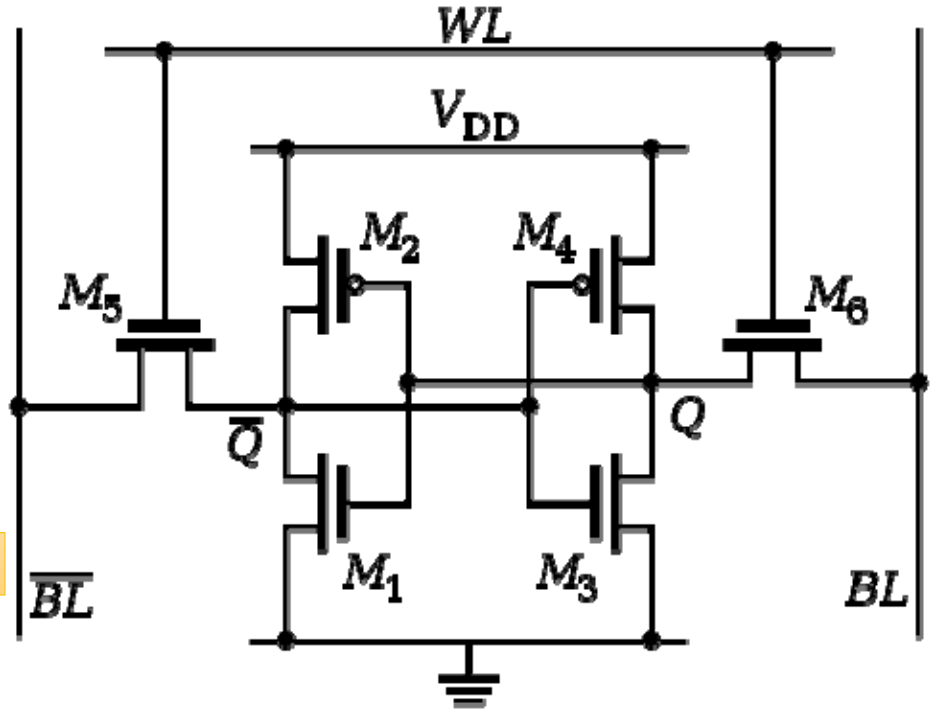
Flash Memory



Memory Devices (5)

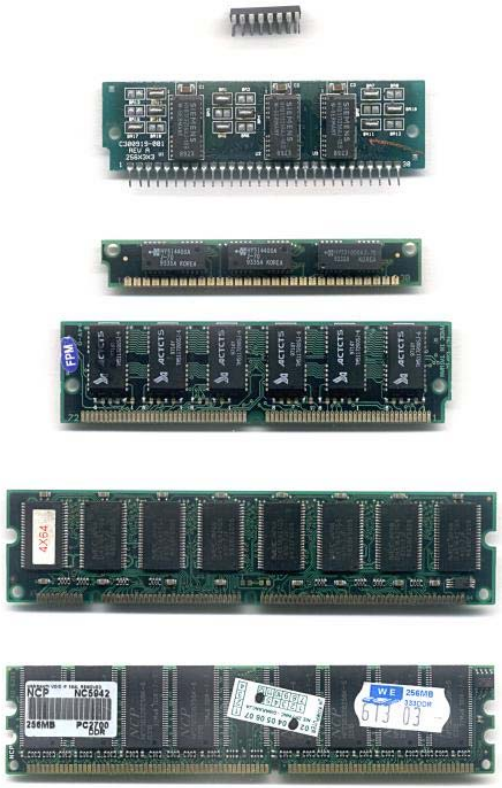
SRAM (Static Random Access Memory)

SRAM Cell (6 Transistors)



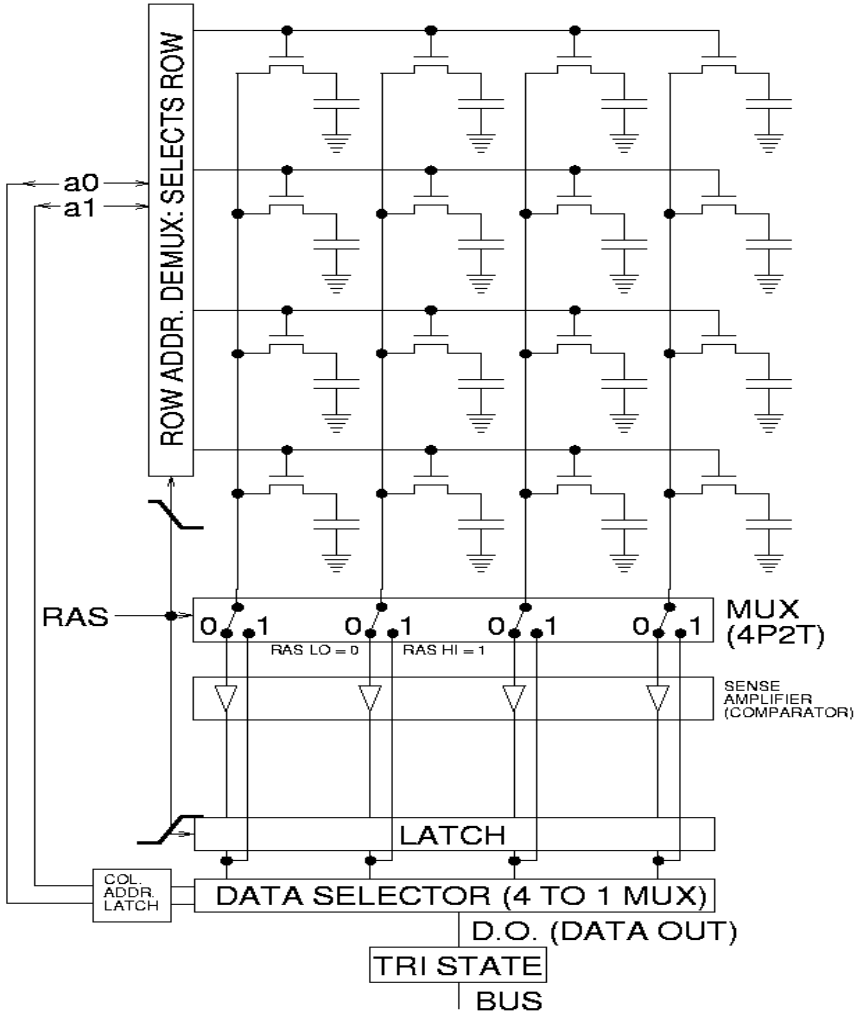
Memory Devices (6)

DRAM (Dynamic Random Access Memory)



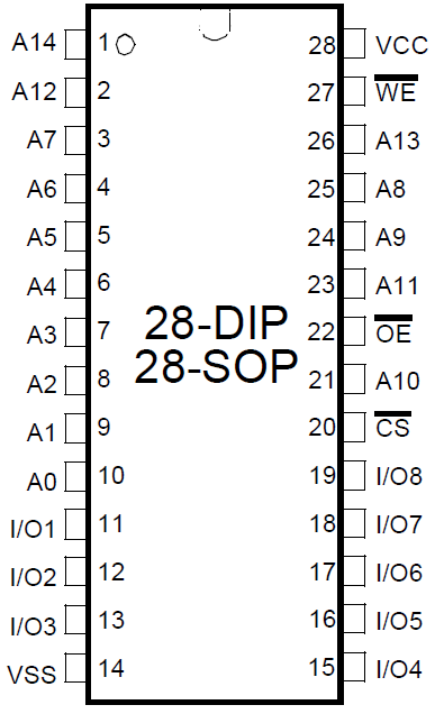
Refresh 필요

4 by 4 DRAM Cells Array



Memory Interface (1)

Memory Interface Example (62256 – 32k x 8 bit Low Power CMOS Static RAM)



