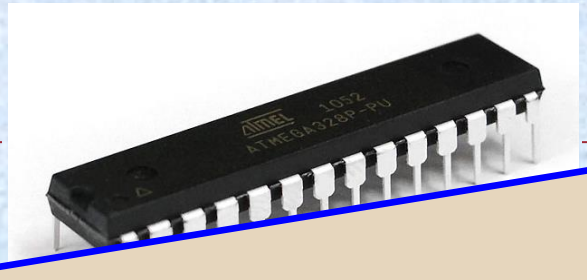
Vishwakarma Institute of Technology

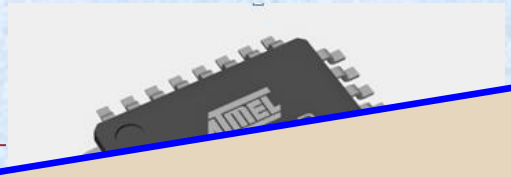
FY - Department of Engineering, Sciences and Humanities

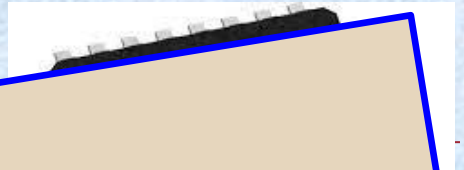
Micro Controller 3.2

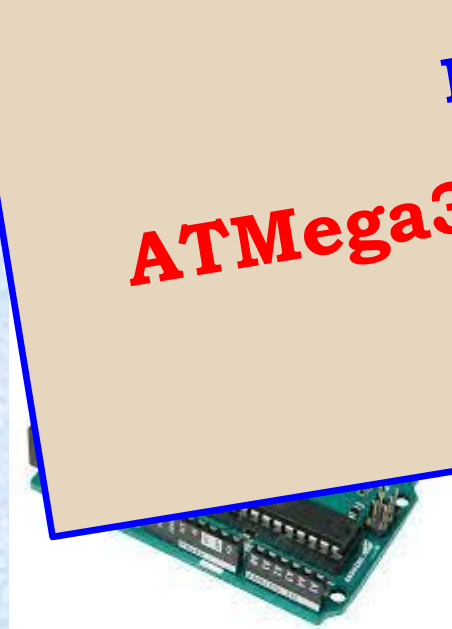
ATMega328P …. SFRs, Addr Modes

FY – DESH – VIT









Status Register of ATMega328P

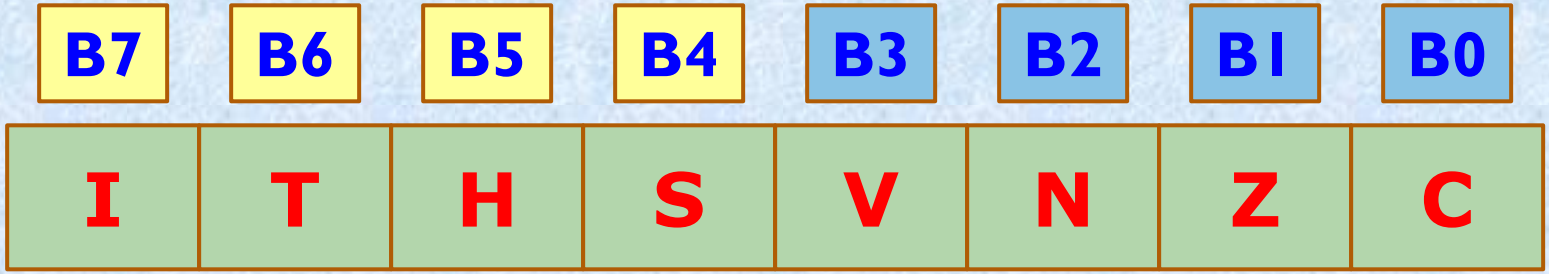
The Status Register contains information about the result of the recent arithmetic instruction executed. This information can be used for conditional decisions.

The Status Register is updated after any ALU operations is over.

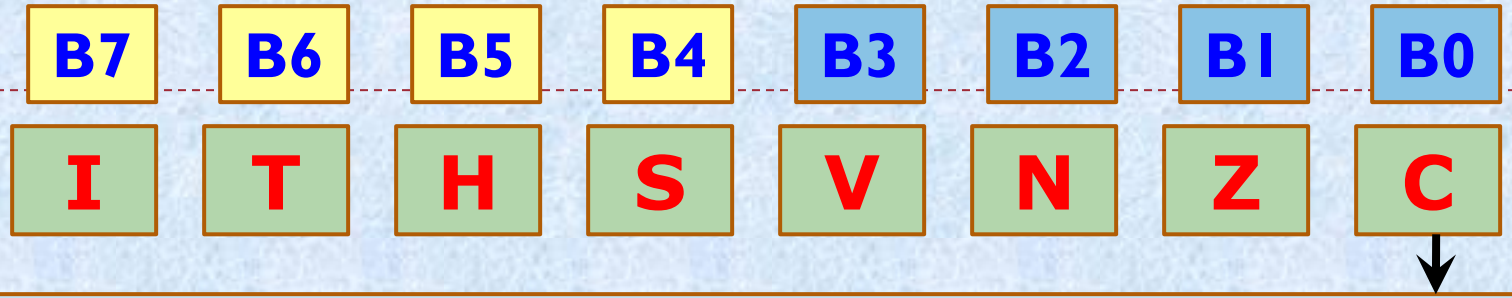
The Status Register is not automatically stored when entering an interrupt routine or restored when returning from an interrupt. This must be handled by software.

Status Register of ATMega328P

8 bit Register



Bits of the Status Register tell about the latest information of the arithmetic results in the ALU of the Accumulator.

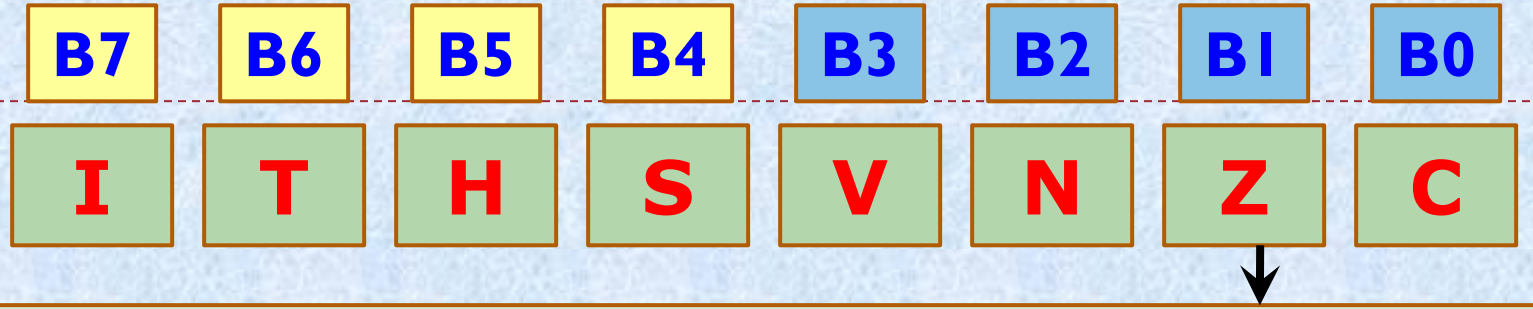


Bit 0 - C
Carry Flag

This bit is required in unsigned arithmetic or logic operations.

If C = 1 → a Carry is generated out of MSB (9^{th} place).

If C = 0 → a Carry is not generated out of MSB.

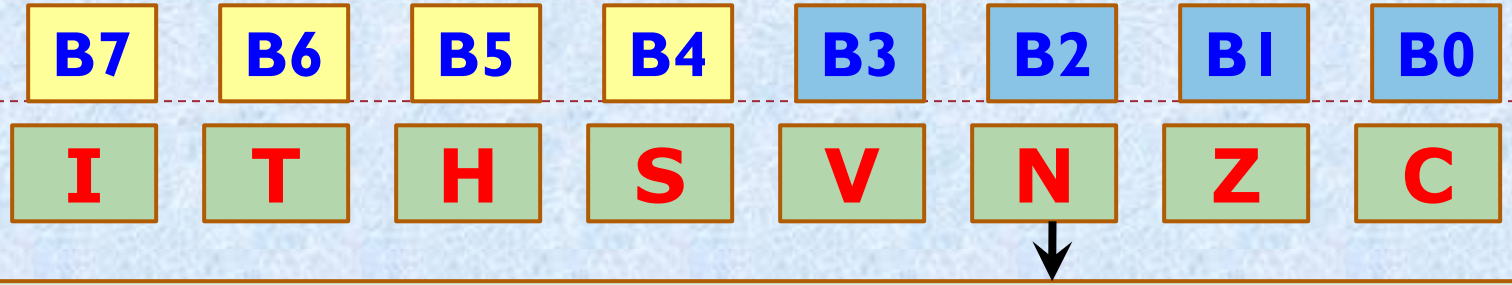


Bit 1 - Z
Zero Flag

The Zero Flag Z indicates a zero result in an arithmetic or logic operation.

Z bit= 1 means the operation has given answer as 0.

Z bit= 0 means the operation has given answer as non zero number.



