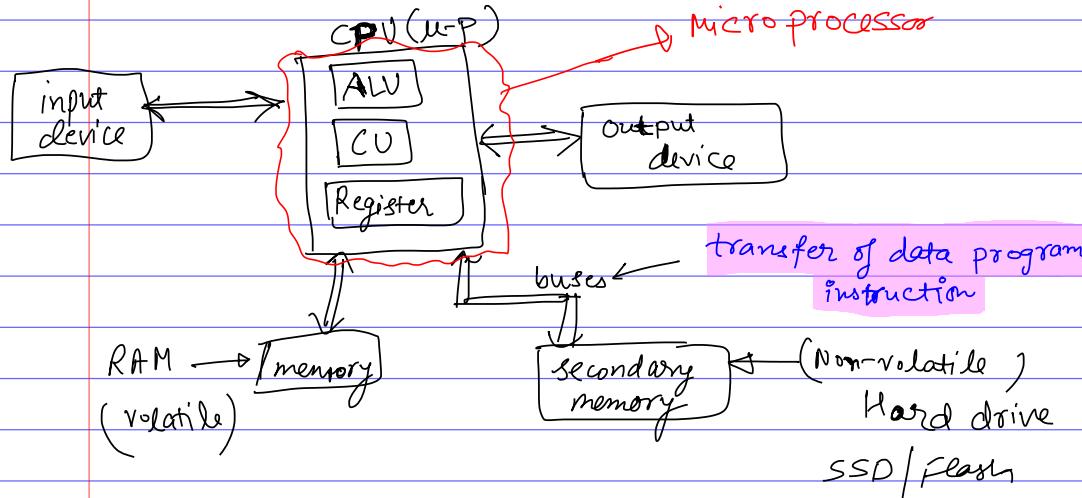


EE 309

- Microprocessors

- Shalabh gupta

Computer



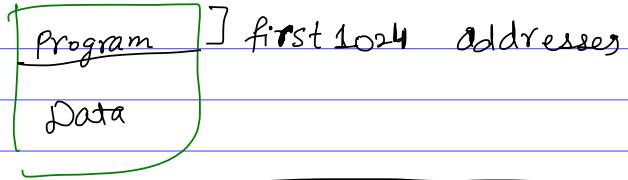
first commercial micro-processor

400 4 → calculators & Arithmetic ops.

4 bit processor, 12-bit address
→ nibble

1024 → 2048 bytes

example



Quizes → 30% (every monday morning) best n-1
of n quizzes

Midsem → 30%

Endsem → 40% (entire syllabus)

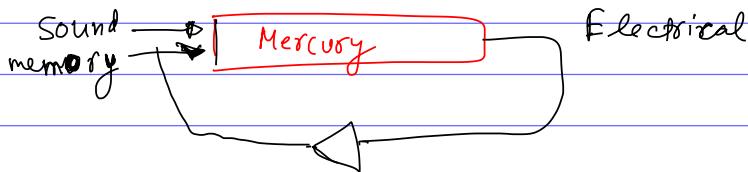
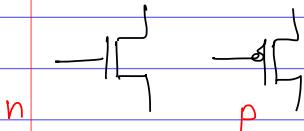
Shalabh @ ee.iitb.ac.in

ENIAC: Electronic Numeric Integrator and computer

MP: # of transistors

in 4004: 2300

Today: ~ 100 billion (10^{11})



Computation power for same electrical power
↑ $\sim 10^{16} - 10^{17}$
55 yrs.

5 10 digit decimal additions = 1 FLOP
Improvement = 3×10^3

Moore's law - scaling in chips

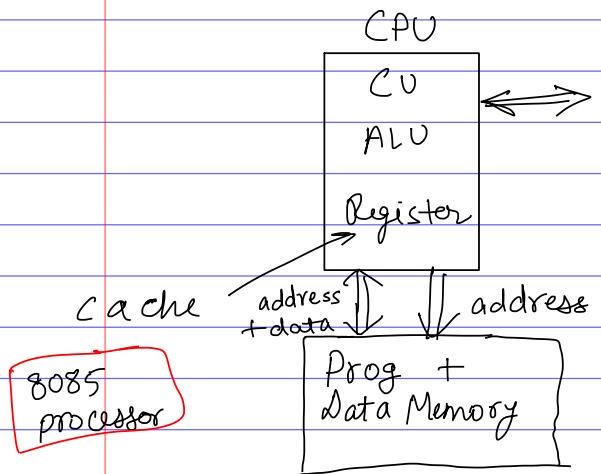
→ no. of transistors on microchips doubles every 2 yrs.

