Expt. 2 IC 555 and ADC (0804)

Aim: To understand the working of the IC 555 and ADC

- IC 555 → Used for generating CLOCK (square wave)
- ADC → for converting voltages to digital values (binary)

555 Timer

- a very popular integrated Circuit (IC) for Timer applications → as a multivibrator

- Multivibrator vs Oscillator

  ↓ ↓

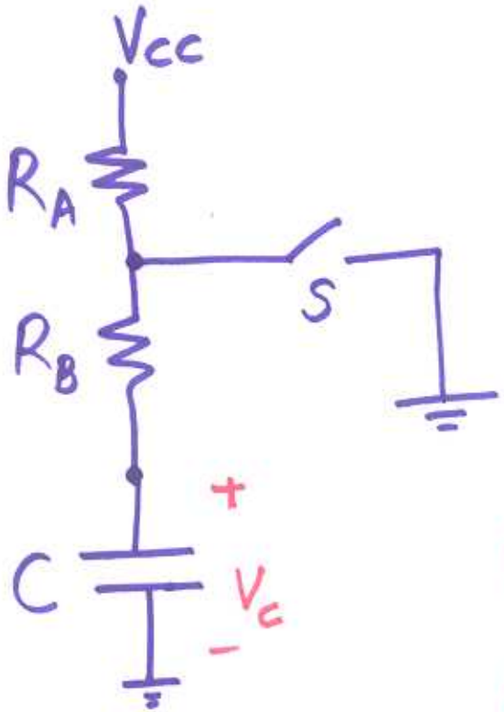
  one that gives square wave output (or rectangular) — Sinusoidal o/p

- Astable Multivibrator → Continuous square wave/ rectangular

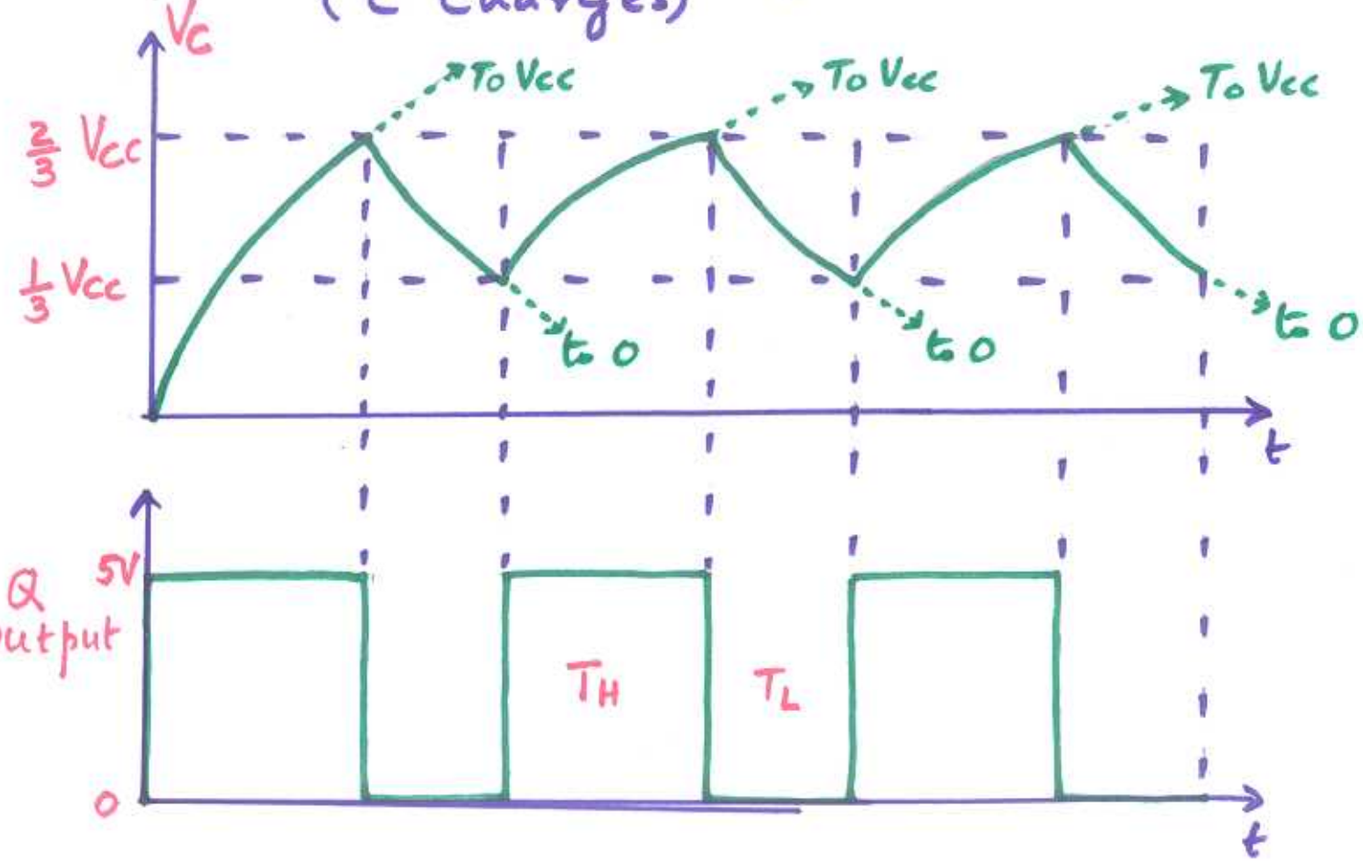
- Monostable Multivibrator → produces ONE pulse when triggered (ONE stable state)

- Bistable Multivibrator → Two stable states (when triggered Toggles between the two)

555 Timer - equivalent Circuit for Operations



- Switch S OPEN for $V_C \leq \frac{2}{3} V_{cc}$ (C charges)
- Switch S CLOSES for $V_C > \frac{2}{3} V_{cc}$ (C discharges) and S remain CLOSED till $V_C \geq \frac{1}{3} V_{cc}$
- S OPENs when $V_C < \frac{1}{3} V_{cc}$ (C charges)