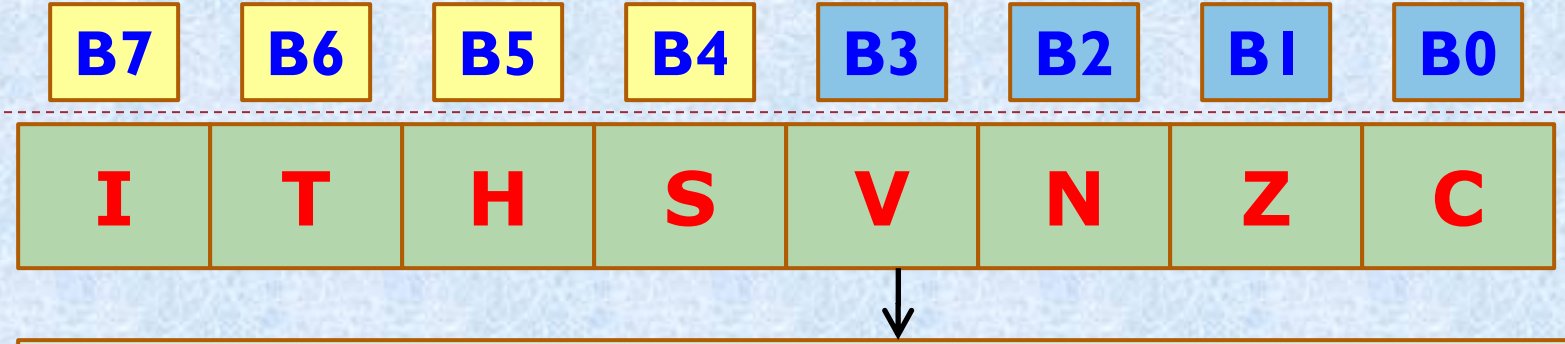
Bit 2 - N
Negative Flag

The Negative Flag N indicates a negative result in an arithmetic or logic operation in the Acc.

The MSB in Acc represents sign of the number in signed arithmetic operations.

MSB =1 → −ve number. N = 1 → result is −ve number

MSB =0 → +ve number. N = 0 → result is +ve number



Bit 3 – V

Overflow Flag

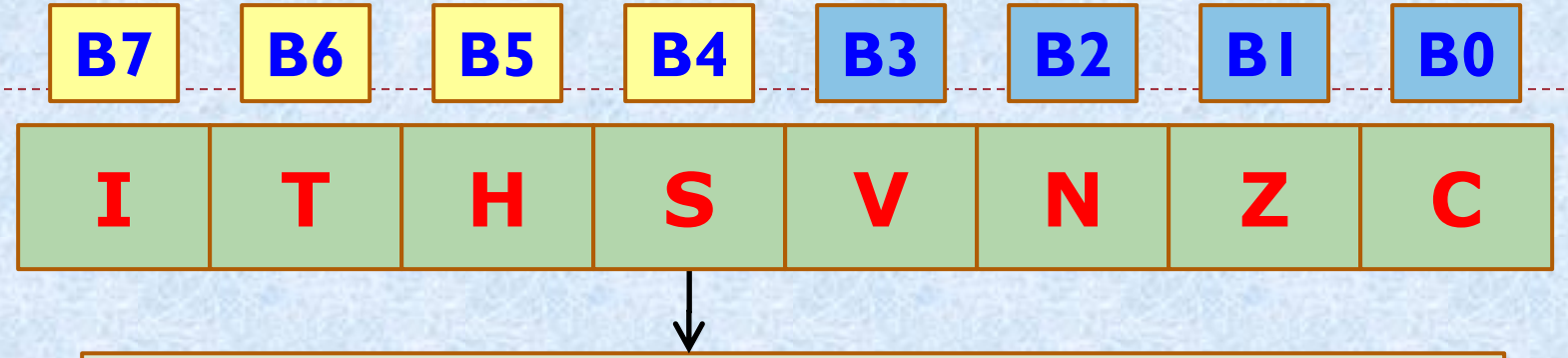
The MSB represents sign of the number in signed arithmetic operations.

MSB = 1 → a −ve number.

MSB = 0 → a +ve number.

B3 will be set to 1, if there occurs a Carry from Bit 6 to 7 in Acc.

B3 = 0 (cleared) if there is no Carry in Acc.



Bit 4 – S

Sign Flag,

$S = V \oplus N$

$\oplus$ means exclusive OR.

Thus, the bit 4 i.e. S is set to 1 if N and V differ from each other.

Example : 120 + 10 = 130 (decimal)

120=78h and 10=Ah ….. Add

```
  0111 1000
+ 0000 1010
  -----------
  1000 0010
```

= 130 but there is a carry from bit 6 to bit 7, thus **bit V will be set to 1.**

What about bit N = ??

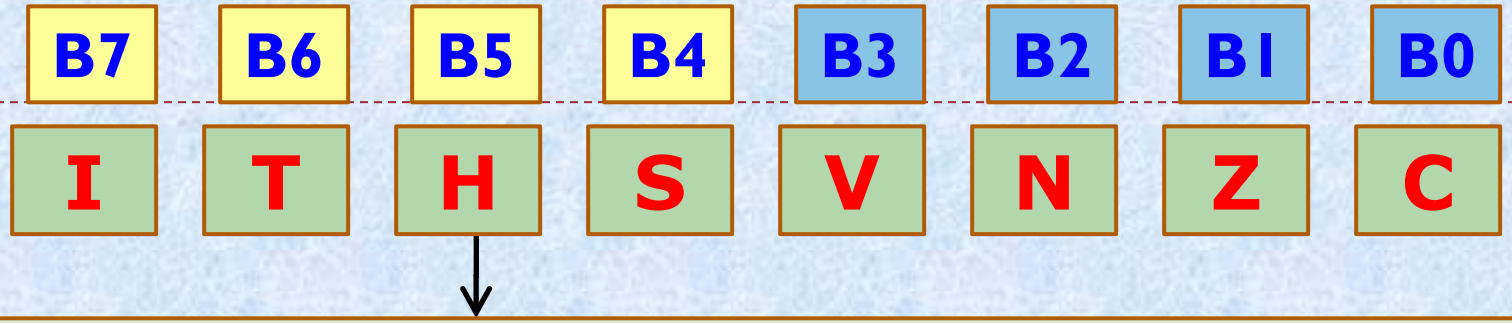
Bit N will also be set to 1 as MSB = 1 i.e. Negative number !

This is absolutely wrong !

120 + 10 = 1000 0010 = -2 ??

Status of bit S = V ⊕ N = ? = 0. This is the correct result.

Thus, Ex-OR will rectify the result.



Bit 5 - H
Half Carry Flag

The Half Carry Flag H indicates a Half Carry in some unsigned arithmetic operations. Half Carry Flag is useful in BCD arithmetic.

Bit = 1 if there is a Carry from **lower nibble** to **higher nibble** during arithmetic operations.

Bit = 0 if there is no such Carry from LN to HN.

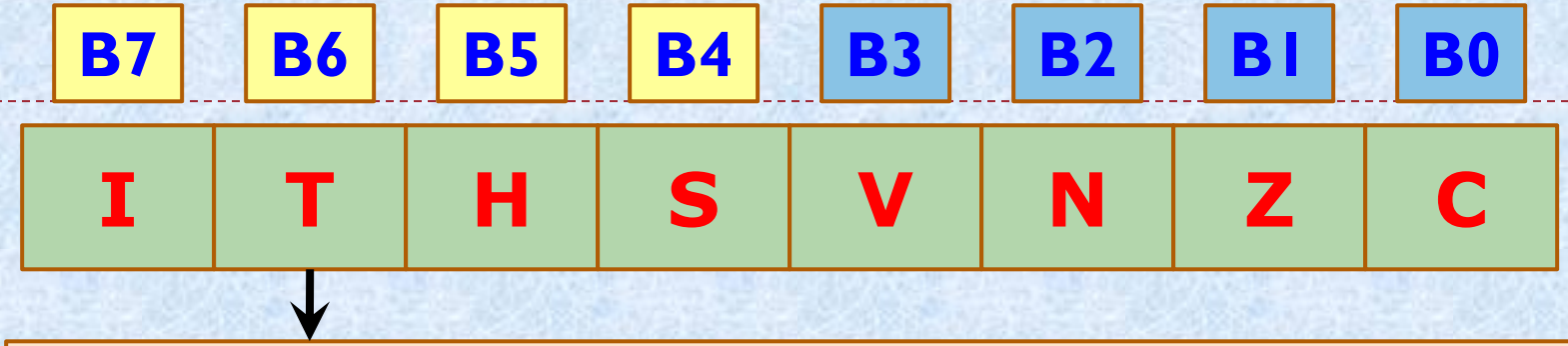
Ex. - Find status of H bit for .....
1) 0Dh + 15h and 2) 16h + 16h addition.

Ex. - Find status of H bit for ……
1) 0Dh + 15h and 2) 16h + 16h addition.

0D = 0000 1101	16 = 0001 0110
+ 15 = 0001 0101	+ 16 = 0001 0110
22 = 0010 0010	2C = 0010 1100

Higher Nibble is same in both cases. What about ……

H bit = ?? H bit = ??