Min^m feature size (MFS) = 10 μm (here)

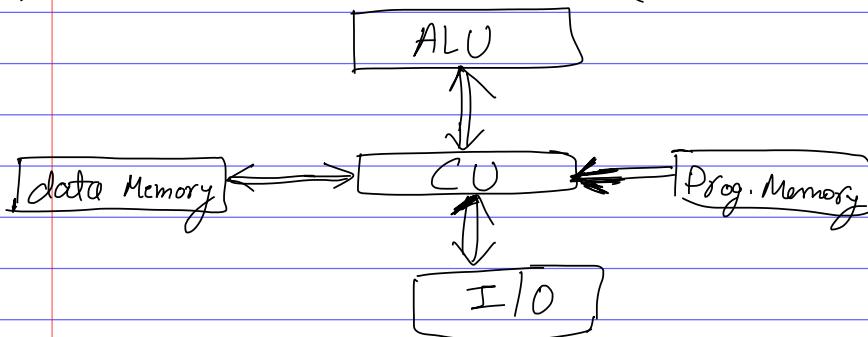
2 types of processor architecture

1) Von-Neumann architecture

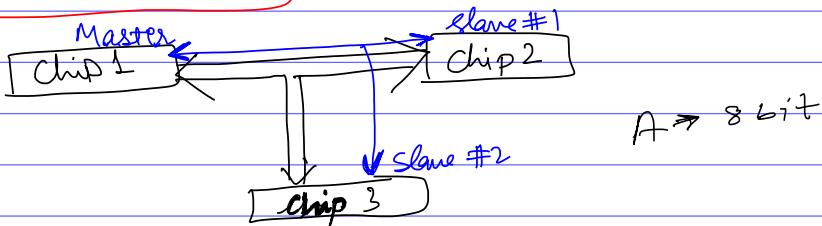
generally used in
MP

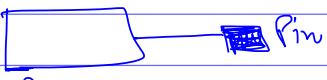


2) Harvard Arch. (used in mc)



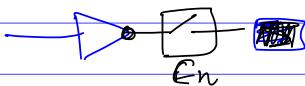
→ 8051 micro controller



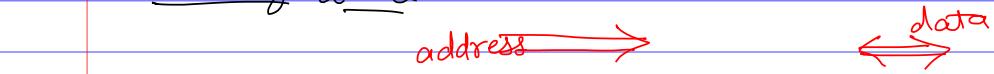


Port

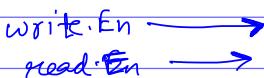
tristate Buffer



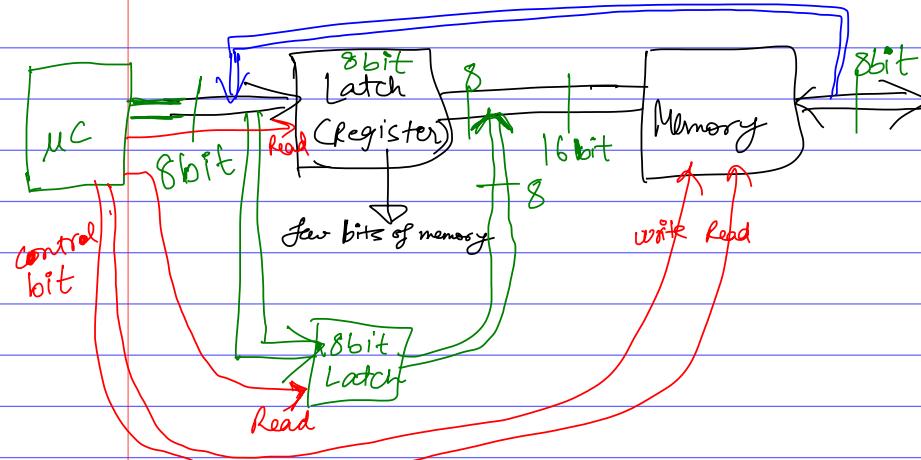
Memory device



Write

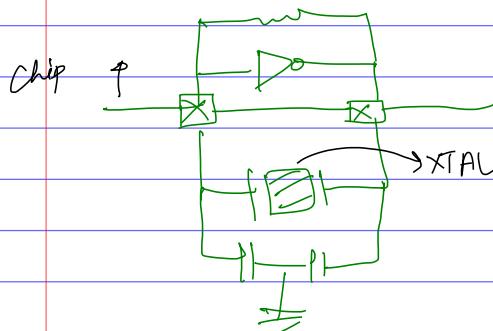


- ① Set the correct address.
- ② Select the correct data
- ③ Strobe write·En.



9 Jan, 25

lec - 3



32 byte / 4 bit

4 bit - nibble

$$\begin{array}{r} 0101 \\ \hline 5 \end{array} \quad \begin{array}{r} 1001 \\ \hline 9 \end{array}$$

0x59

0101	(10)	0	8
<u>5</u>	<u>D</u>	1	9
0x5D		2	A (lo)
		3	B (hi)
		4	C
		5	D
		6	E →
		7	F → 15

Instruction length \rightarrow fixed / variable

8051 \rightarrow min length = 1
variable

Ports:



Configuration

\rightarrow output

\rightarrow input

tristate \rightarrow 'Z'

Low - '0'