Signal Name	Function
A0~A14	Address Inputs
$\overline{WE}$	Write Enable Input
$\overline{CS}$	Chip Select Input
$\overline{OE}$	Output Enable Input
I/O1~I/O8	Data Inputs/Outputs
VCC	Power(5V)
VSS	Ground

Memory Interface (2)

Memory Interface Example (62256 - 32k x 8 bit Low Power CMOS Static RAM)

- Each 8-bit memory cell group has its own address

B7	B6	B5	B4	B3	B2	B1	B0	Address
								0
								1
								2
								3
								4
								5
								6
								…
								32761
								32762
								32763
								32764
								32765
								32766
								32767

4 bits: Nibble

8 bits: Byte

16 bits: Word

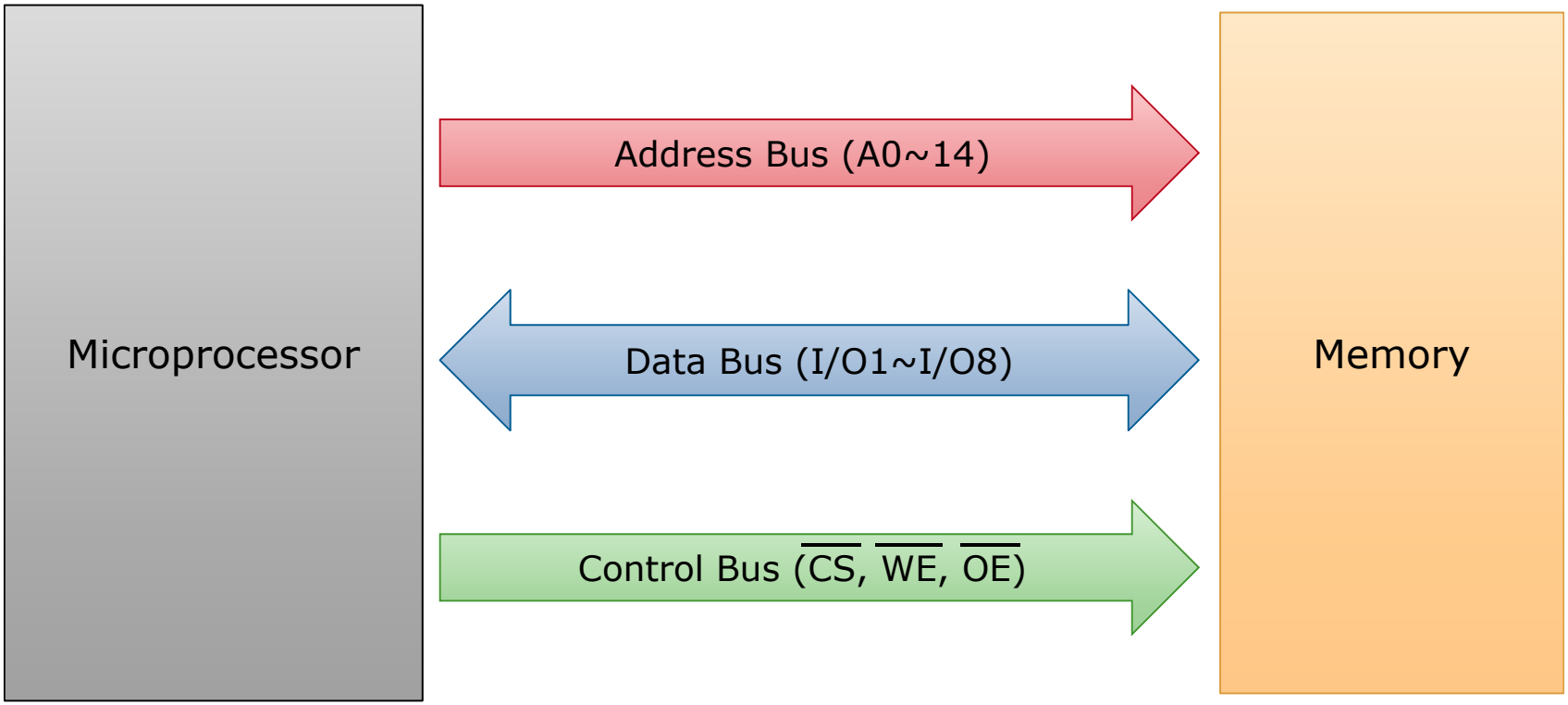
32 bits: Double Word

64 bits: Quadruple Word

1 kB = 2^{10} Bytes = 1,024 Bytes
1 MB = 2^{10} kBytes = 1,048,576 Bytes

Memory Interface (3)

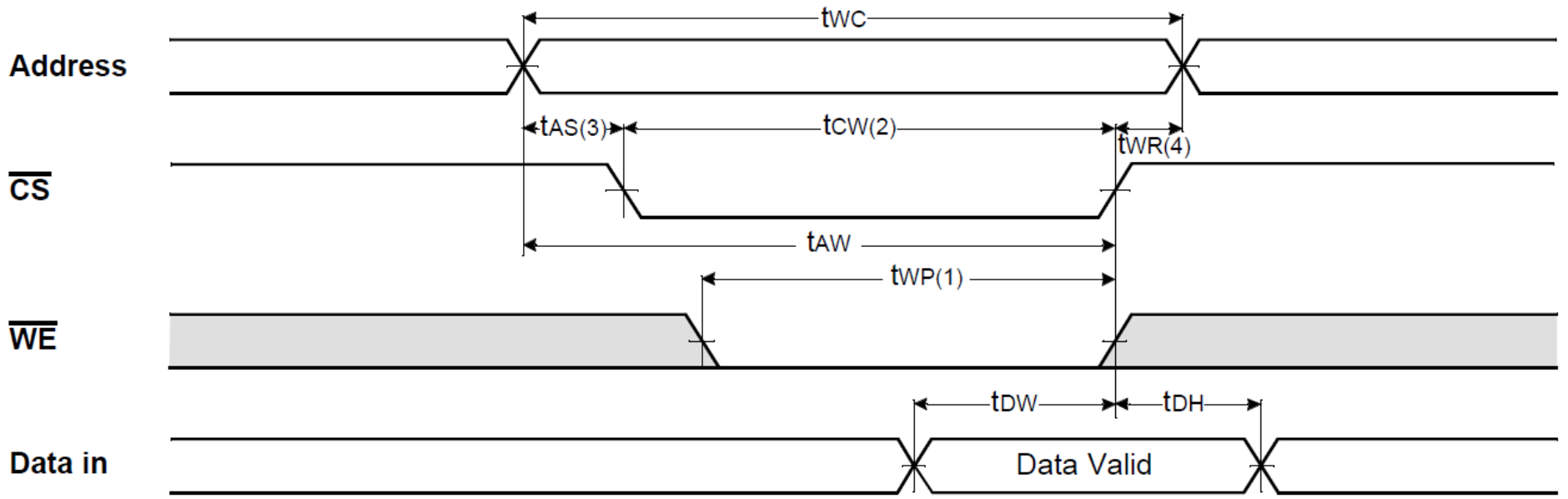
Memory Interface Example (62256 - 32k x 8 bit Low Power CMOS Static RAM)



Memory Interface (4)

Memory Interface Example (62256 - 32k x 8 bit Low Power CMOS Static RAM) (Write Cycle)