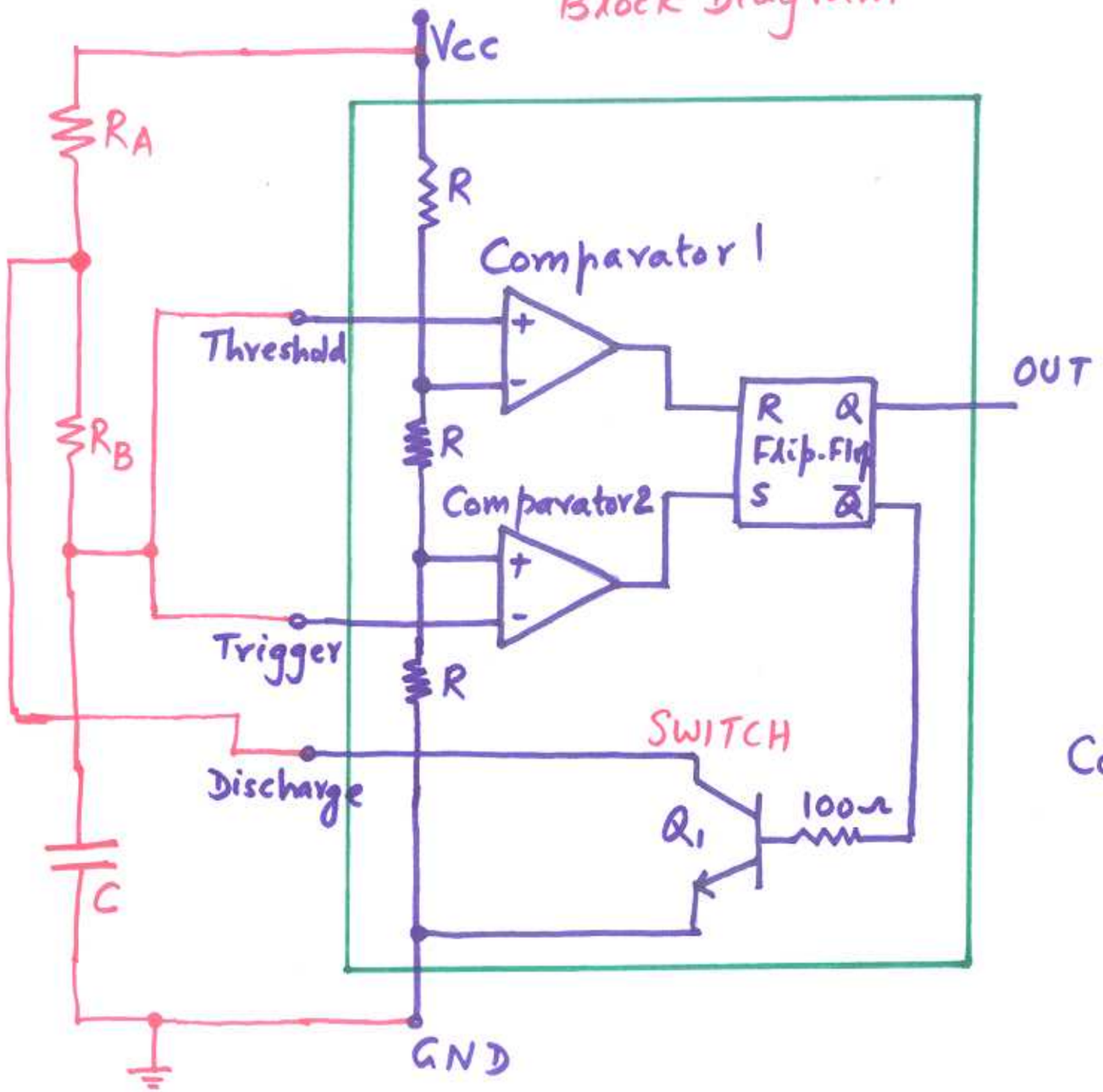
$$T_H = 0.69 (R_A + R_B) \cdot C$$

$$T_L = 0.69 \, R_B \cdot C$$

$$f = \frac{1}{T}, \quad T = T_H + T_L$$

$$T = 0.69 (R_A + 2R_B) \cdot C$$

555 Timer
Block Diagram



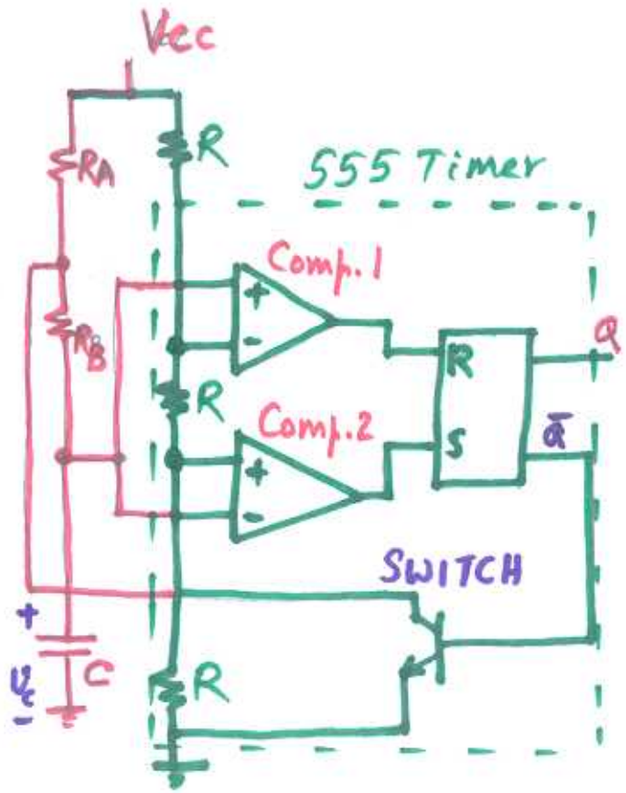
R S Flip-Flop

R	S	Q	$\bar{Q}$	
1	0	0	1	Reset
0	0	0	1	Hold
0	1	1	0	Set
0	0	1	0	Hold
→ 1	1	Not allowed		