High - '1'

ideally tri-stated if no driver

{ No outputting port }

weak

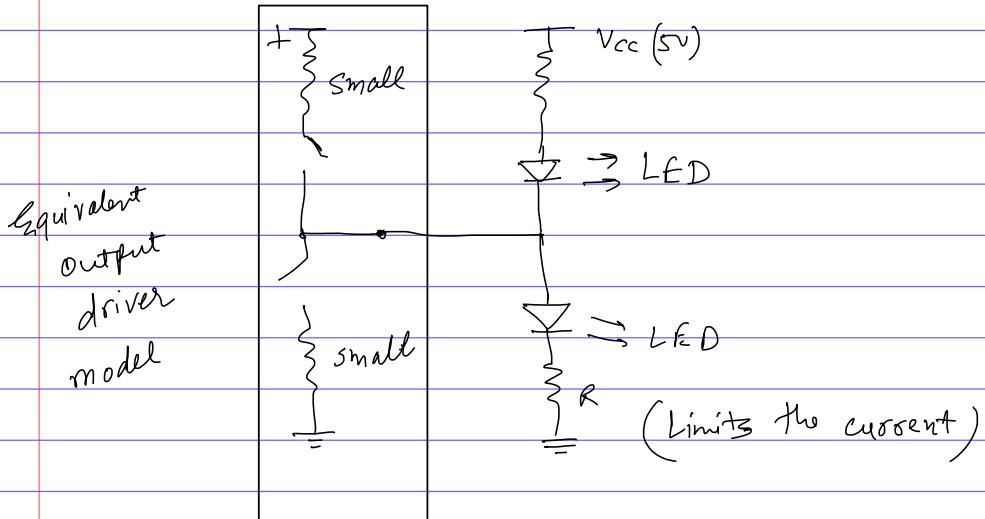
pull up

or pull down

T

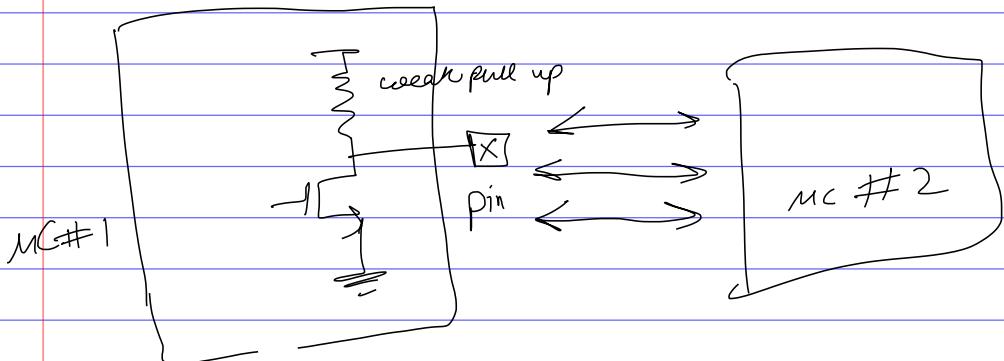


Ideal output driver: Low output Impedance



Some ports $\rightarrow 0, 1, 2$
other $\rightarrow 0, 1$, "weak internal pull-up".

I/O Port Pins:

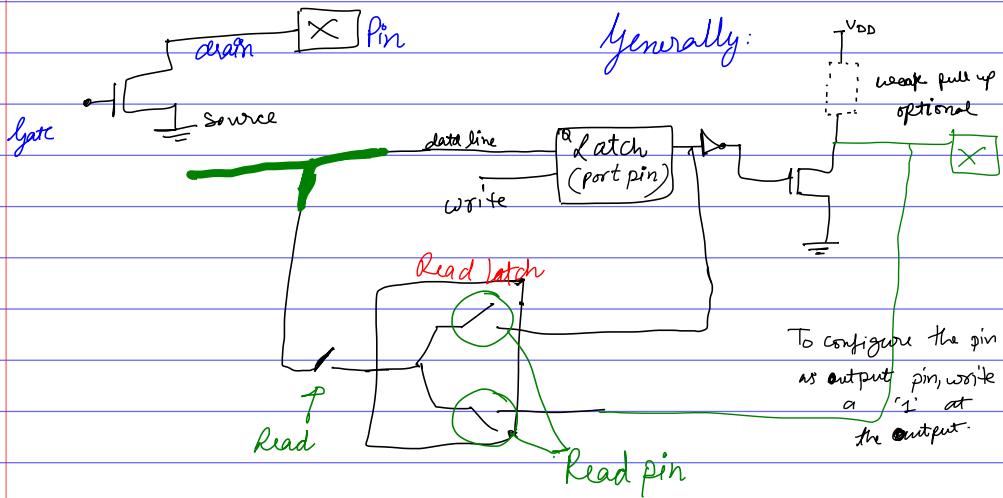


nmos - good low voltage conductor

pmos - good high - -

Example configuration

open drain out put



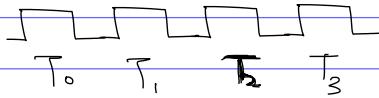
Instructions

T-states

clock 12 MHz

Machine cycles

12 in 8051 uc



8051 → 4 kB ROM internal (program storage)
128 byte RAM (data storage)

8 bits-I/O ports

16 bits up timer (delay)

Serial port

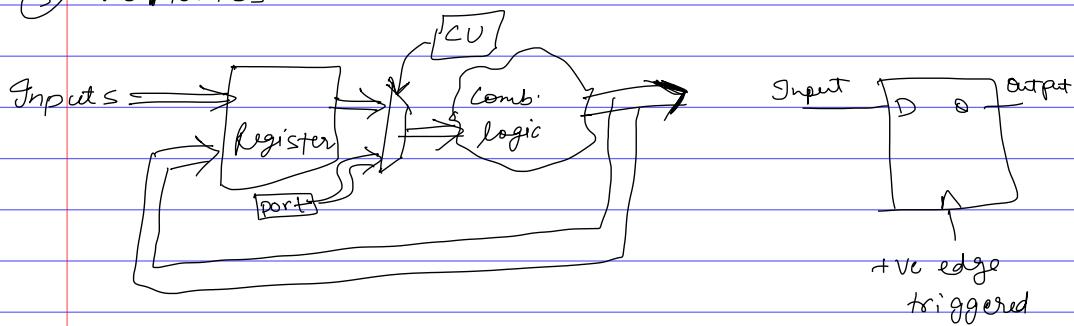
5 interrupts at priority level

2 power saving mode

Digital System

Building Block

- ① Combinational logic → Nand, Nor, XOR, encoder, decoder, adder, other logic gate
- ② Sequential logic → latches, FFs, registers
- ③ Memories