Bit 6 - T

Bit Transfer - Copy and Storage bit

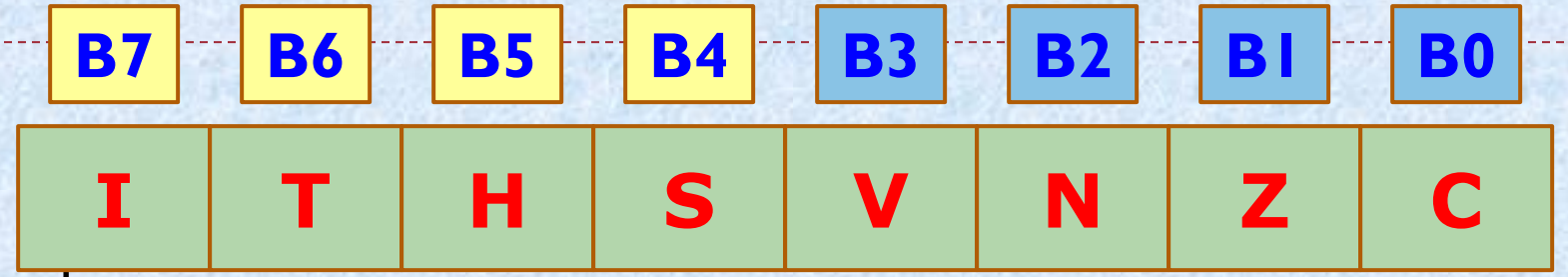
A bit from some Register from Register file can be copied into T by Bit Store instruction – BST.

OR

A bit in T is copied and loaded into some other Register from Register file. Bit Load instruction – BLD

Thus the T-bit becomes either a source or a destination.

Status Register of ATMega328P



Bit 7 - I

Global Interrupt Enable

If set to 1, all the Interrupts are enabled. The individual interrupt enable control is then performed in separate control registers.

If the Global Interrupt Enable bit B7 is cleared i.e. made 0, all the interrupts are disabled irrespective of the individual interrupt enable settings.

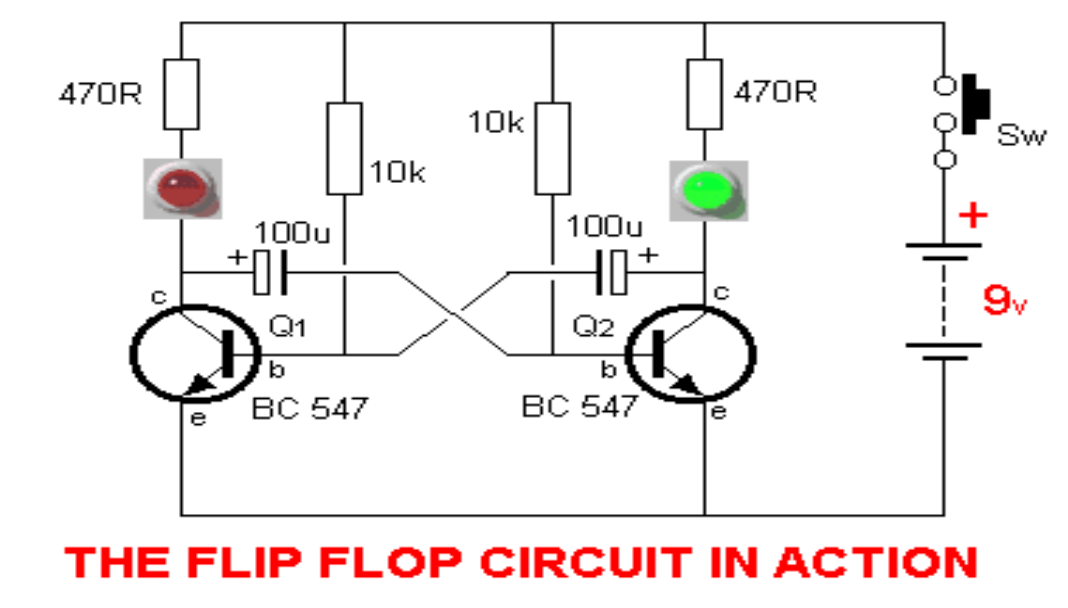
Memory Structure of ATMega328P

These are the two states for the transistors in the Flip Flop circuit. One transistor is CUT OFF while the other is SATURATED.

A transistor that is OFF is said to be in CUT-OFF condition.

and one that is fully turned ON is said to be SATURATED.

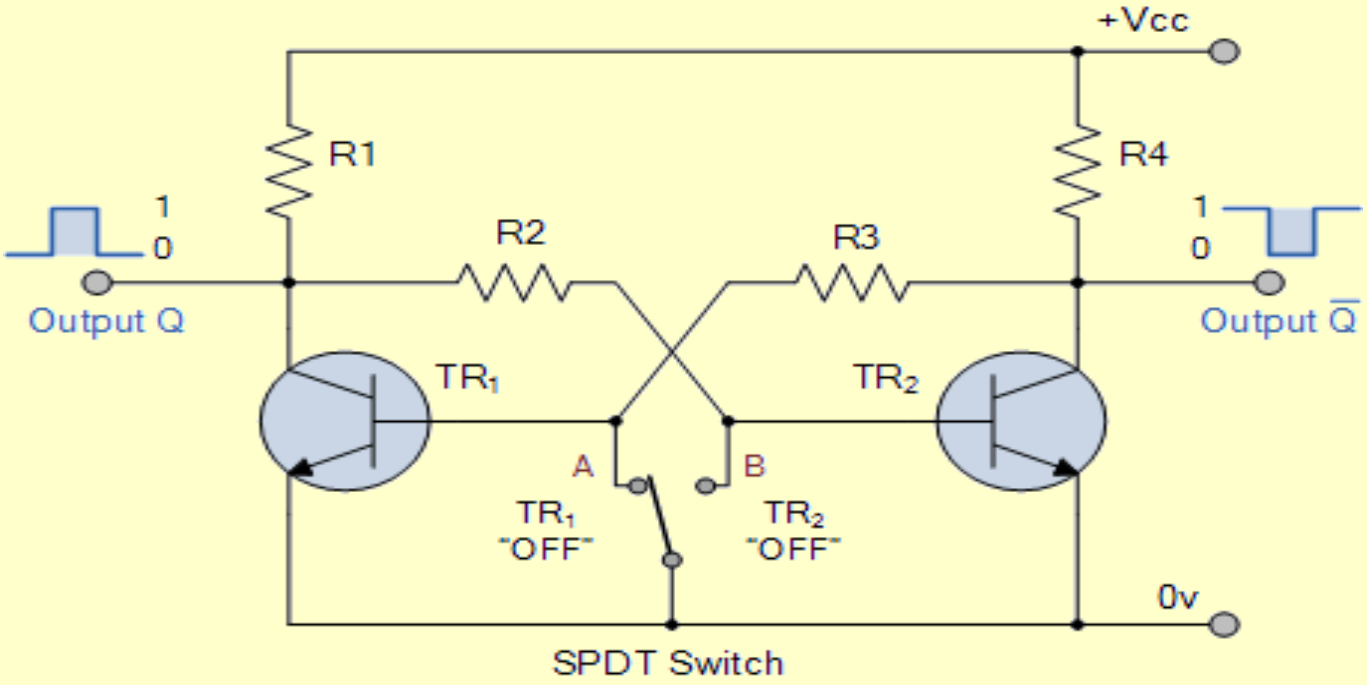
Memory Structure of ATMega328P



THE FLIP FLOP CIRCUIT IN ACTION

Basic memory building block

Memory Structure of ATMega328P



Memory Structure

1) Memory in a microcontroller is a space where a) Data, b) Program Instructions and c) Control instructions are stored.
2) Memory is always divided in multiple small parts called as memory locations.

3) Every memory location is of same size which will store some data in it.
4) To access the data in the memory, each memory location is allotted an identifying unique binary number called as **address**.

Memory Structure

5) Byte addressable memory – The length of data is 1 Byte only.

6) Word addressable memory – The length of data is a Word of 2 Bytes, 4 Bytes or 8 Bytes etc. (Architecture specific)

7) If more than 1 Byte of data is to be stored in Byte Addressable Memory, then consecutive locations are used to store the data.

Memory Structure

Types of memory used depends upon the specific purpose of memory and the features of that memory.

1) Flash Memory –
2) Cache Memory –
3) EEPROM –
4) SRAM –
5) DRAM –
6) Virtual Memory –

Memory Structure :-

Types of memories – depends upon the specific purpose of memory.

1) Main or Primary memory – contains the program currently being executed by CPU. Moderate speed and Small size. Usually Volatile. Mandatory.

2) Associative or Secondary memory – Programs currently not being used. Slowest but Large size. This is in the form of CD, DVD, ext hard disc etc. Non-volatile. Optional.

3) Cache memory – For storage of current and frequently required instructions of the program. Very fast but Smallest in size. Cache is used to reduce the average time to access data from the main memory. Acts as a buffer between CPU and main memory.

Memory Structure

3) Flash Memory –

Flash memory is non-volatile advanced type of "EEPROM" (Electrically Erasable Programmable ROM). Mostly used in Microcontrollers and other electronics devices to store the firmware.

1) Can be quickly erased and programmed, so called as Flash.
2) Performs high speed read, erase and write operations.
3) Has a limited number of erase and write cycles. (10000 times) This number is increasing day by day.
4) Smaller in size. Limited life.
5) Used in Cars, Cameras, Cell phones, Digital diaries etc.
6) Two types – NAND flash and NOR flash

Memory Structure :-

3) Cache memory – SRAM type - Cache memory has 3 levels. viz. L1, L2 and L3.