Comparator output

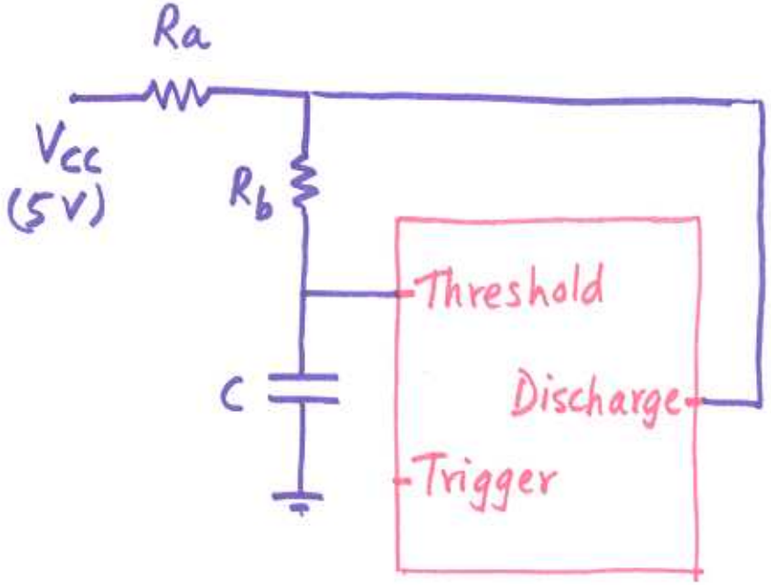
HIGH if $V_+ > V_-$

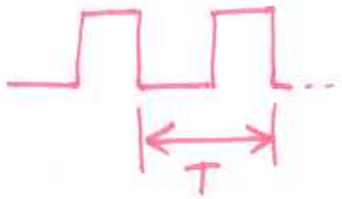
LOW if $V_+ < V_-$



V_c	Comparator 1 (=R)	Comparator 2 (=S)	MODE	Q	$\overline{Q}$	SWITCH	Action
$< \frac{1}{3} V_{cc}$	0	1	SET	1	0	OPEN	C Charging
$\frac{1}{3}V_{cc} < V_c < \frac{2}{3}V_{cc}$	0	0	HOLD	1	0	OPEN	C Charging
$> \frac{2}{3} V_{cc}$	1	0	RESET	0	1	CLOSED	C Discharging
$\frac{1}{3}V_{cc} < V_c < \frac{2}{3}V_{cc}$	0	0	HOLD	0	1	CLOSED	C Discharging
$< \frac{1}{3} V_{cc}$	0	1	SET	1	0	OPEN	C Charging

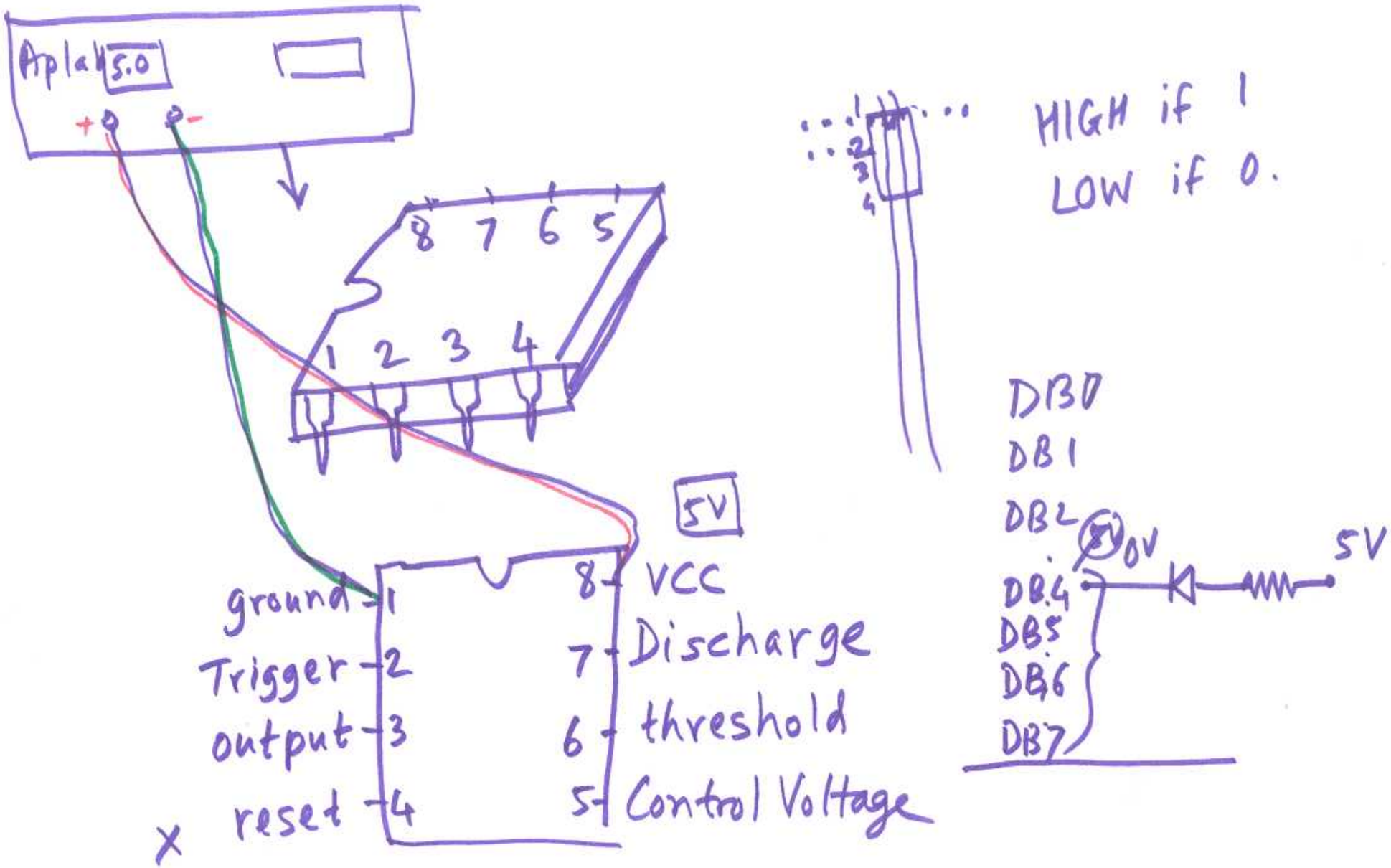
The "clock" signal is provided by a 555 timer circuit.