14 Jan, 25

Memory in 8051:

→ Harvard architecture:

↳ Program memory: $0000H \rightarrow OFFFH$ (4KB) ROM
 EEPROM → ROM - Read only $1000H \rightarrow FFFFH$ (Another 60KB) Program
 non-volatile External, if needed RAM | ROM address

$$k = 10^3$$

$$K_2 = 2^{10} = 1024$$

↳ Data memory: Up to 256 byte(B) (internal)
 - - - 64 KB (external)

internal RAM:

$0H - 7FH$ (128 B typically)

$0H \rightarrow FFH$ (256 B in some devices)

Some of these locations

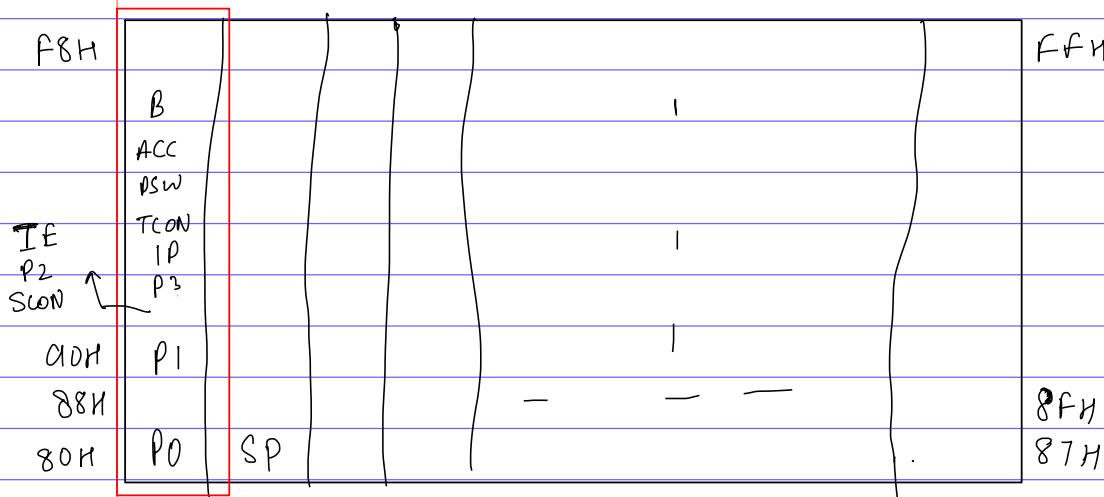
$80H - FFH$

have special function Registers (SFRs)

$$00H \rightarrow 0FH : R0 \rightarrow R7$$

SFRS

→ not all of these addresses are occupied.



Registers are bit addressable

each memory - 8bit

1 bit
8 diff. code value

Addressing Mode:

T ADD

AOD

A D D

~~ADD~~ A, @ R_i

Register Addressing $A \leftarrow A + \text{Contents of } R_n$

no. of opcodes

16 Jan, 25

2 - MOVX @ R_i, A [1 byte]

2 - MOVX A, @ R_i [instruction]

@ DPTR: DPH, DPL (16 bits)

1 - MOVX A, @ DPTR [1 byte]

R_i: R₀ or R₁

1 - MOVX @ DPTR, A [inst]

R_n: R₀ → R_n

Suppose

R₀ → 45H

MOVX @ R₀, A

Store constant of A to

location of 45H

Opcode: Operation to be performed such as addition, subtraction or - -

Operands: Specify data on which operation is performed

Addressing Mode: How operand are accessed.

e.g. immediate, register, direct & indirect address.

Instruction format 1/2/3 byte instruction length

ADD A, R_n 1 byte

MOV A, 30H 2 byte

The value @ 30H is moved to A

MOV A, #30H 2 byte

#30H moved to A

L JMP 1234H 2-3 bytes

In 8051: each inst: 1 byte \rightarrow 3 bytes

Duration: 1 MC \rightarrow 4 MC

Machine cycle = 12 clock cycle

Typically 12 MHz crystal
 \Rightarrow 1 MC = 1 μ s duration

4 MC instruction:

Types of Instruction

MUL

Data Transfer

DIV

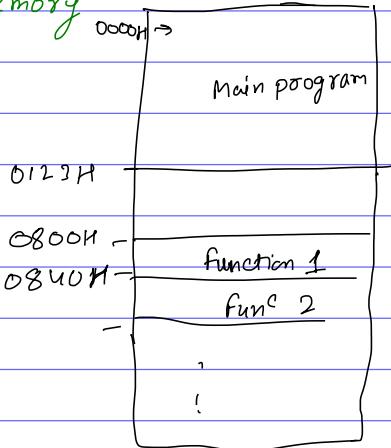
Arithmetic

Logical

Boolean

Program Branching

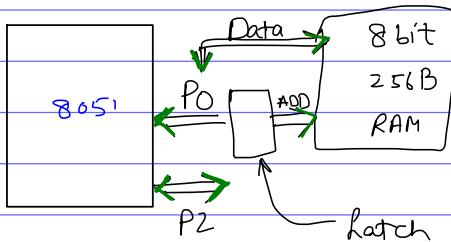
Program Memory



20 Jan, 25

(P2) MOVX A, @DPTR
MOVX A, @Ri } external

1 Byte instruction
MOVX A, @Ri or @Ri, A



2 machine cycle
(24 T-state)