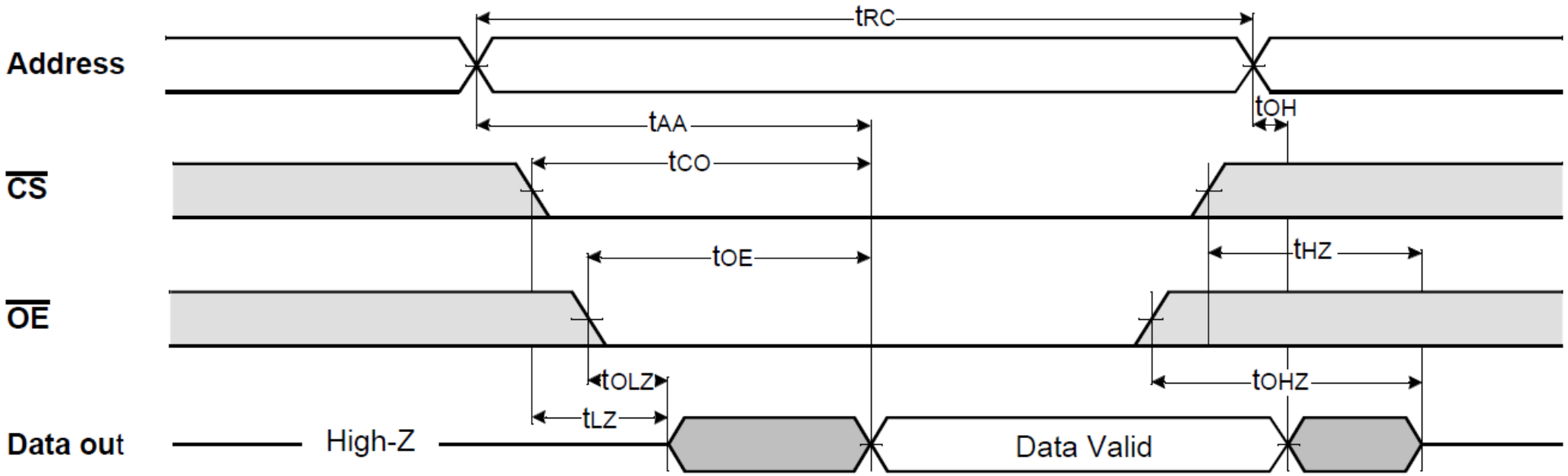
TIMING WAVEFORM OF WRITE CYCLE(2)



Memory Interface (5)

Memory Interface Example (62256 - 32k x 8 bit Low Power CMOS Static RAM) (Read Cycle)

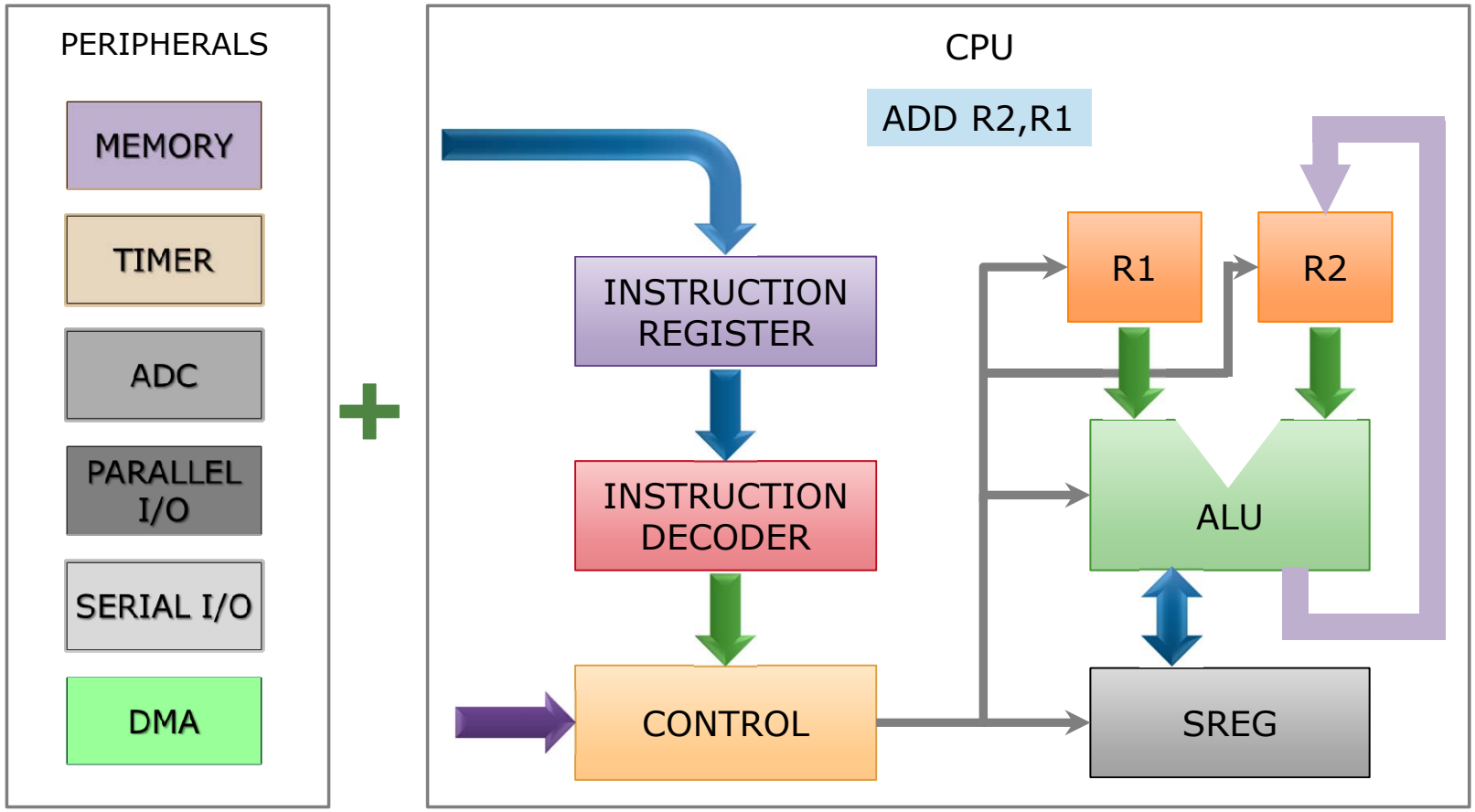
TIMING WAVEFORM OF READ CYCLE(2)



Microcontroller

- A computer implemented on a single VLSI chip
- Contains everything a µP contains plus
  - Memory
  - Timer
  - Analog-to-digital converter (ADC)
  - Digital-to-analog converter (DAC)
  - Parallel I/O interface
  - Serial I/O interface
  - Memory component interface circuitry
  - Direct memory access (DMA)