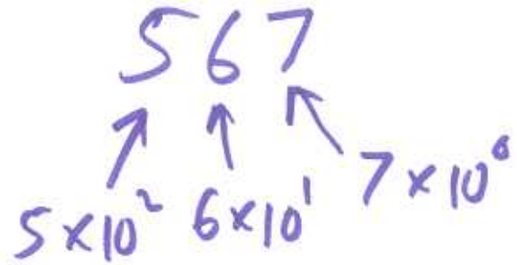
$T = 0.7\,(R_a + 2R_b)C$

$f = \frac{1}{T}$


Aplab 5.0
+
-
8 7 6 5
1 2 3 4
5V
ground
Trigger
output
reset
VCC
Discharge
threshold
Control Voltage
HIGH if 1
LOW if 0.
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
0V
5V

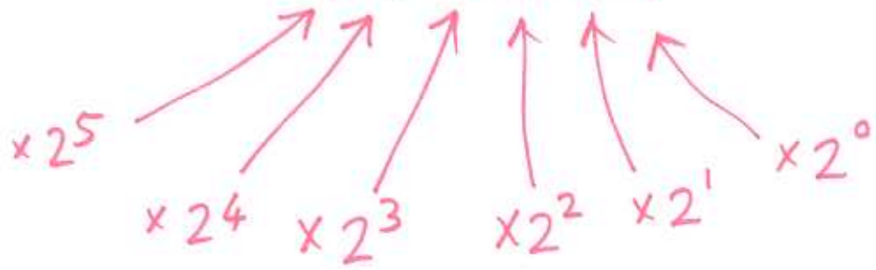
Analog to Digital Converter

Digital (binary) representation



$$101101 = 32+0+8+4+0+1$$

$$= 45|_{\text{decimal}}$$



$$101101|_{\text{base2}} = 45|_{\text{base 10}}$$

N bits $\Rightarrow$ 2^N binary numbers

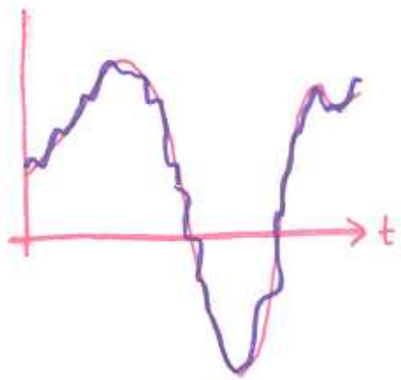
Analog to digital conversion

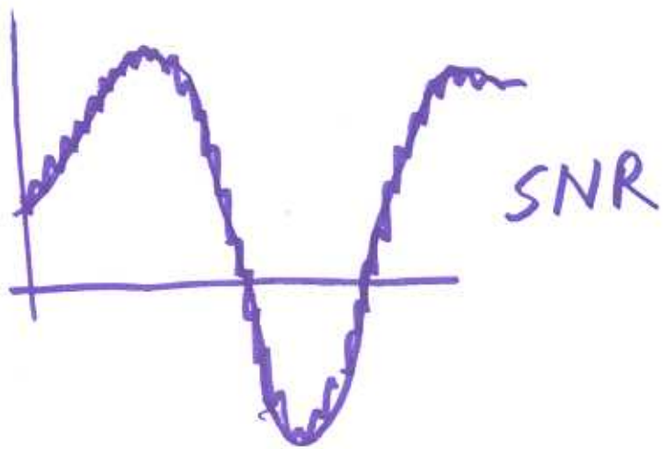
Why digital?

* easier design
* easy to store
* things can be programmed!
* less vulnerable to noise (see next page)
* can integrate large number of functions (on a chip)

Transmission: effect of noise/interference

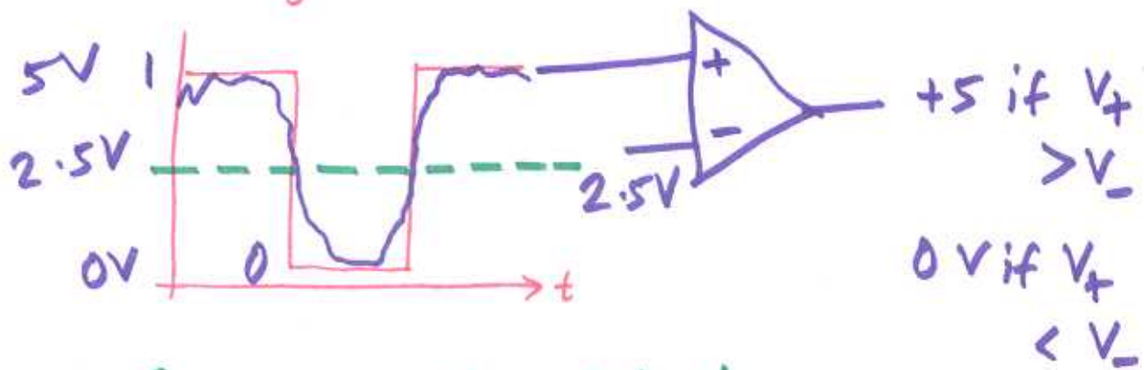
Analog





Digital

Comparator



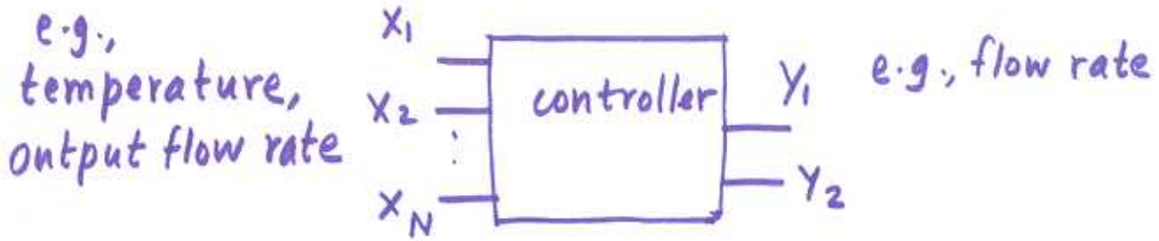
Can recover the original data by using a comparator.

Why digital?