MC #1 : Write addresses at P0, P2

MC #2 : Write or read 8-bit data to or from P0.

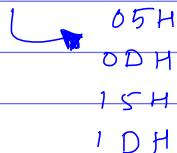
PSW (Program Status Word) SFR

CY	(PSW.0)	- Carry flag
AC	(PSW.1)	- Auxiliary carry
F0	(PSW.2)	- General Purpose flag
RSI	1	
RS0	1	→ Selects Register Bank
OV		
P		→ overflow
		→ Parity

R0 - R7

		RS1	RS0
00H - 07H	0	0	
08H - 0FH	0	1	
10H - 1FH	1	0	
18H - 1FH	()		

MOV A, R5



 05H
 0DH
 15H
 1DH

$R_i:$ 00H, 01H
 08H, 09H
 10H, 11H
 18H, 19H

Location

Example:
 01H has 45H.
 11H has 34H.
 34H has 12H.
 45H has 05H

MOV A, @R1

$\Rightarrow A = 12H$

MOV A, R1

$\Rightarrow A = 34H$

$ADDC A, \#F5H \Rightarrow A \leftarrow A + C + F5H$


PUSH - Store <src> value to the top of the stack.

POP - Retrieve <src> values

SP - Stack pointer points to memory location at
 the top of stack

21 Jan, 25

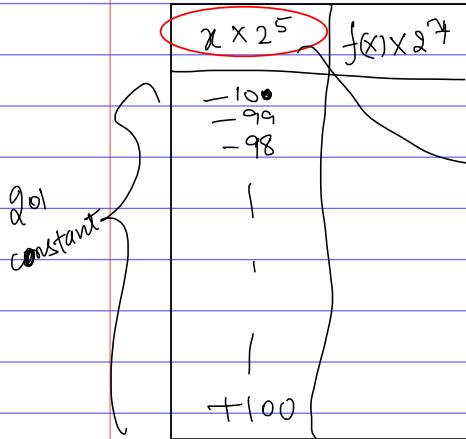
XCH - exchange

XCHDigit - exchange last 4 bit

Look up Table

$$f(x) = \sin x$$

$$\begin{aligned} x &\rightarrow -\pi \rightarrow \pi \\ x \cdot 2^5 &\rightarrow \sim -100 \rightarrow 100 \end{aligned}$$



MOVC : MOV Constant

map to $\begin{aligned} A &\leftarrow @A + DPTR \\ A + DPTR & A \leftarrow @A + PC \\ \text{or } A + PC \end{aligned}$

TABLE: MOVC, @A + PC

RET

if $A = 10$

q^{th} entry \rightarrow

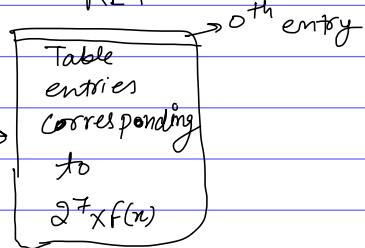


TABLE1: MOVC A, @A+PC

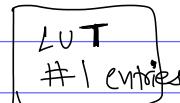
RET \leftarrow opcode for RET

0 \leftarrow 00H

1 \leftarrow 01H

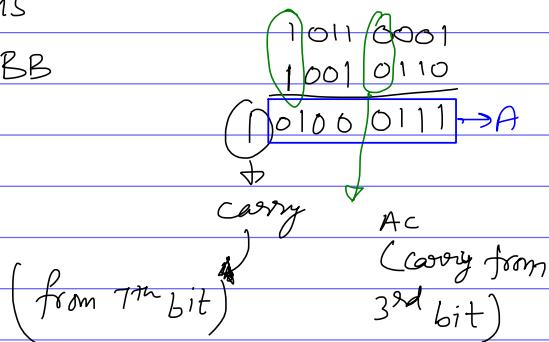
LUT1: MOVC A, @A+PC

RET



Flags getting affected by ADD

C # 45
 AC # BB
 OV (overflow)



$$100 - 64H$$

$$\begin{array}{r} 100 \\ - 64H \\ \hline 200 \end{array}$$

8 bit 2's complement Range:
 $\underline{C8H}$
 ↓
 No overflow

Verify that overflow occurs when 6th bit carry
 XOR 7th bit carry is '1'.