Level I Cache

Very Fast, very costly, lies inside the CPU.
Few KBs e.g. 32 or 64 KB

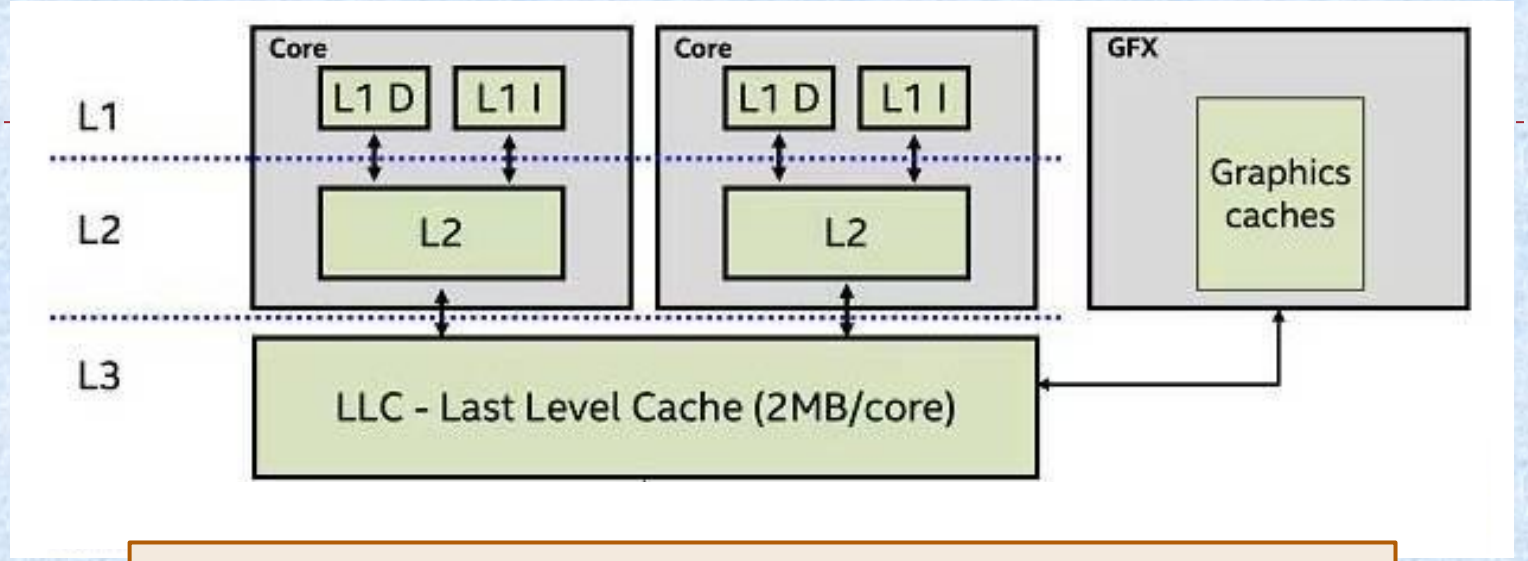
Level 2 Cache

Medium Fast, Moderate cost, lies inside or outside the CPU.
e.g. 256 KB or 512 KB.

Level 3 Cache

Slow , Low cost, lies outside the CPU, optional.
e.g. 8 MB, 16 MB etc.

But still all types are faster than RAM



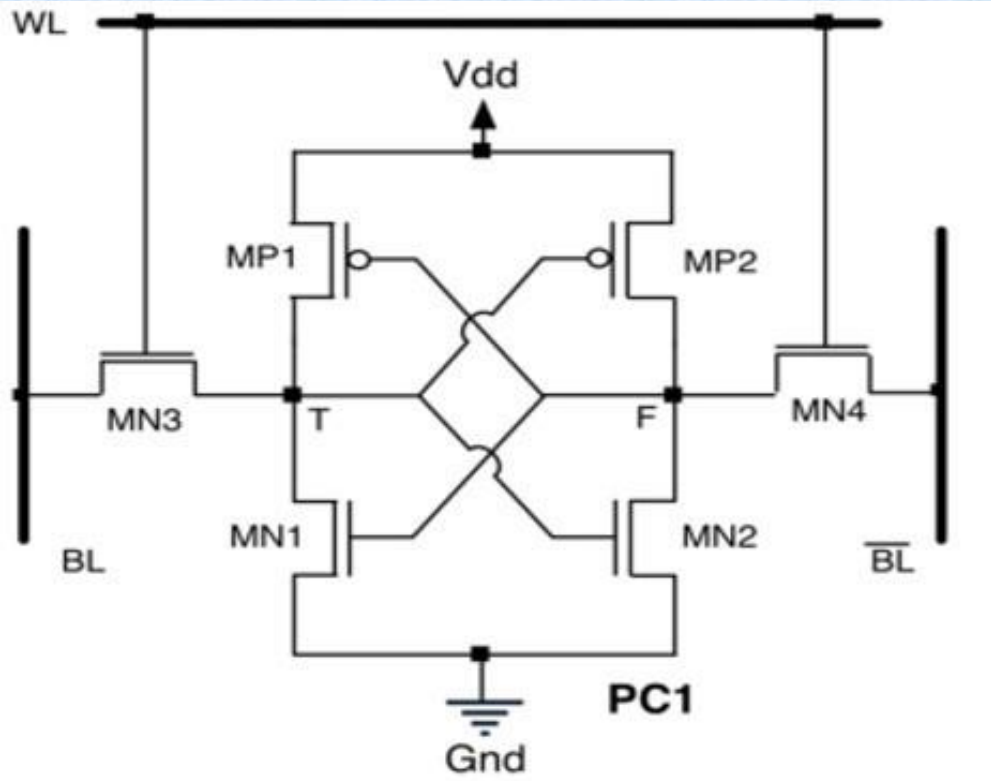
Cache memory – Relative size and location in CPU.

SRAM type - Cache memory - L1, L2 and L3.

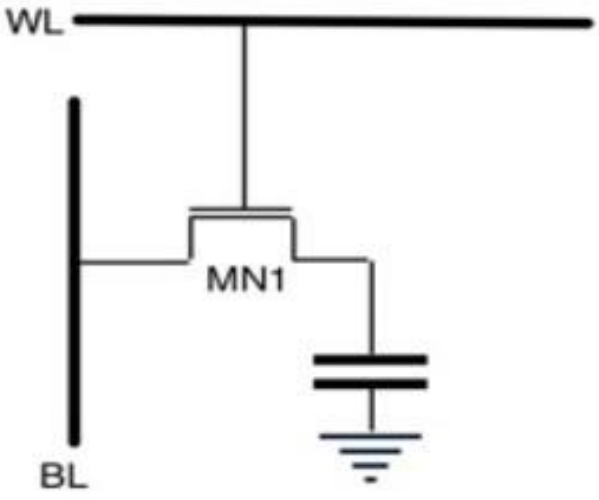
D – Data / I - Instruction

Memory Structure

Vishwakarma Institute of Technology
FY - Department of Engineering, Sciences and Humanities



(A) 6-transistors SRAM cell



(B) single-transistor DRAM cell

Memory Structure

SRAM (Static RAM)	DRAM (Dynamic RAM)
1) Used as Cache memory	1) Used as Main memory
2) Very fast speed	2) Fast
3) Costliest of all	3) Cheaper than SRAM
4) Low density (6 Tr per cell but has few cells)	4) High density (1 Cap per cell but has more cells)
5) High manufacturing cost	5) Low manufacturing cost
6) Data is stored in the form of voltage developed between two cross coupled 6 or 4 transistors forming an inverter.	6) Data is in the form of voltage across a charged Capacitor which gets slowly discharged.
7) Made up of transistors thus, periodic refreshing is not required, thus called as Static.	7) Made up of capacitors, so requires periodic refreshing. (few hundred times in a second) Thus called as Dynamic.
8) Low power consumption.	8) High power consumption.