1.357 V → 0101011 .

* We may want to display a temperature

* Control application



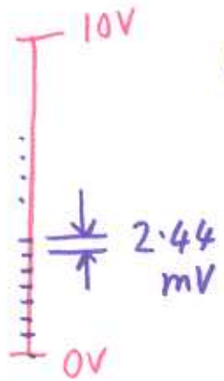
- It is easy to implement control algorithms using DSP chips
- DSP chips can be re-programmed if we want to change the algorithm.

A to D examples

* Full scale measurement range: 0 to 10 V

ADC resolution: 12 bits $\rightarrow 2^{12} = 4096$ quantization levels

ADC voltage resolution $= \frac{10V - 0V}{4096} = 2.44$ mV



LSB 2^0

0000 0000 0000 → 0V

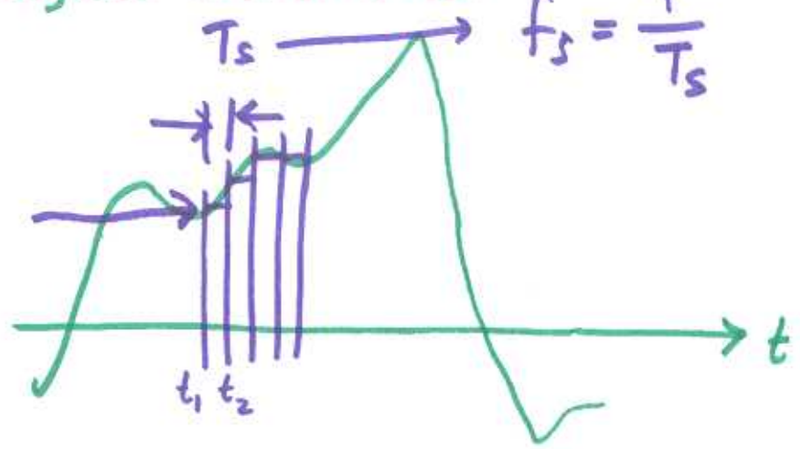
0000 0000 0001 → 2.44 mV

MSB 2^{11}

1111 1111 1111 → 10 V − 2.44mV $\left[= 10V \times \left(\frac{4095}{4096}\right)\right]$

* Some ADCs allow negative voltages, e.g., −10 V to +10 V.

Digital transmission



Convert each "sample" from analog to digital.

Say, we have 8 bits to represent each ~~signa~~ sample.

Bandwidth = 8 × sampling rate

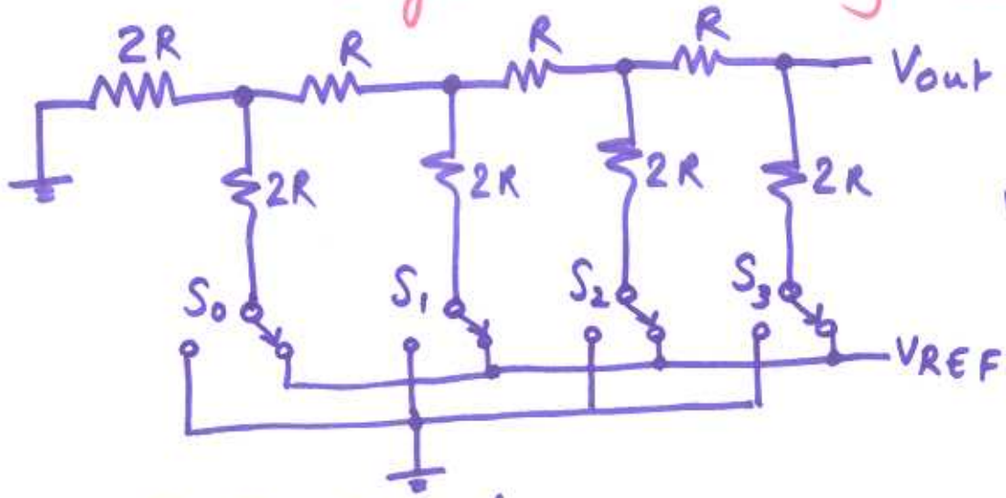
MP3: 128 kbps, sampling rate = 44 kHz

~ 3 bits per sample ?!

This is allowed by compression algorithms

GSM: 13 kbps, f_s = 8 kHz

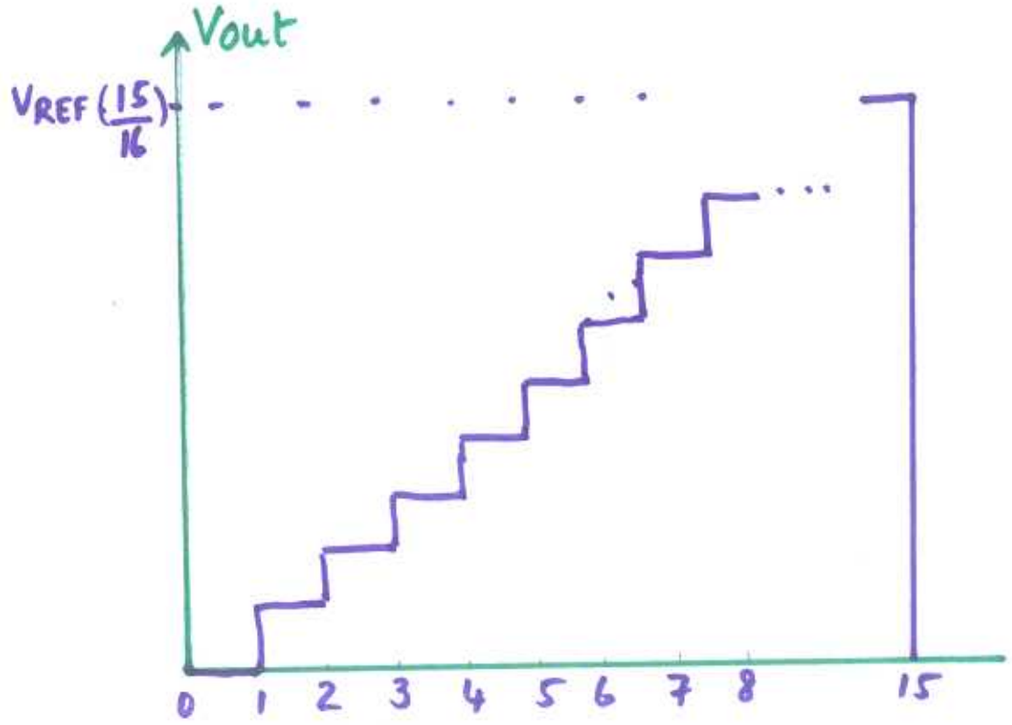
Digital-to-Analog Converter (DAC)



$$V_{out} = \frac{V_{REF}}{2}\left[1.S_3 + \frac{1}{2}.S_2 + \frac{1}{4}.S_1 + \frac{1}{8}.S_0\right]$$