DA A → decimal adjust

if you are using the digits as decimal no. digit
 use DAA after ADD operation

$$A: 99H \rightarrow (153)_{10}$$

$$\text{ADD } A, \#99H \Rightarrow A = 32H$$

$$C = 1$$

$$AC = 1$$

BCD : (Binary coded decimal) $\Rightarrow A(98)_{16}; C=1$

Q A: If there is an AC bit or the lower nibble $y \rightarrow 9$ it adds 6 to it.
 Also, if the lower nibble is adjusted.
 add carry bit to upper nibble
 and if upper > 9 or C bit = 1,
 adds 6 more to the upper nibble.
 and $C \leftarrow 1$.

$$\begin{array}{r}
 153 + 10011001 \\
 10011001 \\
 \hline
 \overbrace{10011001}^{C=1} \quad \overbrace{0010}^{A = 32H}
 \end{array}$$

11 bit address 2Kb memory

RAM address : 20H - 2FH

16 bytes \rightarrow 128 bits \leftarrow first 128 bits
 these are direct bit addressable
 there are 256 direct bit addresses

Bit addresses goes from 00H - 7FH

Remaining 128 bit addresses point to bit addressed SFRs.

P0.0 \rightarrow 80H

P0.1 \rightarrow 81H

P0.2 \rightarrow 82H

27 Jan, 25

99H	77H
99H	77H
<u>1 32H</u>	<u>EEH</u>
DAA: 098	DAA: ①54
\downarrow C	\downarrow C

ACALL (absolute call)

2 Bytes, 2 cycle

Encoding : $a_{10} \ a_9 \ a_8 \ | \ 0 \ 0 \ 0 \ 1 \ a_7 - a_0$

$$PC \leftarrow PC + 2$$

$$SP \leftarrow SP + 1$$

$$SP \leftarrow PC_{7-0}$$

$$SP \leftarrow SP + 1$$

$$SP \leftarrow PC_{15-8}$$

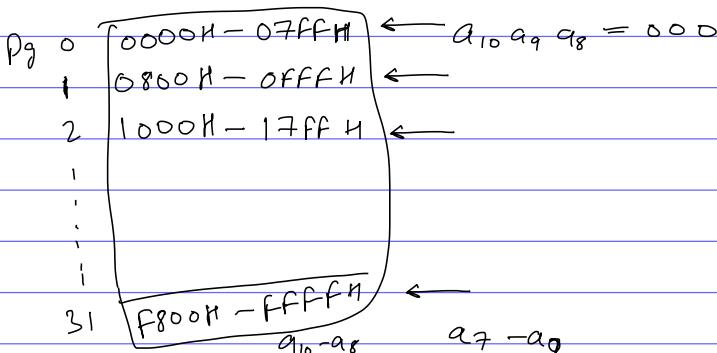
$$PC \leftarrow PC_{15-11}; \text{ add } \oplus 11$$

page is not changed

ACALL addr 11

64KB RAM

2KB RAM



PC \rightarrow 10110011 00011000
5b 3b 4b 4b
page address

RET (Return from subroutine) inst.

Byte - 1
cycle - 2

Encoding : 0010 0010

AJMP addr11 } no return
LJMP addr1b } location is stored on the stack

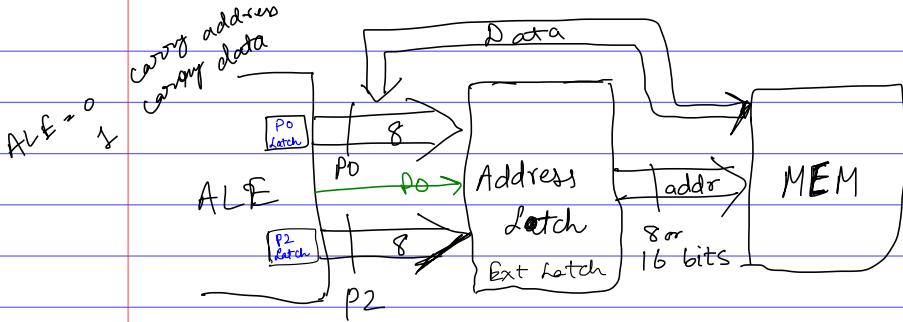
SJMP rel-address,
(2byte) 8 bit address, $-128 \Rightarrow 127$

ANL C, bit
1 byte $\xrightarrow{\text{one byte}}$ one byte (bit address)
for opcode

CLR C	1 byte
CLR <bit	2 —
JC rel	2 —
JNC rel	2 —
JB JB bit, rel	3 —
JNB bit, rel	3 —
JB C bit, rel	3 —

RL A Rotate left 1 bit
RLC A

MUL AB \rightarrow B
Higher 8 bit A
lower 8 bit



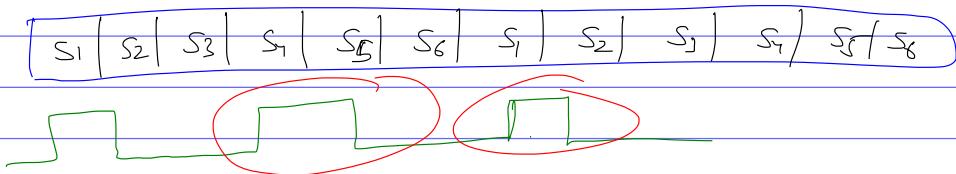
P0. open drain output (as a port)

Both ports P0, P2 when used for accessing external memory.

Logic 1 → strong internal pull up
the port pin

after address is provided on P0,
P0 is tri stated to read the data.