General Microcontroller

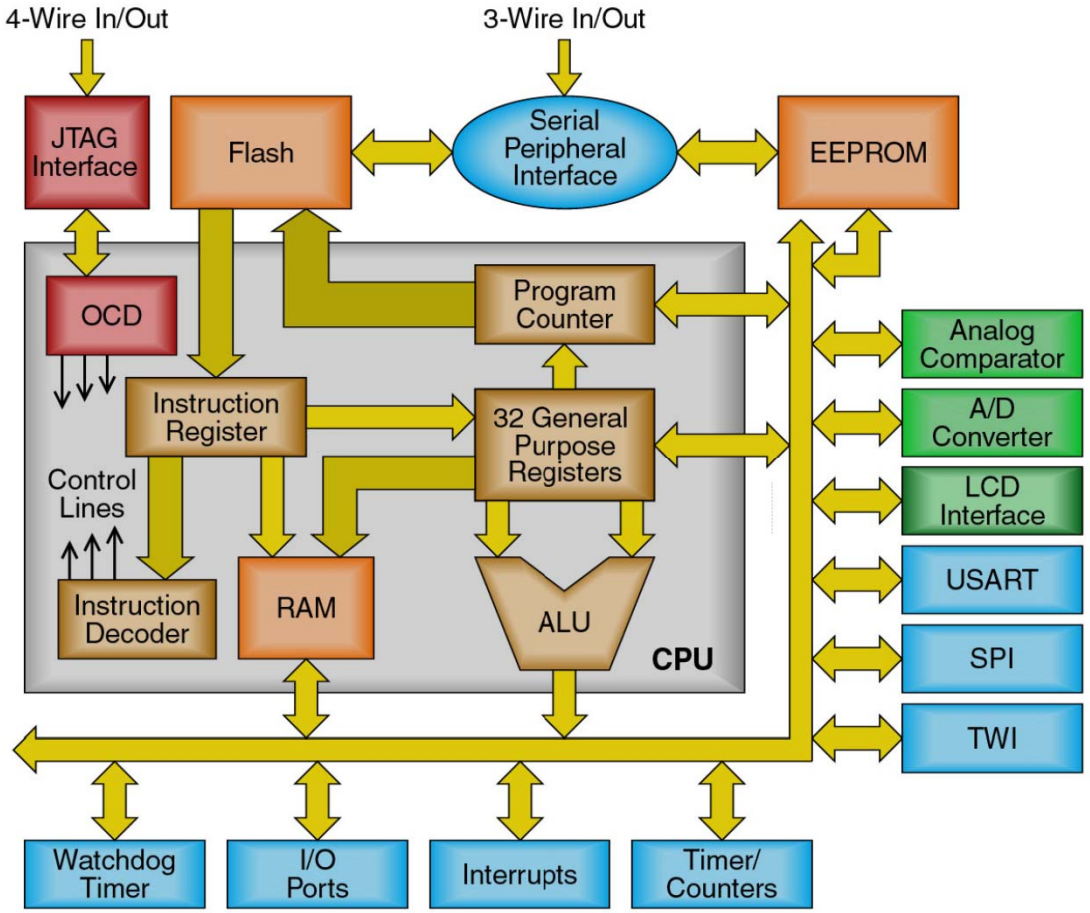


Overview of 8-bit AVR Microcontroller (1)

It is commonly accepted that **AVR** stands for **A**lf-Egil Bogen and **V**egard Wollan's **R**ISC processor.

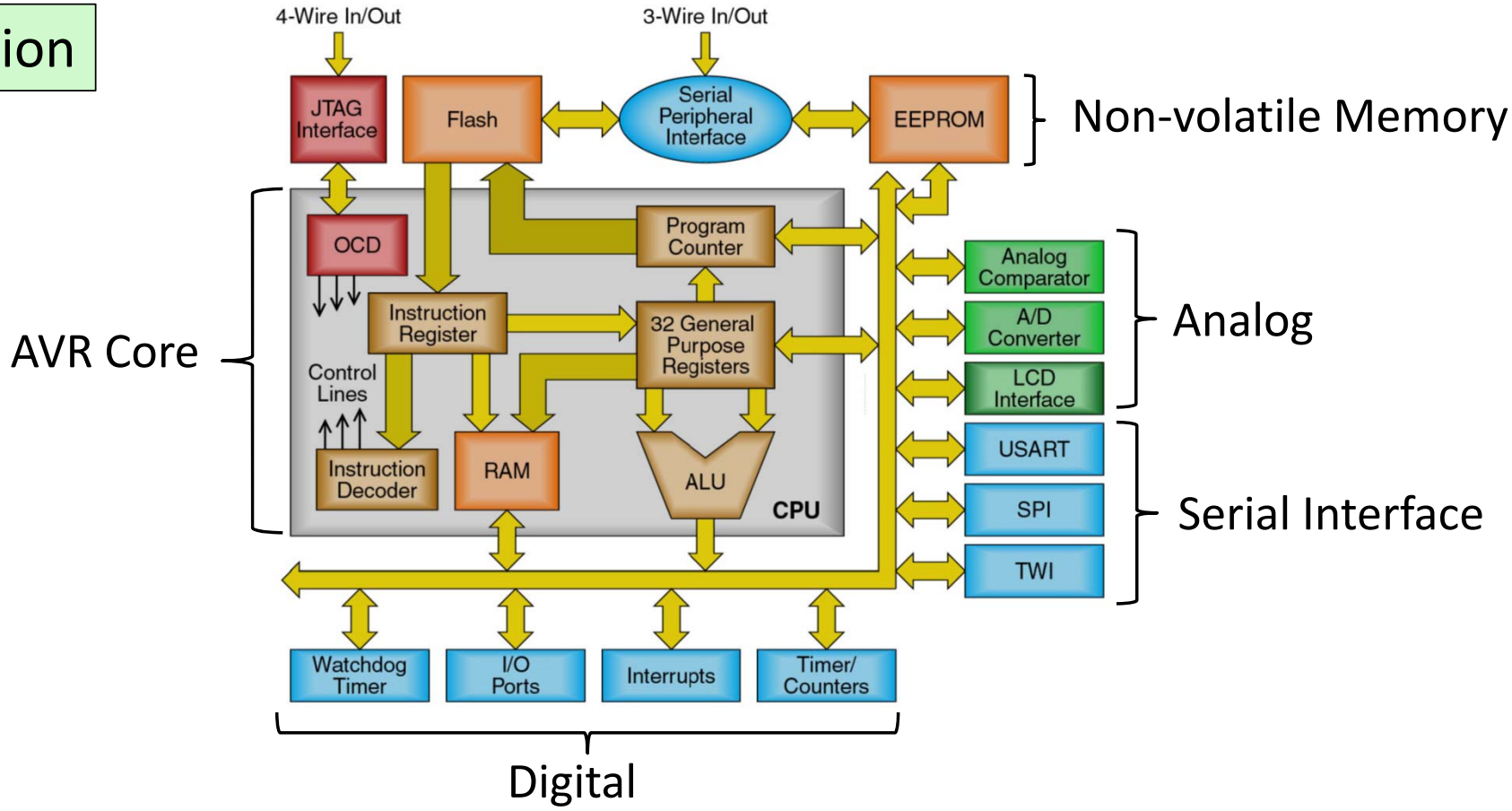
Overview of 8-bit AVR Microcontroller (2)

- RISC architecture with CISC instruction set
  - ✓ Powerful instruction set for C and Assembly
- Scalable
  - ✓ Same powerful AVR core in all devices
- Single cycle execution
  - ✓ One instruction per external clock
  - ✓ Low power consumption
- 32 working Registers
  - ✓ All directly connected to ALU!
- Very efficient core
  - ✓ 20 MIPS @ 20MHz
- High System Level Integration
  - ✓ Lowest total system cost