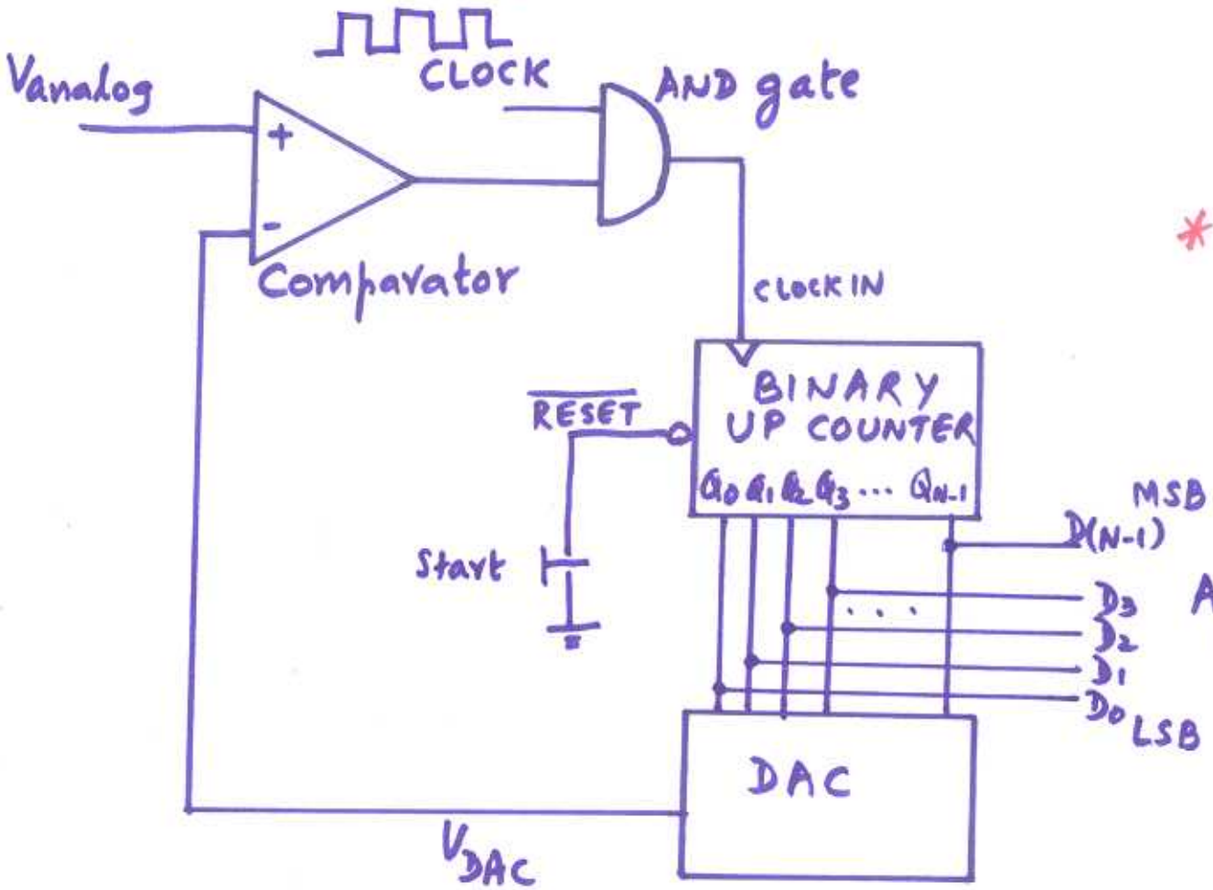
S_3, S_2, S_1, S_0 take either '0' or '1'

S_3	S_2	S_1	S_0	Decimal Equivalent
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15



A Simple Analog-to-Digital Converter (ADC)

Counting-ADC



- Too slow

* - Conversion time (worst case)