The ext. memory access cycle does not affect the P0, P2 Latches.



case of MOX, these 2 pulses are not there

EA > 0, no internal, use ext. ROM for 0000ff.

PSEN, use internal, then ext.
PSN, Program status for

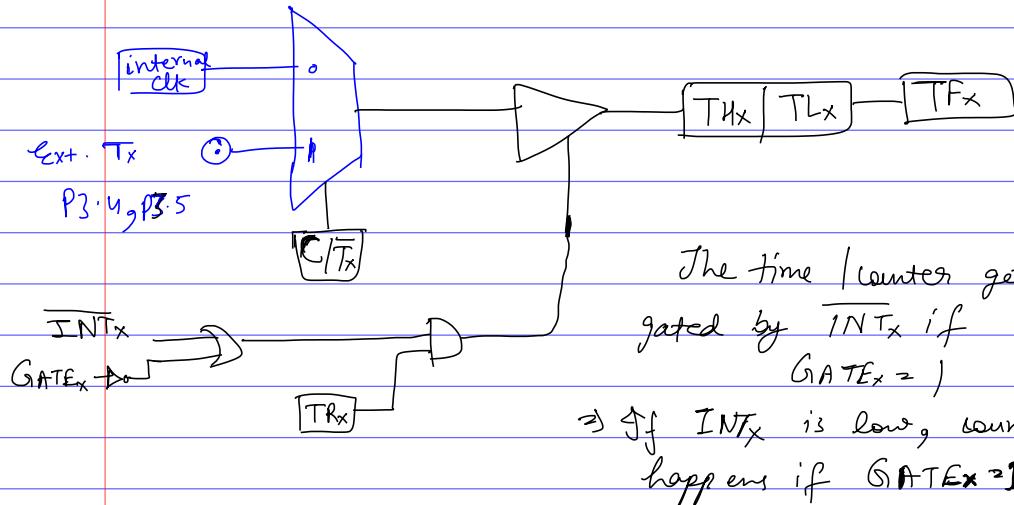
3 Feb, 25

Mode

- Some for timer 1
- 0 → 13 bit timer / counter
- 1 → 16 bit timer / counter
- 2 → 8 bit timer | — , with auto reload.
- 3

TH_x TL_x

(8 bits) (lower 5 bits)
(8) (8)



The timer / counter gets gated by $\overline{INT_x}$ if $GATE_x = 1$

⇒ If INT_x is low, counting happens if $GATE_x = 1$

After vectoring to ISR, TF_x flag gets reset automatically.

MOV $TH_x, \#156$

ISR - TO:

{ Load some present value for TO

1

REF

we wanted a count of #2000

Reload value

= # 6193 in the mode 0 config.

Timers:

↳ serial port

↳ clock / stop watch. } jittery clock or Period should not drift }

↳ stepper motor control

↳ PWM → generating analog waveform using pulses.

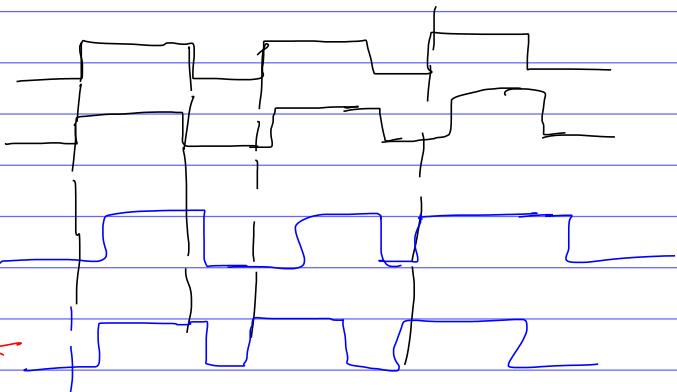
Timing error:

↳ shift in periodicity or average phase

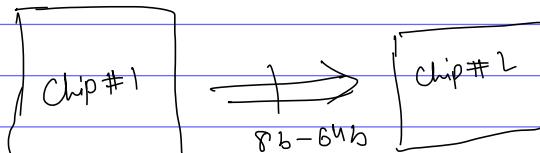
→ Avg. period is accurate but transition points are not well defined.

① Ideal clock

generated clock

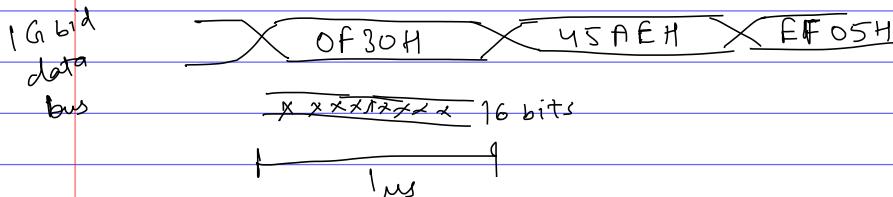
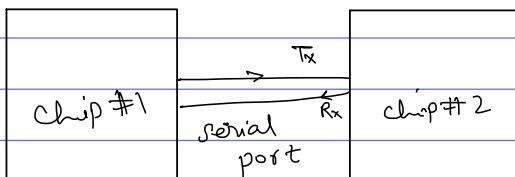