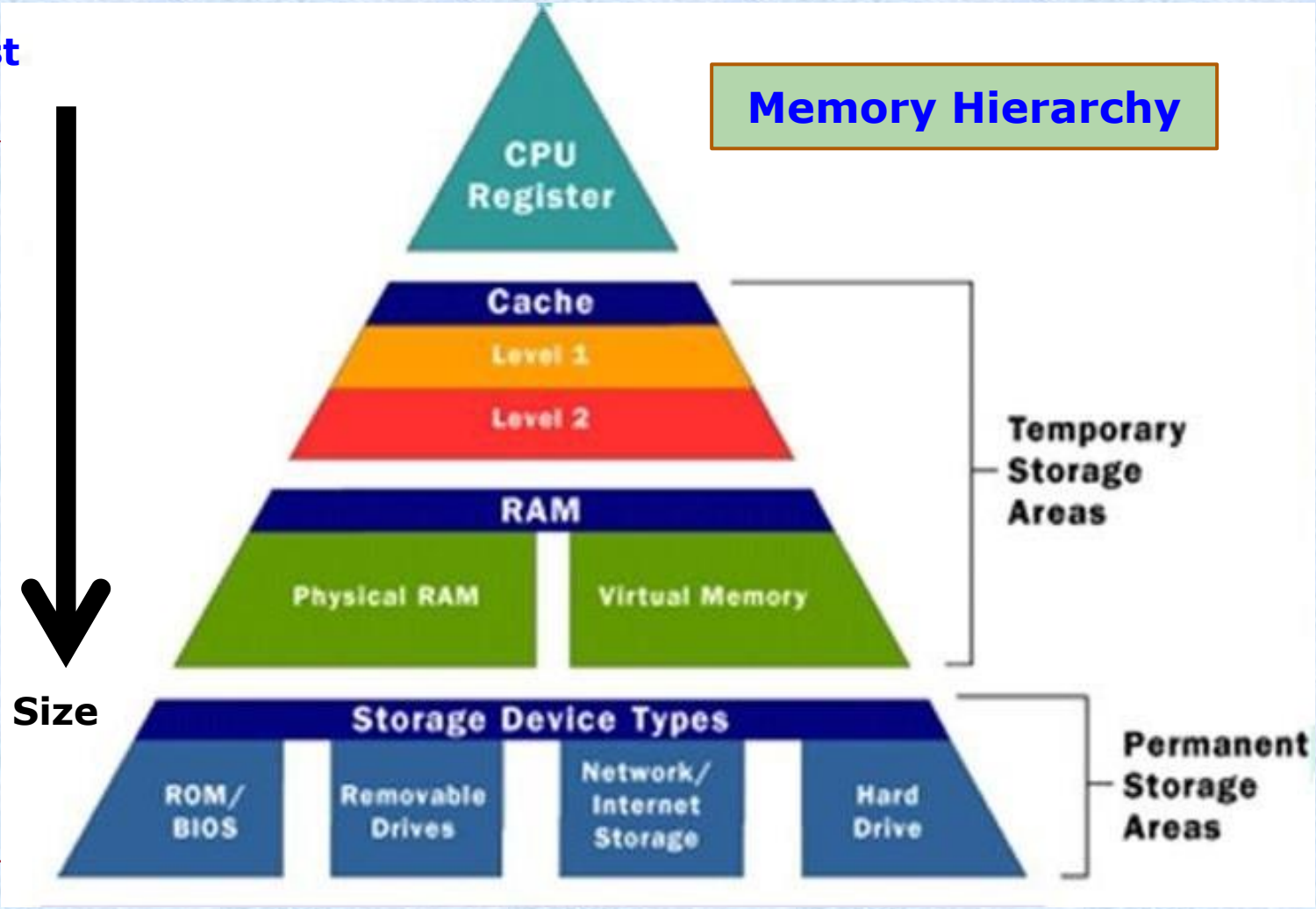
Memory Structure

Virtual Memory – Virtual memory is a feature of an operating system (OS) that allows a computer to compensate for shortages of physical memory by temporarily transferring pages of data from random access memory (RAM) to disk storage.

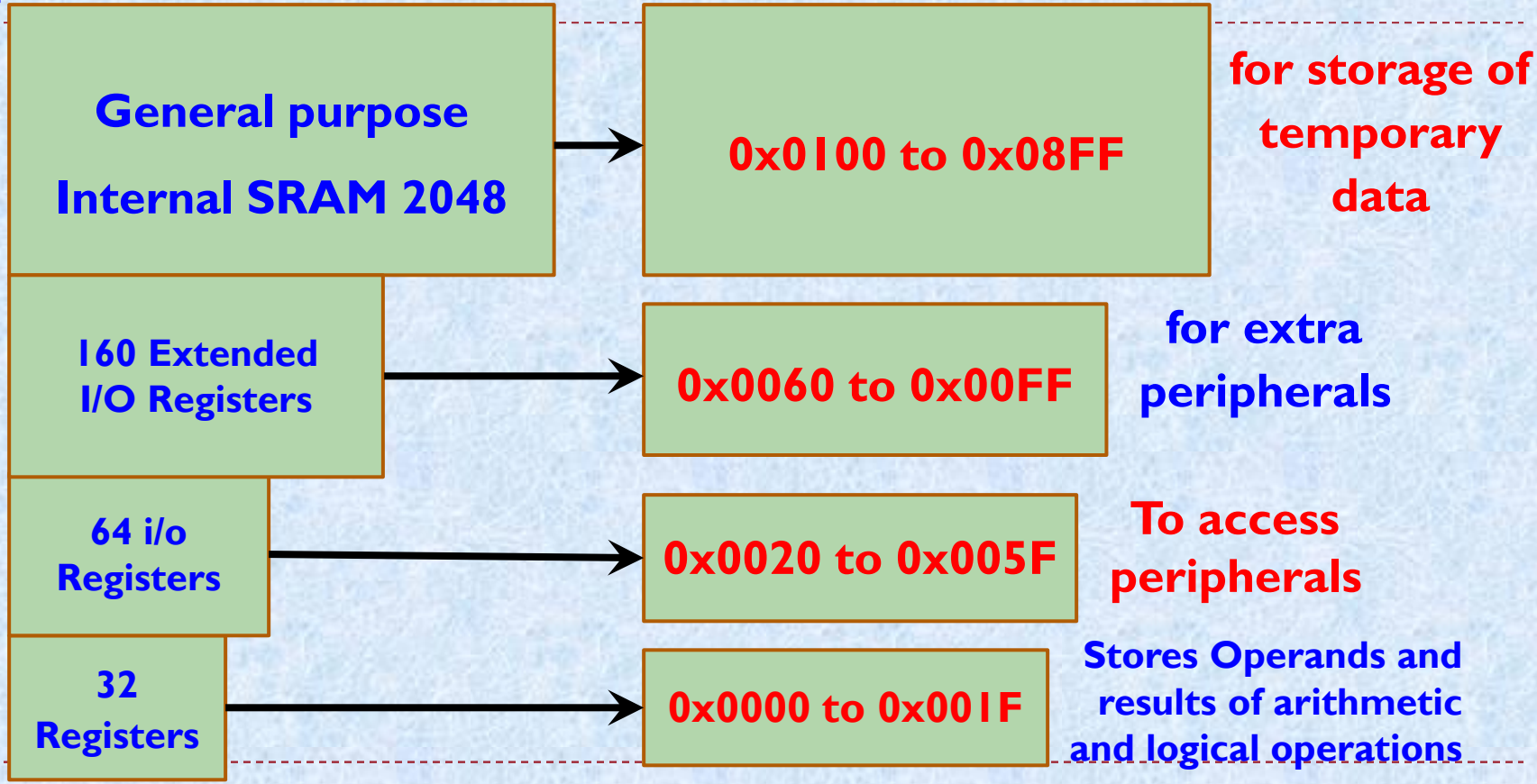
Virtual Memory is not included as it is not actual memory.

Virtual memory gives a feel / illusion of having a large RAM.

Memory Hierarchy



Memory Structure in ATMega328P



Register File of General purpose Registers of ATMega328P

Register	Address
R31	0x1F
R30	0x1E
R29	0x1D
R28	0x1C
R27	0x1B
R....	0x....
R....	0x....
R4	0x04
R3	0x03
R2	0x02
R1	0x01
R0	0x00

R0 to R31

0x00 to 0x1F

i.e. 32 locations

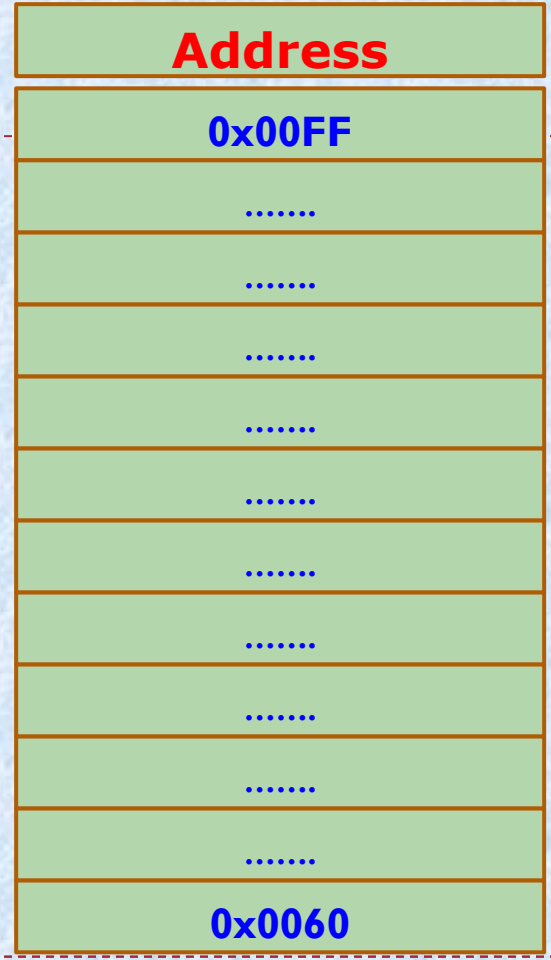
0x..... means it is a hexadecimal number

The X, Y and Z registers of ATMega328P can additionally work as 16 bit address pointers for indirect addressing to all the remaining registers in the register file. (R26 to R31)