$= (2^N - 1).T$ secs

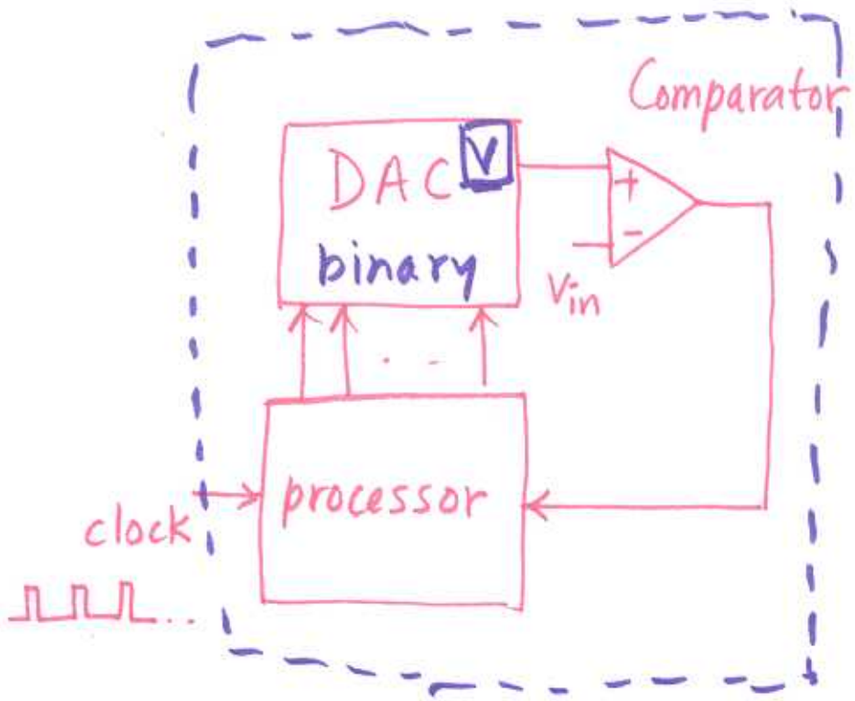
T = clock period

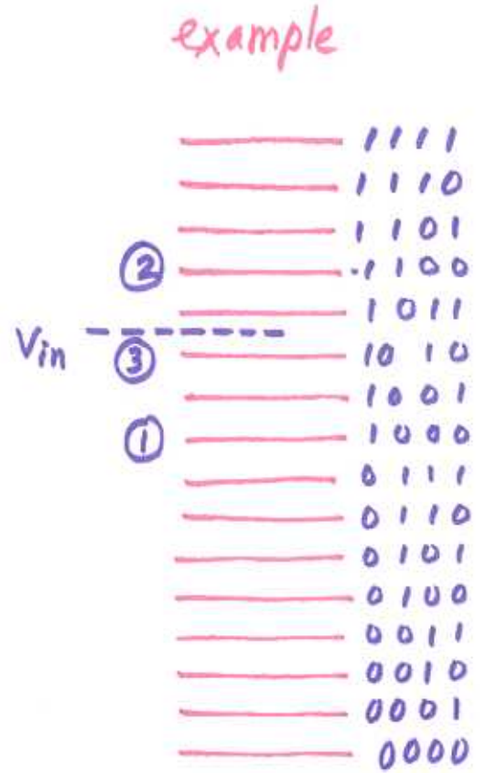
* No. of bits: N

* Resolution

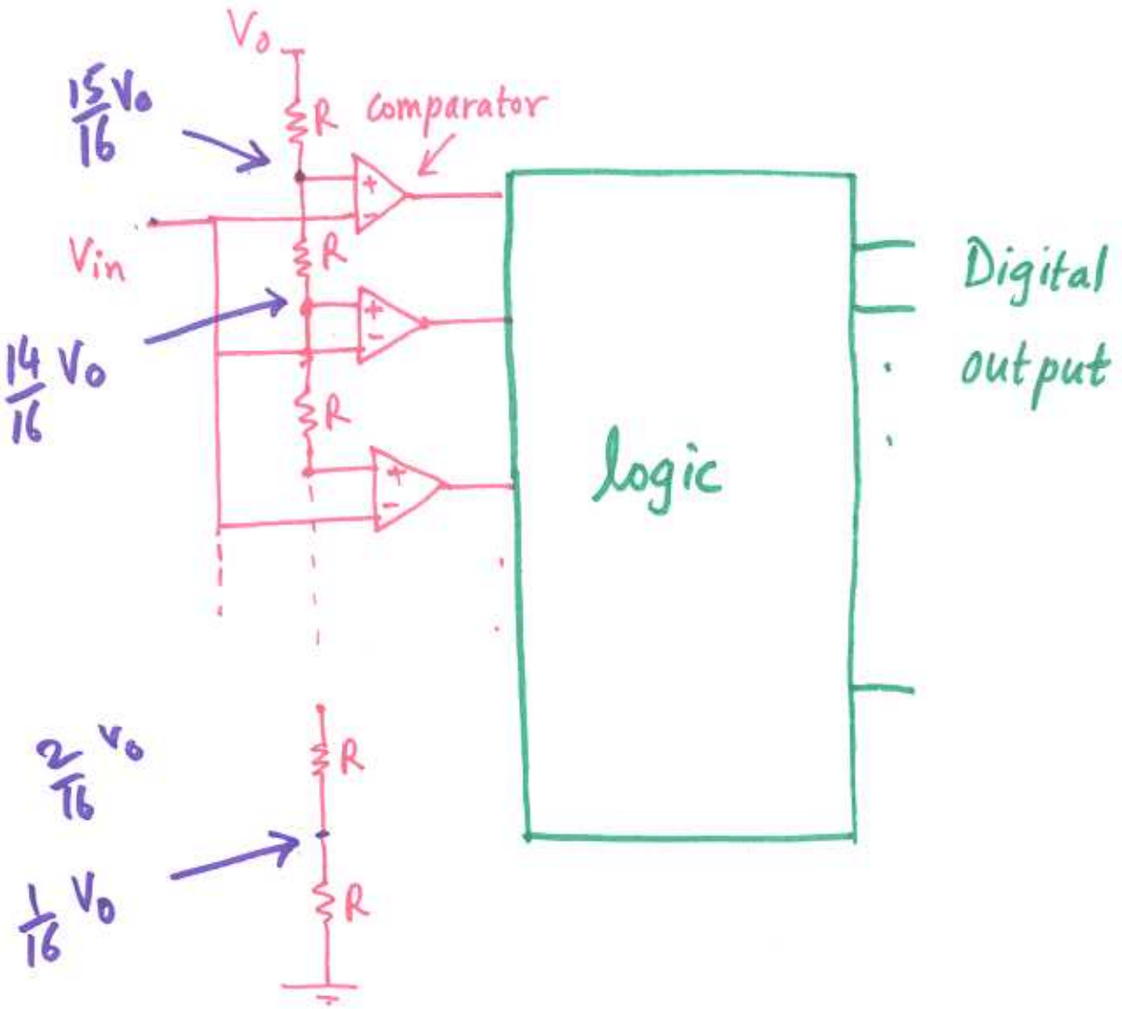
$\frac{V_{REF}}{2^N}$

Successive approximation ADC





Flash ADC


V_0
$\frac{15}{16}V_0$
R
comparator
V_{in}
R
$\frac{14}{16}V_0$
R
logic
Digital
output
$\frac{2}{16}V_0$
R
$\frac{1}{16}V_0$
R

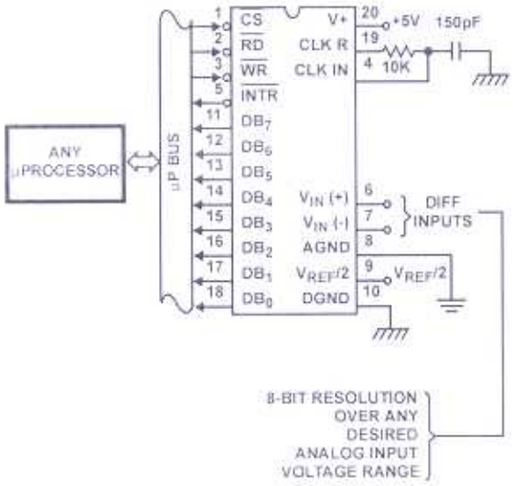

V_0 10V
$\frac{V_0}{16}$

8-Bit, Microprocessor-Compatible, A/D Converters

The ADC080X family are CMOS 8-Bit, successive-approximation A/D converters which use a modified potentiometric ladder and are designed to operate with the 8080A control bus via three-state outputs. These converters appear to the processor as memory locations or I/O ports, and hence no interfacing logic is required.