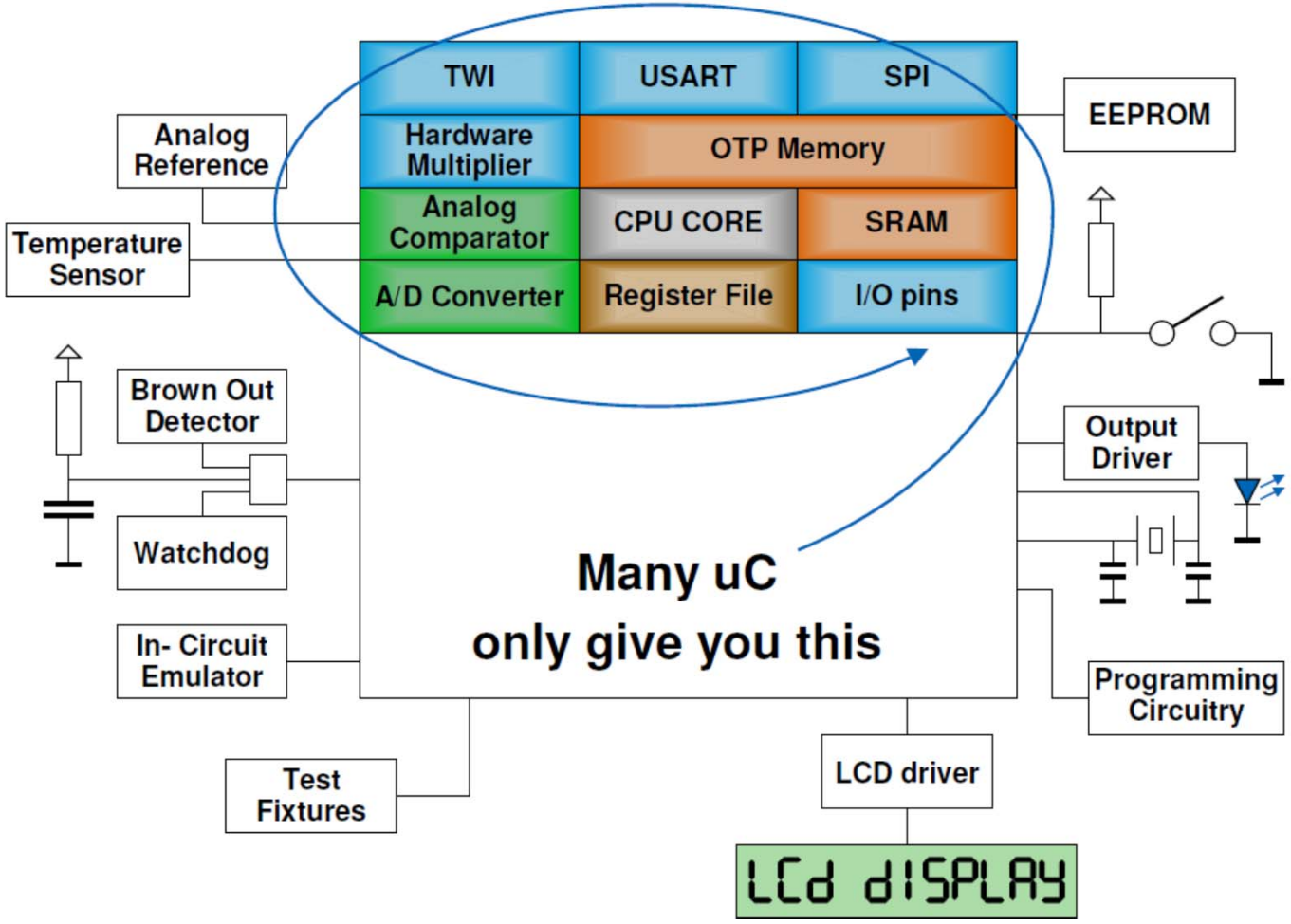
Overview of 8-bit AVR Microcontroller (3)

High Integration



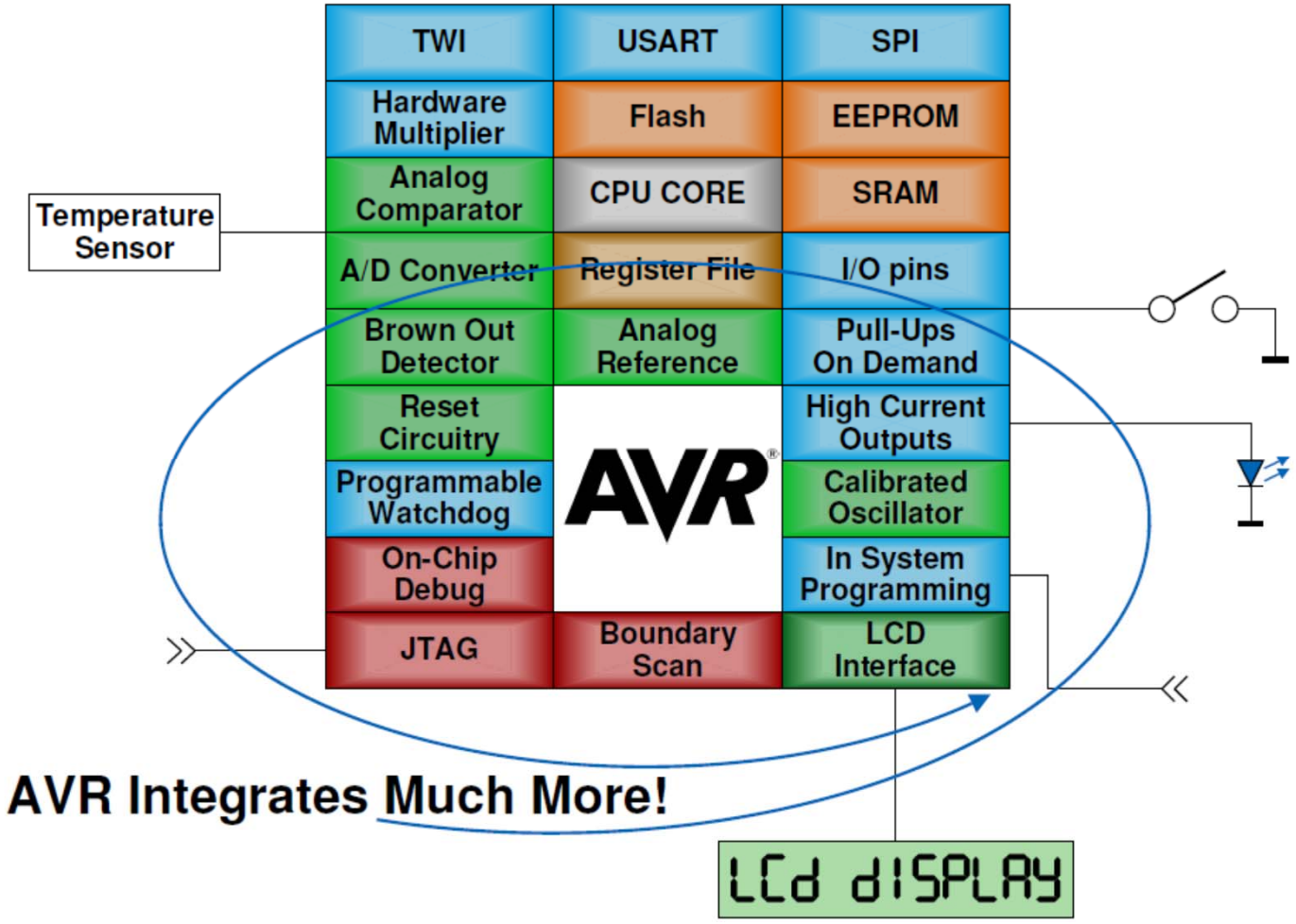
Overview of 8-bit AVR Microcontroller (4)

Single Chip Solution



Overview of 8-bit AVR Microcontroller (5)