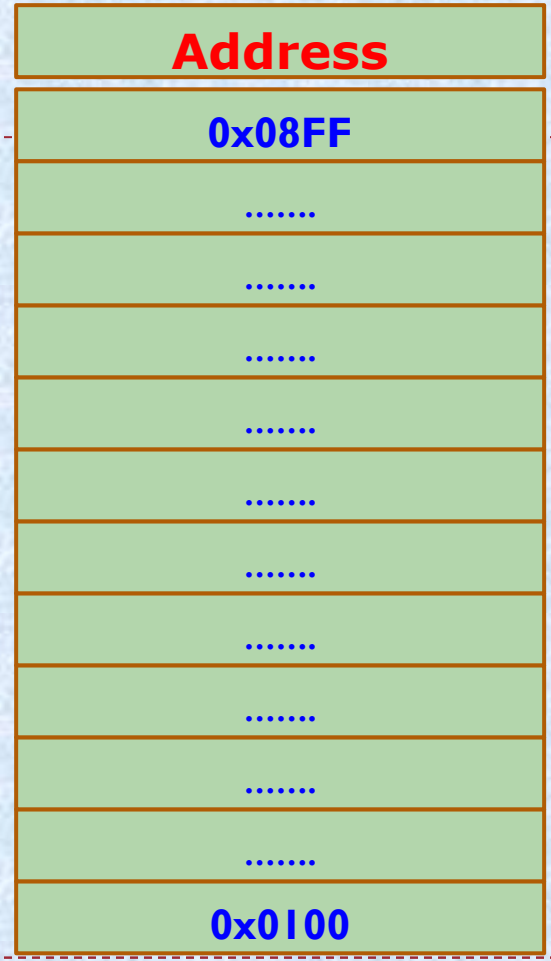
Register	Address
SREG	0x5F
SPH	0x5E
SPL	0x5D
OCRO	0x5C
.....	.....
.....	.....
.....	.....
.....	.....
TWDR	0x23
TWAR	0x22
TWSR	0x21
TWBR	0x20

Register File of SFRS - Special Function Registers of ATMega328P

32 to 95
i.e. next 64 locations
(0x20h to 0x5Fh)



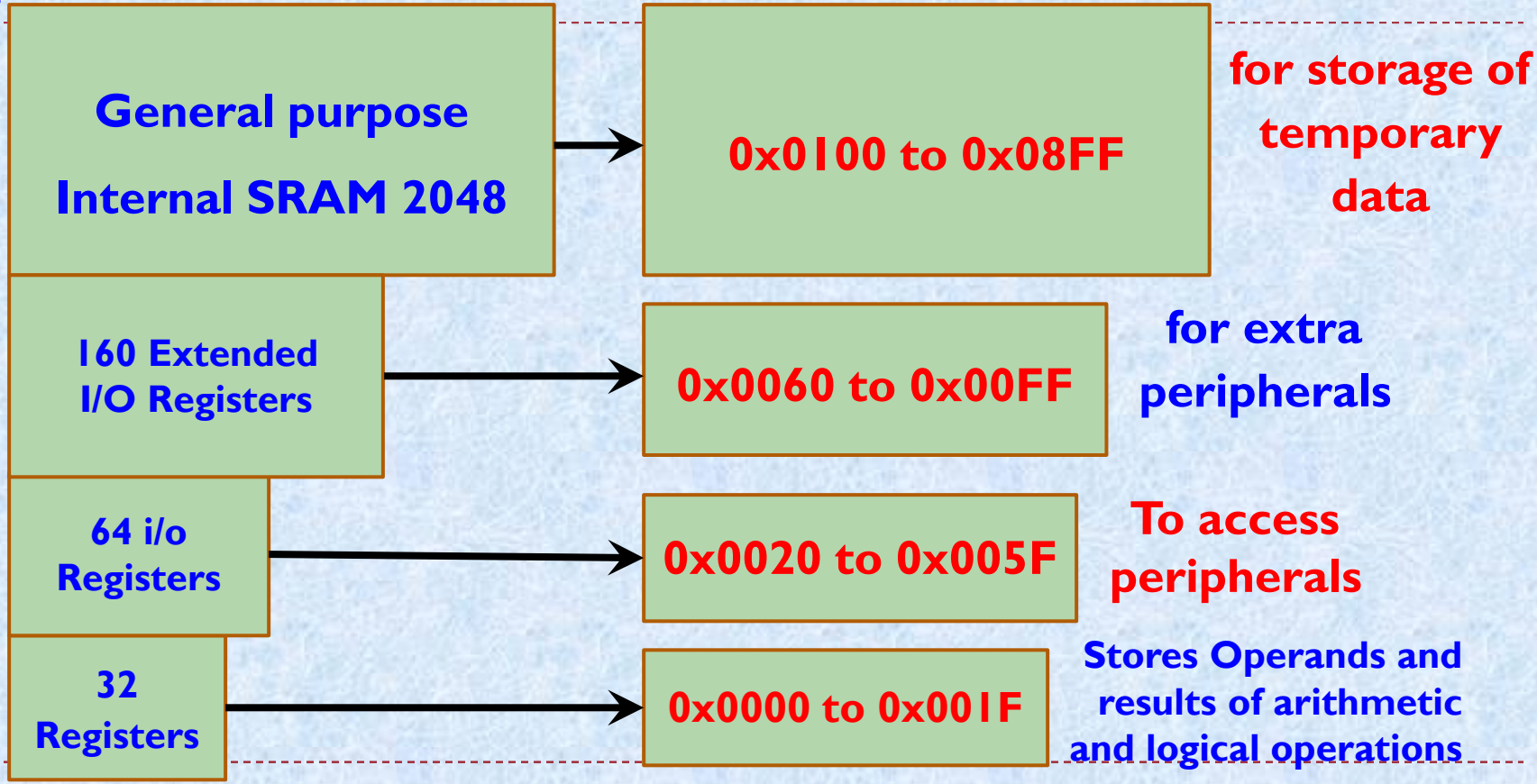
160 extended I/O registers for extra peripherals 0x0060 to 0x00FF



General Purpose SRAM of ATMega328P

0x0100 to 0x08FF = 2048

Memory Structure in ATMega328P – Data Memory Map



Addressing Modes of AVR (ATMega328P)

Addressing mode – It is a way to tackle / handle the Operand i.e. the Data or a way to deal with the Data.

1) Immediate addressing mode
2) Register addressing mode
3) Direct addressing mode
4) Register Indirect addressing mode
5) Register Indirect addressing mode with displacement
   Register Indirect addressing mode
   (with pre-decrement and post-increment)
6) I/O direct addressing mode

Addressing Modes of AVR (ATMega328P)

1) **Immediate addressing mode – Single register mode –**

deals with the contents of a single register directly.

Format → Instruction Register, (Number)

e.g.

LDI R18, 0x3C;	**Load immediately**
INC R20;	**Increment**
AND R22, 0b00110101	**ANDing**

Register should be between R16 to R31 only.

Thus LDI R12, 0x3C; is illegal

0 <= Number <= 255 i.e. FFh only (as 8 bit)

Thus LDI R18, 0x013C; is illegal

Addressing Modes of AVR (ATMega328P)

1) Immediate addressing mode – Single register mode –

```
LDI R22, 0x3C;
AND R22, 0b00110101      Find contents of R22
```

```
        00111100
        00110101
--------------------------------
        00110100         = 34h = answer
```

```
LDI R22, 0x3C;
OR R22, 0b00110101       Find contents of R22
```

```
00111101                 3D = contents of R22
```

Addressing Modes of AVR (ATMega328P)

2) Register addressing mode (direct) –

A) deals with the contents of two registers.

Format → Instruction Register 1, Register 2

ADD R18, R19; Add contents of R19 to R18, result stored in R18

SUB R20, R21; Subtract contents of R21 from R20, result in R20

MOV R22, R23; copy contents of R23 to R22

Register should be between R16 to R31

0 <= Number <= 255 (as 8 bit)

Addressing Modes of AVR (ATMega328P)

3) Direct addressing mode –

B) Deals data between a register and a memory location.

Format → **Instruction Register, Address**

OR **Instruction Address, Register**

e.g.

LDS R20, 0x0045 — data at memory location 0045H will be loaded into R20 (**L**oad from **D**ata **S**pace)

STS 0x0045, R20 — Contents of R20 will be stored at memory location 0045H (**ST**ore direct to Data **S**pace)

Addressing Modes of AVR (ATMega328P)

3) Direct addressing mode –

LDS 0x0095, 0x0045 → what is meaning of this ?

This is illegal, as a *data can not be directly loaded into a memory location. OR can not be moved from one memory location into another memory location.*

It has to be transferred through a register only !!

Addressing Modes of AVR (ATMega328P)

LDS R22, 0x0045 → what is meaning of this ?

LDI R22, 0x0045 → what is meaning of this ?

LDS R22, 0x0045

(Load from data space)

→ Data is at memory location 45H which will be moved into R22.

LDI R22, 0x0045

(Load immediate)

→ Actual data is the number 45H which will be moved into R22

Addressing Modes of AVR (ATMega328P)

4) Register indirect addressing mode – uses X, Y or Z registers as pointers to indicate the memory location where data is stored.

e.g.

LDI XL, 0x40	**→ 40H is loaded in lower of X**
LDI XH, 0x02	**→ 02H is loaded in higher of X**
LDS R20, X	**→ (Load indirect) contents of memory location 0240H are copied into R20**

Addressing Modes of AVR (ATMega328P)

4) Register indirect with displacement :-

e.g.

LD R18, Y → control will go to Y, (consider YL and YH)

(Load Indirect)

find the 16 bit number in it,

treat it as address,

find the data at that address and

load that data in R18.