The differential analog voltage input has good common-mode-rejection and permits offsetting the analog zero-input-voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

Typical Application Schematic

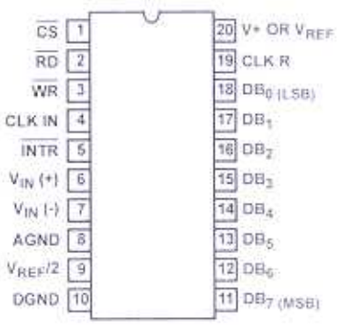


Features

- 80C48 and 80C80/85 Bus Compatible - No Interfacing Logic Required
- Conversion Time ..... <100µs
- Easy Interface to Most Microprocessors
- Will Operate in a "Stand Alone" Mode
- Differential Analog Voltage Inputs
- Works with Bandgap Voltage References
- TTL Compatible Inputs and Outputs
- On-Chip Clock Generator
- Analog Voltage Input Range (Single + 5V Supply) ..... 0V to 5V
- No Zero-Adjust Required
- 80C48 and 80C80/85 Bus Compatible - No Interfacing Logic Required

Pinout

~~ADC0803~~, ADC0804
(PDIP)
TOP VIEW



Ordering Information

PART NUMBER	ERROR	EXTERNAL CONDITIONS	TEMP. RANGE (°C)	PACKAGE	PKG. NO
ADC0803LCN	±1/2 LSB	$V_{REF}/2$ Adjusted for Correct Full Scale Reading	0 to 70	20 Ld PDIP	E20.3
ADC0804LCN	±1 LSB	$V_{REF}/2 = 2.500V_{DC}$ (No Adjustments)	0 to 70	20 Ld PDIP	E20.3

With $\overline{CS}=0$, ADC Conversion Starts when $\overline{WR}=0$

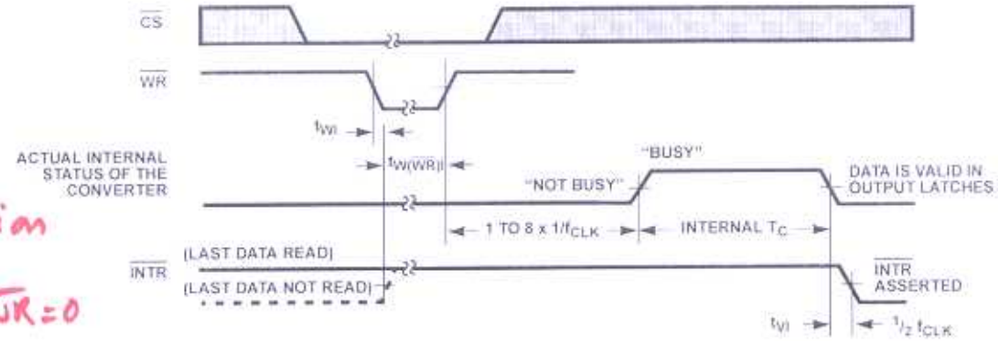
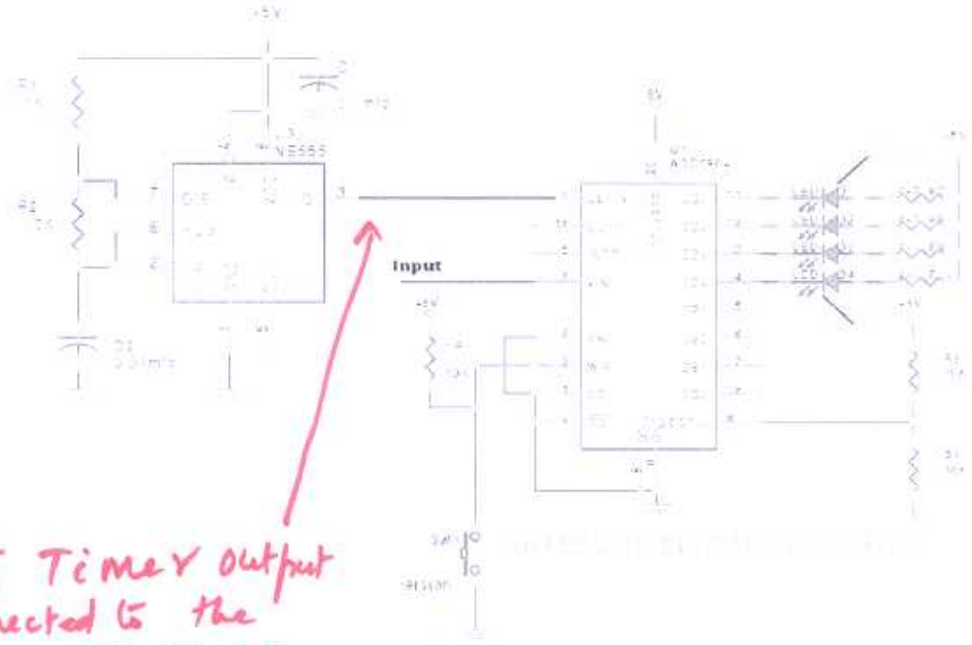


FIGURE 10A. START CONVERSION



555 Timer output connected to the ADC CLOCK IN.

$\overline{CS}$ - Chip Select
$\overline{RD}$ - Read
] connected to GND

$\overline{WR}$ - Write

ADC conversion initiated by pressing switch SW1 (making $\overline{WR}$ = '0')