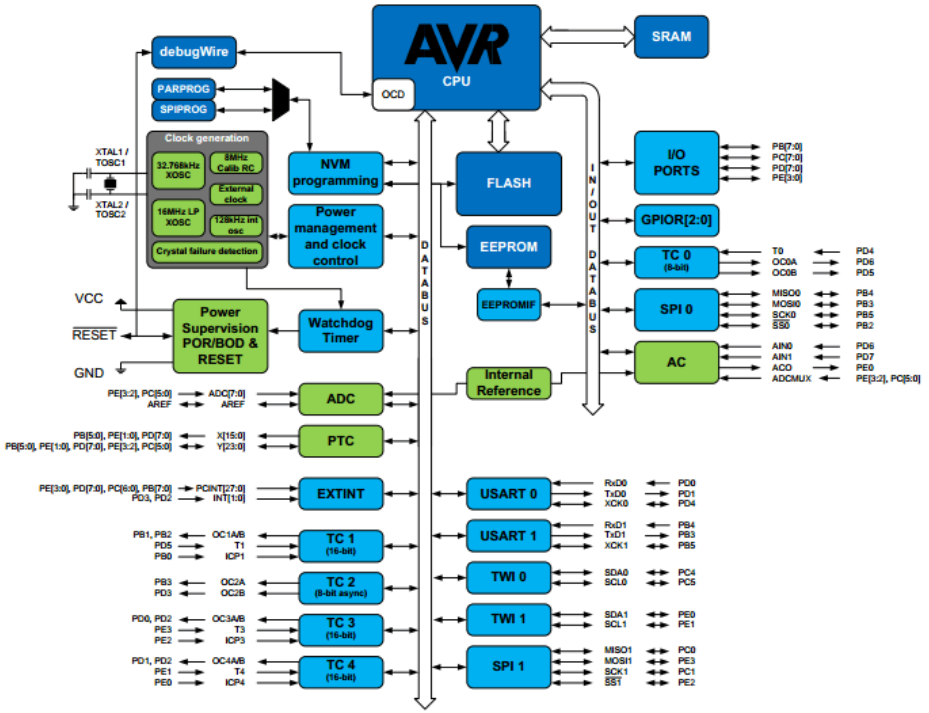
Single Chip Solution



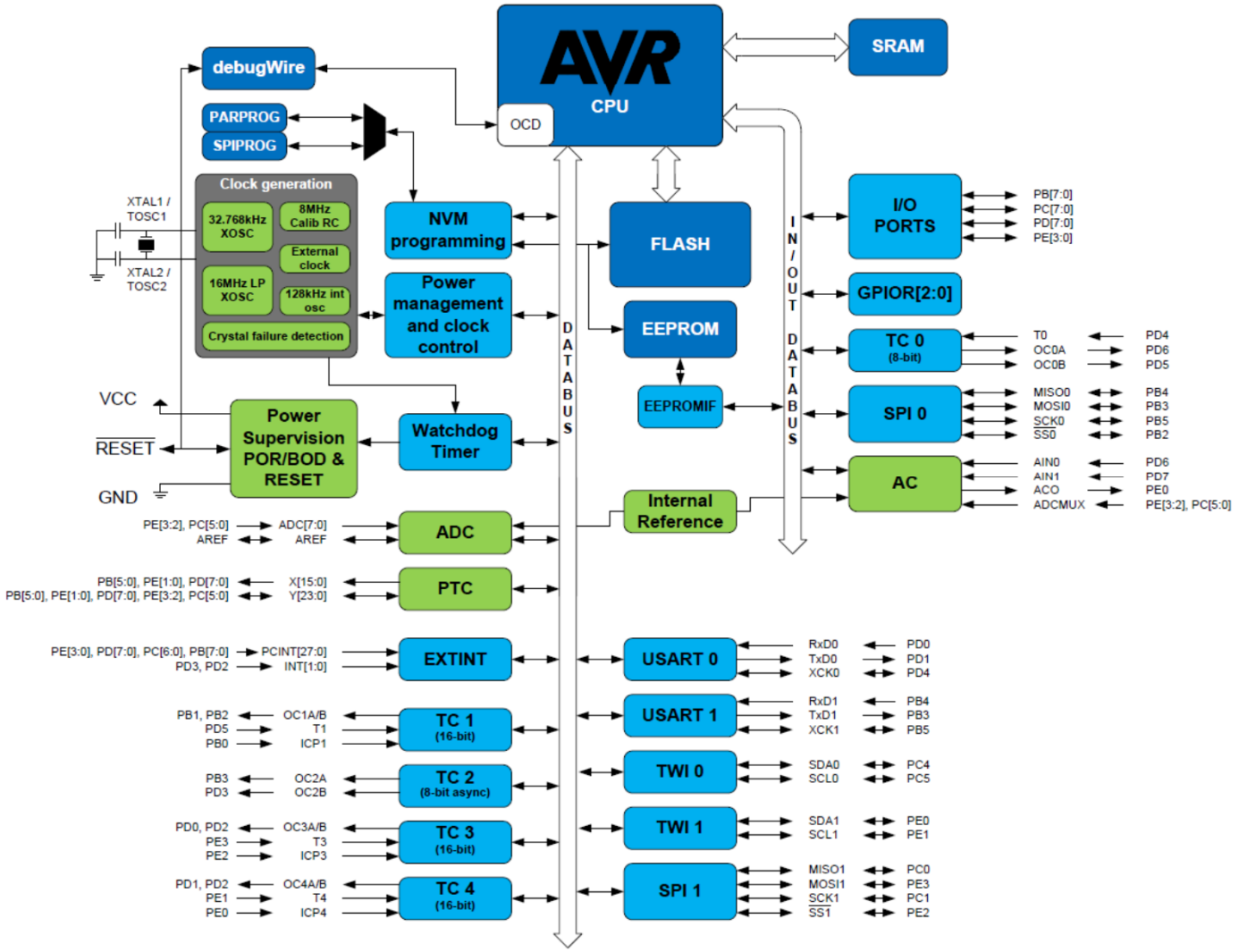
Introduction to ATmega328PB

ATmega328PB Features

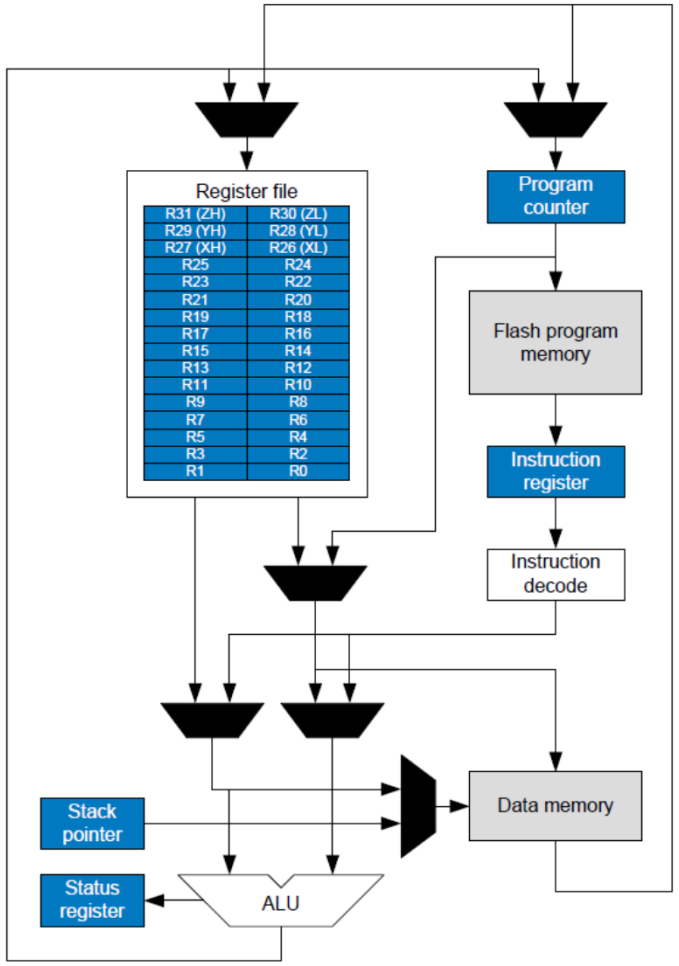
- Memories
  - 32 kBytes of In-System Self-Programmable Flash program memory
  - 1 kBytes EEPROM
  - 2 kBytes Internal SRAM
- Peripherals
  - Timer/Counters
  - 10-bit ADC
  - USART
  - SPI
  - TWI
- Operating Voltage: 1.8 - 5.5V
- Speed Grade:
  - 0 - 4MHz @ 1.8 - 5.5V
  - 0 - 10MHz @ 2.7 - 5.5.V
  - 0 - 20MHz @ 4.5 - 5.5V



ATmega328PB Block Diagram



AVR CPU Core



- AVR uses Harvard architecture
  - Separate memories and buses for program and data
- Single level pipelining for instruction
- Register File
  - 32 x 8-bit general purpose working registers with a single clock cycle access time
- ALU (Arithmetic Logic Unit)
- Status Register
  - Contains information about the result of the most recently executed arithmetic instruction.
  - This information can be used for altering program flow in order to perform conditional operations.
  - The Status Register is updated after all ALU operations.