LDD R19, Y+0x10 → Load contents of memory location Y+0x10 to R19

(Load Indirect with Displacement)

STD X+0x05, R20 → Contents of R20 are stored at memory location X+0x05

Addressing Modes of AVR (ATMega328P)

Find the output of ....

Answer = R18 = 2Bh **R22 = 2Bh**

Y = 0135h **0x0135 = 2Bh**

e.g.

LDI R22, 0x2B	→	**??**	**(**Load immediate**)**
LDI YL, 0x35	→	**??**	
LDI YH, 0x01	→	**??**	
STS 0x0135, R22	→	**??**	**(**Store Direct to SRAM**)**
LDS R18, Y	→	**??**	**(**Load Direct from SRAM**)**

What are contents of R18, R22, Y, 0x0135 after execution of all the instructions above ?

What is the actual data in each location and register ?

Addressing Modes of AVR (ATMega328P)

How to transport a data of 300 different nos. to 300 consecutive memory locations ?

How to do this ?

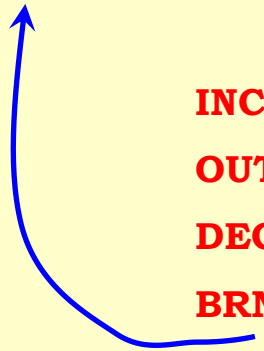
5) A) Register indirect with post-increment –

5) B) Register indirect with pre-decrement –

Addressing Modes of AVR (ATMega328P)

5) A) Register indirect with post-increment –

```
        LDI  R16, 0x20      (for e.g. counter set = 20)
        LDI  R20, 0xFF
        OUT  DDRB, R20      (All pins of Port B defined as output)
        LDI  ZL, 0x90       (Lower byte of Z = 90)
        LDI  ZH, 0x00       (Higher byte of Z = 00)
L1:     LDS  R20, Z         (Go to memory location 0090 pointed by Z and copy the
                             data to R20)
        INC  ZL             (increment pointer: 90 → 91)
        OUT  PORTB, R20     (contents of R20 i.e. data is sent to Port B)
        DEC  R16            (decrement counter: 20 → 19)
        BRNE  L1            (if R16 is not equal to 0, continue the loop –
                                    BRNE = Branch (Loop) if not equal to 0)
```



Addressing Modes of AVR (ATMega328P)

5) A) Register indirect with pre-decrement / increment –

LD R17, – Z : Data in Z will be decremented first and then loaded in R17

e.g. suppose Z contains a 16 bit address. Just e.g. Z = 0234h.

Thus, data at the address 0233h will be loaded in R17

After this, Z will have 0233h in it. (which is actually an address)

LD R17, + Z : Data in Z will be incremented first and then loaded in R17

e.g. suppose Z contains a 16 bit address. e.g. Z = 0234h.

Thus, data at the address 0235h will be loaded in R17.

After this, Z will have 0235h in it. (which is actually an address)

Addressing Modes of AVR (ATMega328P)

5) B) Register indirect with post-increment / decrement –

LD R17, Z + : Data in Z will be loaded in R17 and then incremented by 1.

e.g. Suppose Z contains a 16 bit address. e.g. Z = 0234h.

Thus, data at the address 0234h will be loaded in R17.

After this, Z will have 0235h as data in it (which is actually an address)

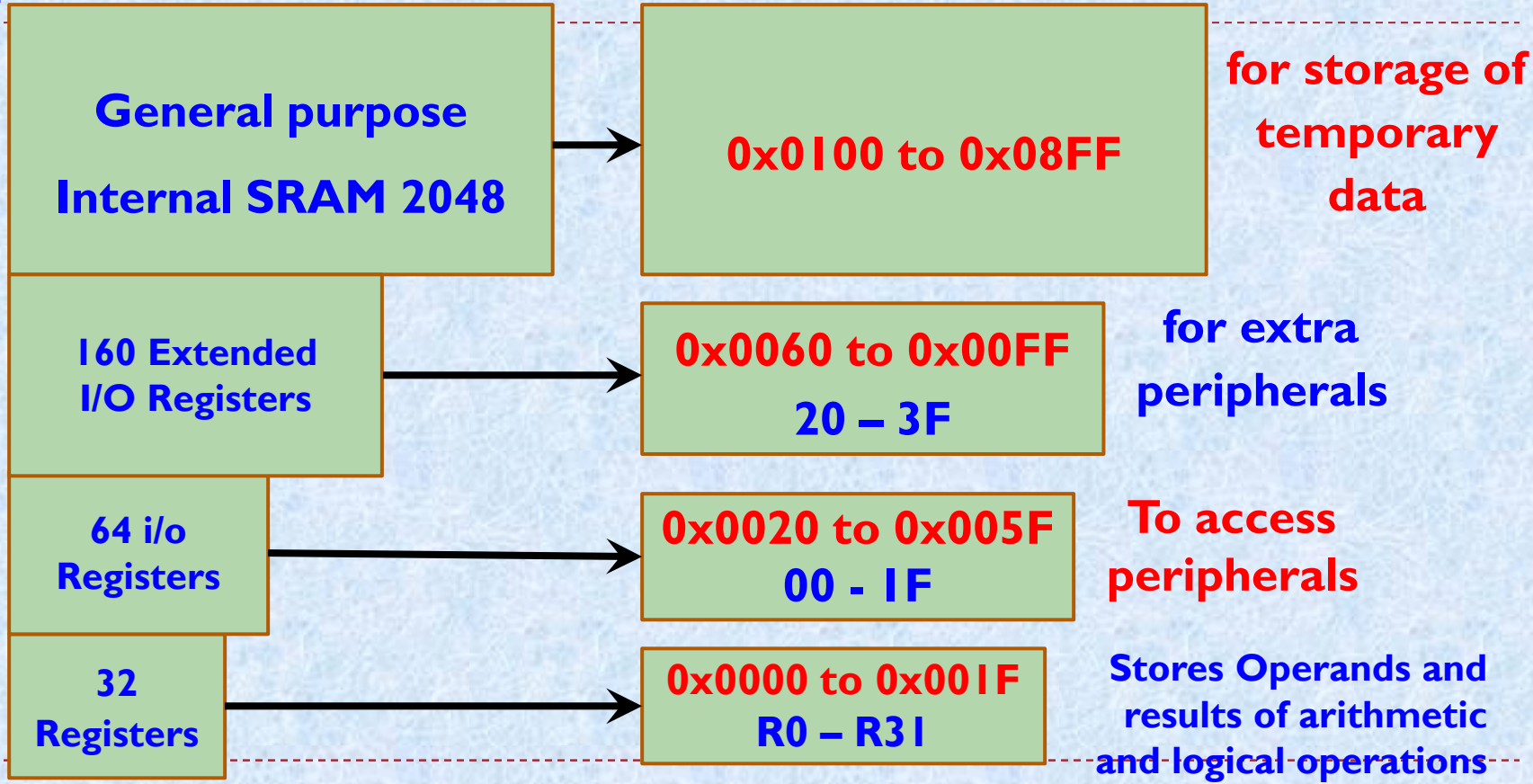
LD R17, Z – : Data in Z will be loaded in R17 and then decremented by 1.

e.g. Suppose Z contains a 16 bit address. e.g. Z = 0234h.

Thus, data at the address 0234h will be loaded in R17

After this, Z will have 0233h as data in it (which is actually an address)

Memory Structure in ATMega328P



Addressing Modes of AVR (ATMega328P)

6) I/O direct addressing mode – This is a Special addressing mode to access the 64 i/o registers in the ATMega328P from 0x0020 to 0x005F.

Only IN and OUT instructions are used.

To understand the **I/O direct addressing mode**, we should first know **DDR, PORT** & **PIN** registers.

The 3 I/O registers - DDRx, PORTx and PINx

There are 3 very important registers used to handle / control the i/o operations of different ports using GPIO pins. They are called as i/o registers.

1) Data Directions Register – DDRx
2) Port Output Register – PORTx
3) Pin Input Register – PINx

…… where x is the name of the port (B, C or D)

Data Directions Register – DDRx of ATMega328P

Data Directions Register – DDRx (....x is the port B, C or D)

All the digital pins can work as an input pin or an output pin.

DDRx will decide whether a pin would be used as i/p pin or o/p pin.