6 Feb , 25



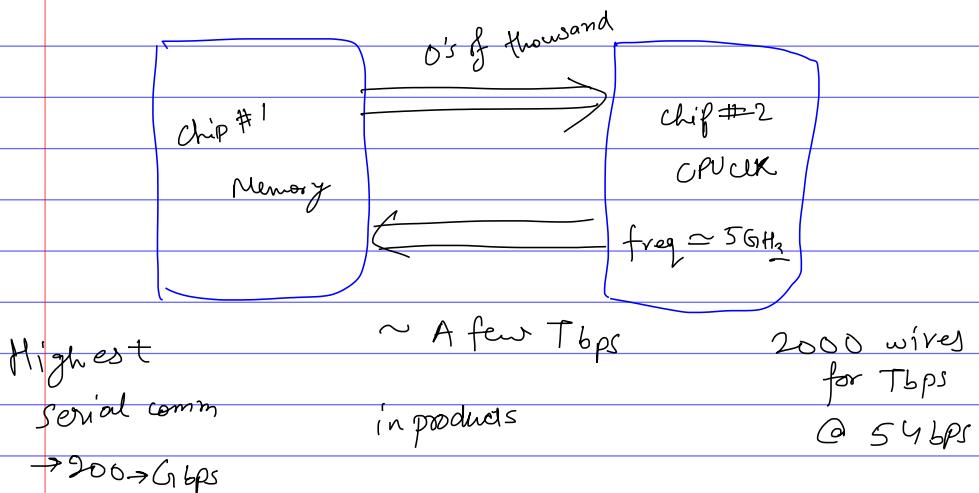
parallel port

Band rate \rightarrow symbol rate (Hz or baud) or baud



$$\text{Band rate} = 1 \text{ MHz} \quad (\text{bit rate} = 16 \text{ Mb/s})$$

Serial comm = for same data rate } for a given
Band rate = 16 MHz } throughput

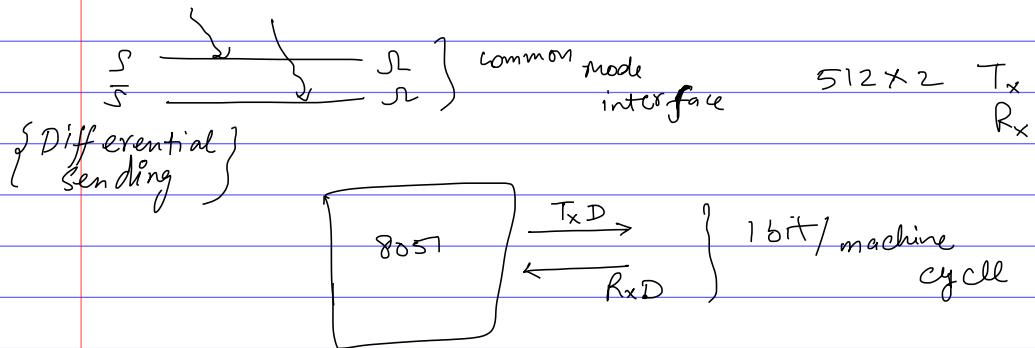


Serial interfaces:

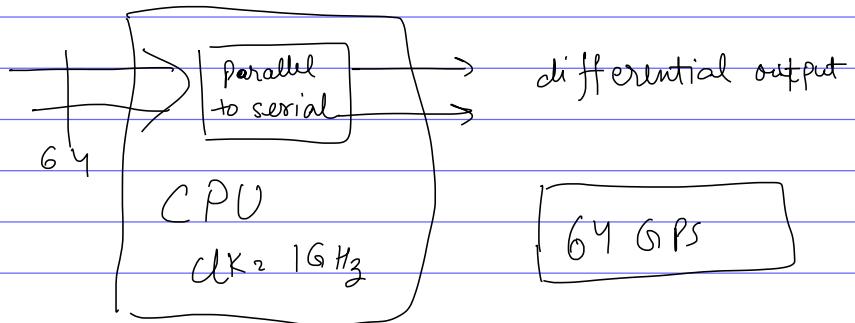
1) Asynchronous — Band rate has small error.
↳ data transitions are not align perfectly with some ref clock transitions

2) Synchronous — data & clock are sync.

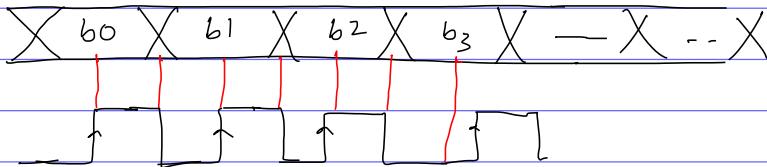
10 Feb, 25



Today's serial link:



1 Gbps



$0.9 \text{ GHz} \rightarrow \text{proper clk}$

recovery & phase
alignment
are reqd.

alignment destroyed
after some cycle

BER of $10^{-6} \rightarrow \text{BER of } 10^{-15}$ using FEC

(Fast error correction)
(overheads $\approx 15\%$)

asynchronous links $\sim 2-3\%$ clk rate mismatch is acceptable

2% mismatch \rightarrow in 50 cycles clk entirely shifts by one bit
variability

low cost crystal oscillator

$\rightarrow \sim 30 \text{ ppm}$

2% mismatch \rightarrow clock drifts from bit center to bit edges
in 25 bit periods.

UART - Universal Asynch. Receiver / Transmitter

Start bit - 0

Stop bit - 1

Stop Start

USART

↳ synch.