AVR Status Register

Bit No.	7	6	5	4	3	2	1	0
Name	I	T	H	S	V	N	Z	C
Reset	0	0	0	0	0	0	0	0

- Bit 7 – I: Global Interrupt Enable
- Bit 6 – T: Copy Storage
- Bit 5 – H: Half Carry Flag
- Bit 4 – S: Sign Flag. $S = N \oplus V$
- Bit 3 – V: Two's Complement Overflow Flag
- Bit 2 – N: Negative Flag
- Bit 1 – Z: Zero Flag
- Bit 0 – C: Carry Flag

AVR General Purpose Register File

	Address	
R0	0x00	
R1	0x01	
R2	0x02	
…		
R13	0x0D	
R14	0x0E	
R15	0x0F	
R16	0x10	
R17	0x11	
…		
R26	0x1A	X-register Low Byte
R27	0x1B	X-register High Byte
R28	0x1C	Y-register Low Byte
R29	0x1D	Y-register High Byte
R30	0x1E	Z-register Low Byte
R31	0x1F	Z-register High Byte

Registers are special storages with 8 bits capacity.
They are connected directly to the CPU → fast access time.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Additional Function:
These registers are 16-bit address pointers for indirect addressing of the memory space.
X, Y, Z: for Data Memory
Z: Program Memory

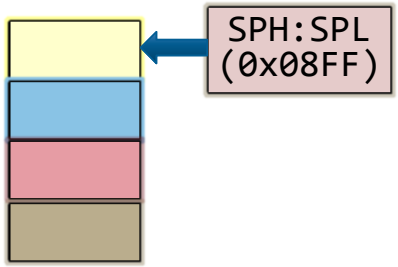
Stack Pointer for AVR

- Stack Pointer, SPL and SPH (0x3D and 0x3E)
  - A stack is a last-in first-out data structure
  - AVR stack is implemented as growing from higher to lower memory locations
  - 16-bit stack pointer points to the top of stack

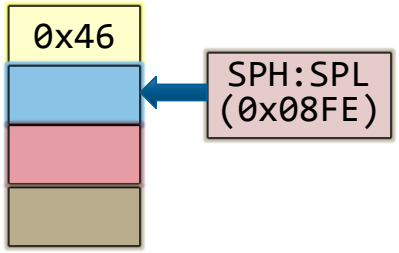
Stack and Stack Pointer for AVR

```
1.   LDI   R16,0x08
2.   OUT   SPH,R16
3.   LDI   R16,0xFF
4.   OUT   SPL,R16

5.   LDI   R16,0x46
6.   LDI   R17,0x47

7.   PUSH  R16
8.   PUSH  R17
9.   POP   R17
10.  POP   R19
```