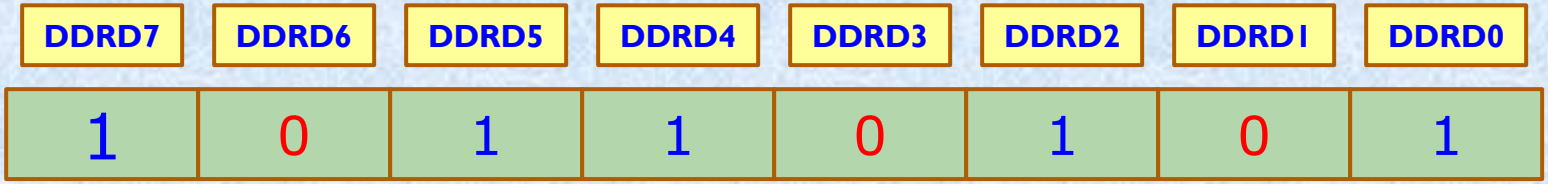
If the corresponding bit in the DDR is HIGH (1), the pin will work as OUTPUT pin – **eligible to Deliver the data.**

If the bit in the DDR is LOW (0), the pin will work as INPUT pin – **eligible to Accept the data.**

DDR of ATMega328P

e.g. DDRD = 0xB5; A Hexa no. B5 is pushed in DDRD OR

DDRD = 0b1011 0101; A binary no. 10110101 is pushed in DDRD



Direction of Data will be decided by DDRD

1 = output pin = deliver the data 0 = input pin = accept the data

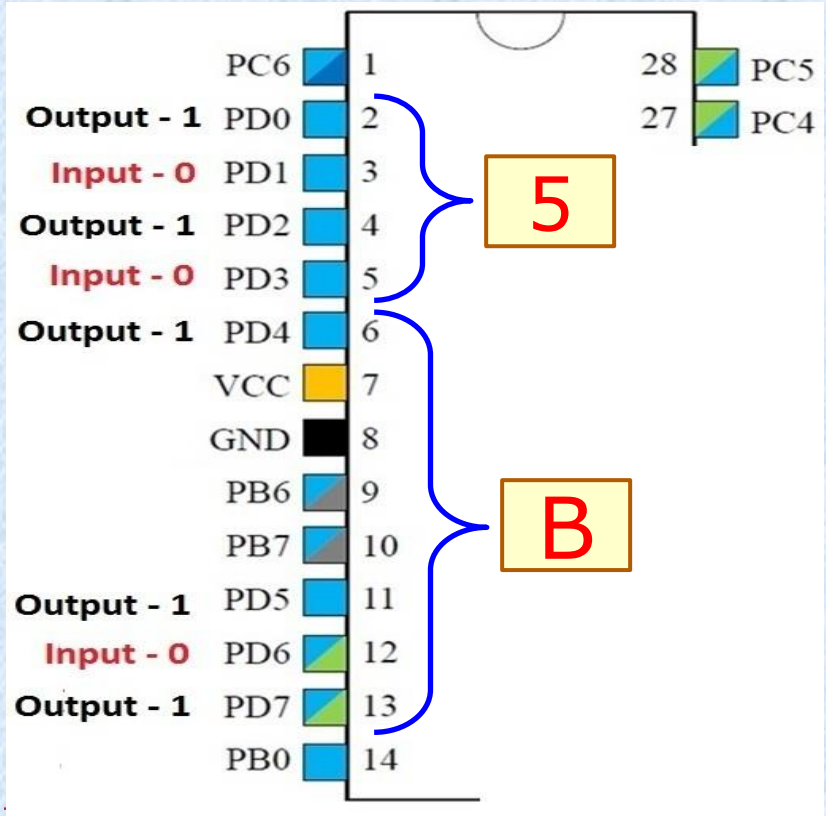
output and input w.r.t. whom ??

DDR of ATMega328P

DDRD= **B** | **5**

1	0	1	1	0	1	0	1

B5 in DDRD will make all 8 pins of Port D to work as i/p or o/p pins as per the "0" or the "1"



PORTx and PINx registers of ATMega328P

Once the DDR decides whether a pin is configured for outputting the data or inputting the data ……

then the PORTx register decides whether a pin should *deliver* a *HIGH* output or a *LOW* output.

Similarly ……. the PINx register decides whether a pin should *accept* a *HIGH* input or a *LOW* input.

DDRB	**DDRC**	**DDRD**
PORTB	**PORTC**	**PORTD**
PINB	**PINC**	**PIND**

PORTx of ATMega328P

e.g. PORTD = 0xA6; A Hexa no. A6 is pushed in PORTD OR

PORTD = 0b1010 0110; A binary no. 10100110 is pushed in PORTD

PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
1	0	1	0	0	1	1	0
High	Low	High	Low	Low	High	High	Low

What is the Direction of the above Data ?

The direction will be decided by the DDRD.

(1 = output pin = deliver the data)
(0 = input pin = accept the data)

PINx of ATMega328P

e.g. PIND = 0xC7; A Hexa no. C7 is pushed in PIND OR

PIND = 0b1100 0111; A binary no. 1100 0111 is pushed in PIND

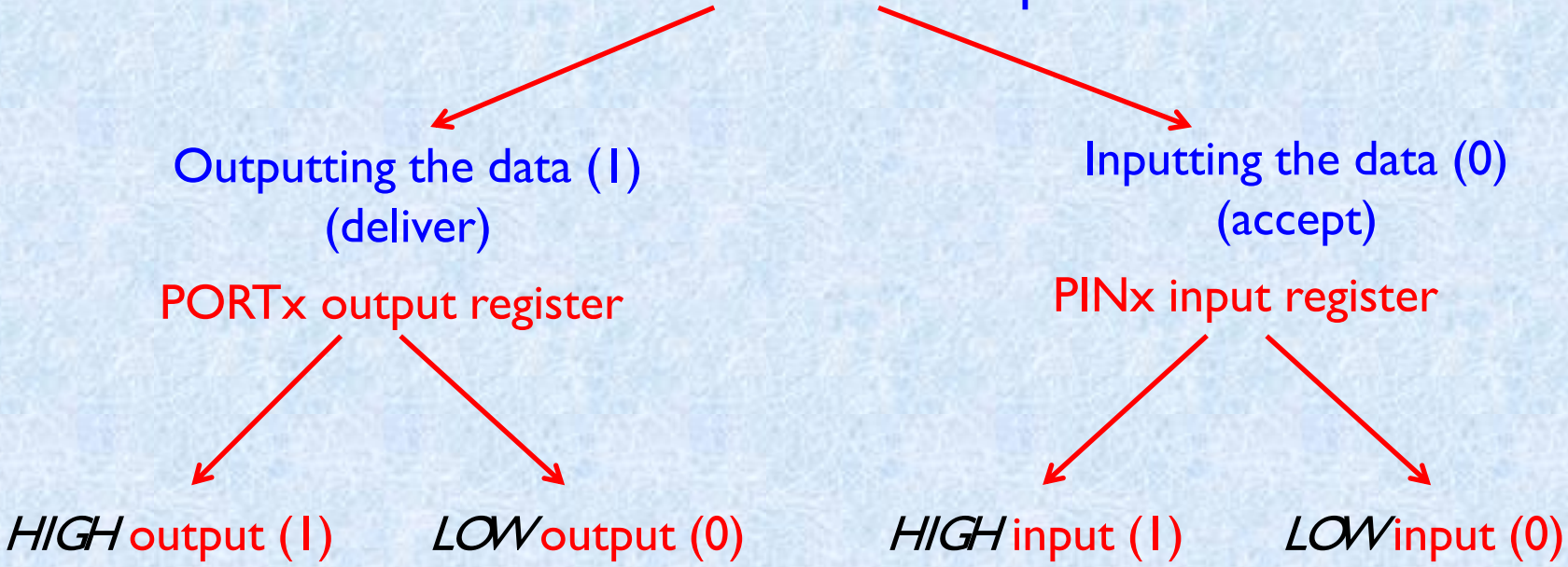
PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
1	1	0	0	0	1	1	1
High	High	Low	Low	Low	High	High	High

What is the Direction of the above Data ?

The direction will be decided by the DDRD.

(1 = output pin = deliver the data)
(0 = input pin = accept the data)

PORTx and PINx registers of ATMega328P



PORTx of ATMega328P

e.g. 1

DDRD (1 = output pin = deliver the data) (0 = input pin = accept the data)

DDRD = 0b 1 0 1 1 0 0 0 1

PORTD = 0b 1 0 0 1 0 0 0 1

1 = High 0 = Low

OUTPUT = ?? = 0b 1 * 0 1 * * * 1

e.g. 2

DDRD = 0b 1 0 1 1 0 0 0 1

PORTD = 0b 1 0 0 1 0 1 0 1

OUTPUT = ?? = 0b 1 * 0 1 * * * 1

DDRx will decide the direction and PORTx will decide High o/p or Low o/p.

PINx of ATMega328P

e.g. 1

DDRD
(1 = output pin = deliver the data)
(0 = input pin = accept the data)

DDRD = 0b 1 0 1 1 0 0 0 1
PIND = 0b 1 0 1 0 1 0 1 0

1 = High 0 = Low

INPUT = ?? = 0b * 0 * * 1 0 1 *

e.g. 2

DDRD = 0b 1 1 1 1 0 0 0 0
PIND = 0b 1 1 1 1 1 1 1 1
INPUT = ?? = 0b * * * * 1 1 1 1

PINx can *not change* the properties of the pins which are already decided by the DDRx. MuC can read only from those pins which are decided as i/p pins.

*** pins will enter in Tri state.**

Addressing Modes of AVR (ATMega328P)

6) I/O direct addressing mode – This is a Special addressing mode to access the 64 i/o registers in the ATMega328P from 0x0020 to 0x005F.

Only IN and OUT instructions are used.

Addressing Modes of AVR (ATMega328P)

Example –

Read the data and store in memory

IN R6, PIND : Data coming at PIN register of D will loaded in R6

STS 0x0200, R6 : This data will be stored at the memory location 0x0200

Read the data and write to a Port

IN R7, PIND : Data coming at PIN register of D will loaded in R7

OUT PORTB, R7 : This data will be written at the Port B as output.

Q.1) In order to run above instructions, what is a pre-requisite ?

Q.2) What is the role of DDRD and DDRB ?