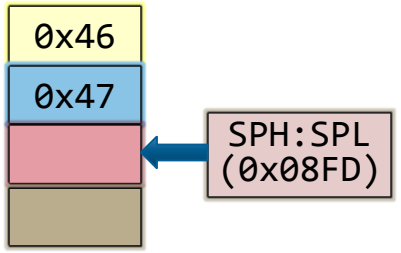
ATmega328PB RAM ADDRESS: 0x0100~0x08FF



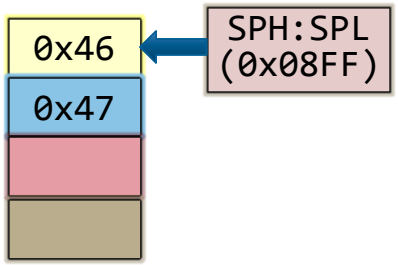
After Line #4



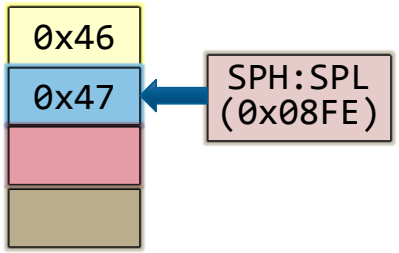
After Line #7



After Line #8

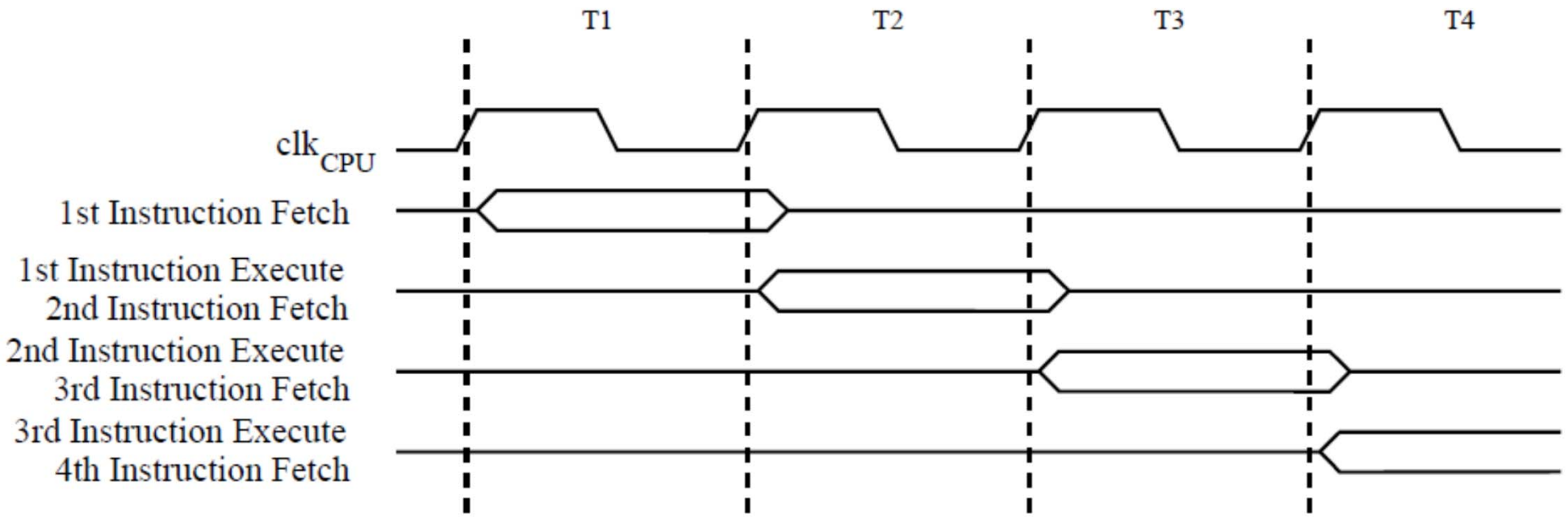


After Line #10



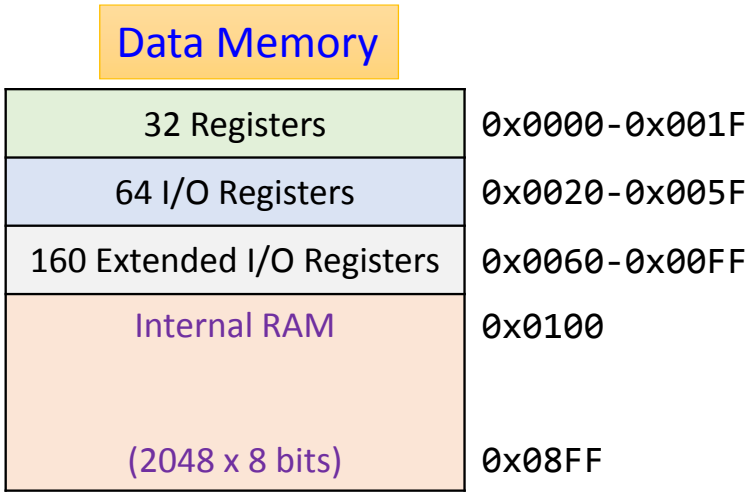
After Line #9

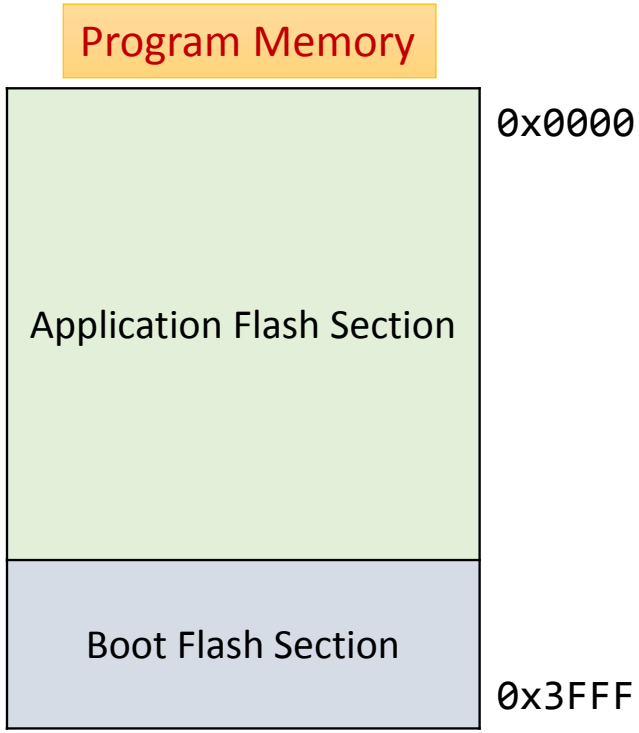
AVR Instruction Execution Timing (Pipeline)



ATmega328PB Memories (1)

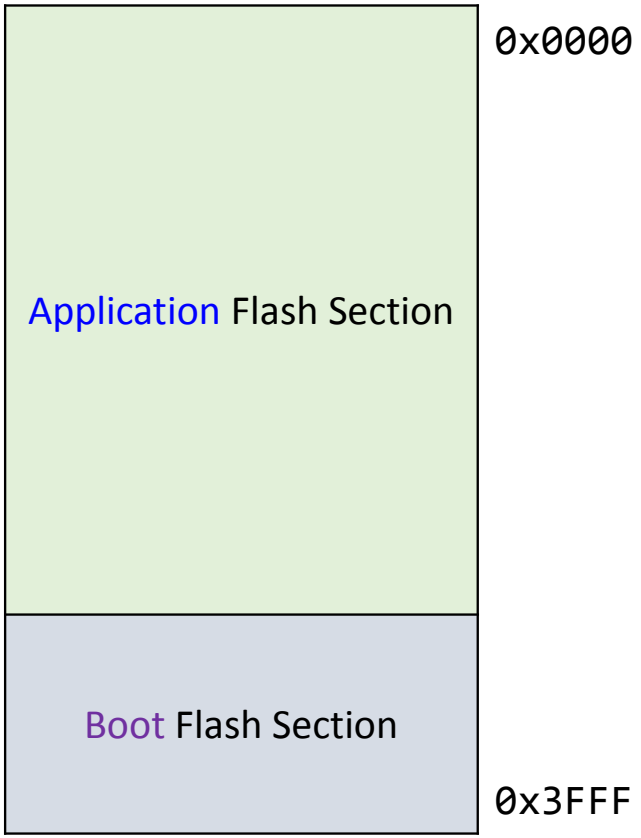
- Two memory spaces of ATmega328PB
  - Data memory
  - Program memory





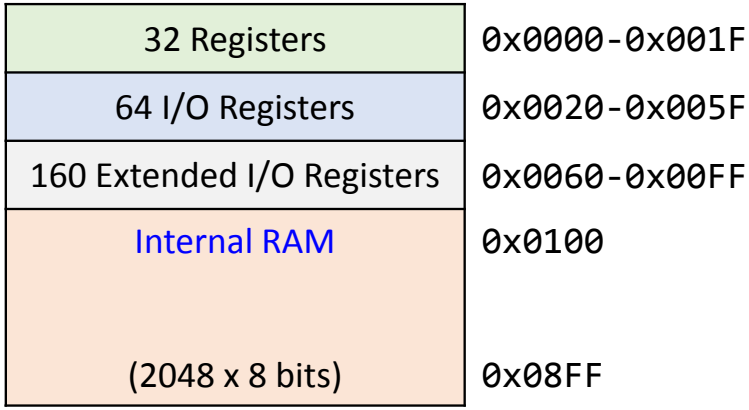
ATmega328PB Memories (2)

- In-System Reprogrammable Flash Program Memory
  - Boot Loader Section
  - Application Program Section



ATmega328PB Memories (3)

- SRAM Data Memory Space
  - Register File: 32
  - I/O Registers: 64
  - Extended I/O Registers: 160
  - Internal data SRAM: 2048



ATmega328PB Memories (4)

- EEPROM Data Memory
  - Electrically Erasable Programmable Read Only Memory
  - 1,024 Bytes of data EEPROM
  - Can be accessible by byte unit.
  - Endurance of at least 100,000 write/erase cycles.

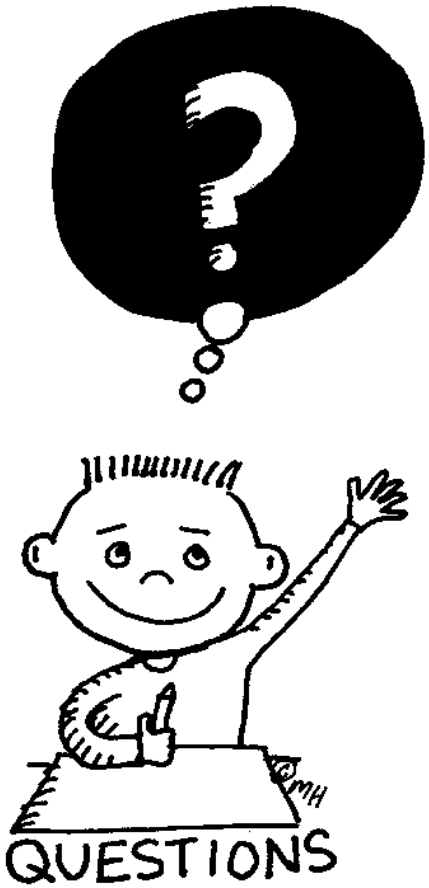
Units for Memory Size

Specific units of IEC 60027-2 A.2 and ISO/IEC 80000
(International Electrotechnical Commission)

IEC prefix		Representations				Customary prefix	
Name	**Symbol**	**Base 2**	**Base 1024**	**Value**	**Base 10**	**Name**	**Symbol**
kibi	Ki	2^{10}	1024^{1}	1024	≈1.02×10^3	kilo	k, K
mebi	Mi	2^{20}	1024^{2}	1,048,576	≈1.05×10^6	mega	M
gibi	Gi	2^{30}	1024^{3}	1,073,741,824	≈1.07×10^9	giga	G
tebi	Ti	2^{40}	1024^{4}	1,099,511,627,776	≈1.10×10^12	tera	T
pebi	Pi	2^{50}	1024^{5}	1,125,899,906,842,624	≈1.13×10^15	peta	P
exbi	Ei	2^{60}	1024^{6}	1,152,921,504,606,846,976	≈1.15×10^18	exa	E
zebi	Zi	2^{70}	1024^{7}	1,180,591,620,717,411,303,424	≈1.18×10^21	zetta	Z
yobi	Yi	2^{80}	1024^{8}	1,208,925,819,614,629,174,706,176	≈1.21×10^24	yotta	Y

Summary

- Number system
  - Decimal, binary, and hexadecimal numbers
- Computer organization
  - hardware and software
- Hardware
  - CPU, input, output, and memory
- Software
  - Machine language, Assembly language
- CPU
  - ALU, register file, and control unit
- Microprocessor
- Microcontroller
- AVR microcontroller


QUESTIONS


WHAT'S
NEXT?